Estimating Image Motion in Layers: The "Skin and Bones" Model

by

Xuan Ju

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Estimating Image Motion in Layers:
The "Skin and Bones" Model

Xuan Ju
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
1998

This thesis addresses the problem of recovering a locally layered representation of image motion. We develop the "Skin and Bones" model for estimating optical flow that strikes a balance between the flexibility of regularization techniques and the robustness and accuracy of area-based regression techniques. The approach assumes that image motion can be represented by an affine flow model within local image patches. Since some image regions may not have sufficient information to estimate an affine motion model robustly, we define a spatial smoothness constraint on the affine flow parameters of neighboring patches. We refer to this as a "Skin and Bones" model in which the affine patches can be thought of as rigid patches of "bone" connected by a flexible "skin."

Since local image patches may contain multiple motions we use a layered representation for the affine bones. With the possibility of multiple motions at a given point, standard regularization schemes cannot be used to smooth the multiple sets of affine parameters. We therefore develop a new framework for *regularization with transparency* that can applied to produce a smoothed layered motion representation.

The motion estimation problem, with layered locally affine patches and transparent regularization, is formulated as an objective function that is minimized using a variant of the Expectation-Maximization (EM) algorithm. In addition, we also formulate spatial and temporal smoothness constraints on the EM ownership weights at the pixel level. This formulation fits naturally into the EM framework. We also exploit an incremental revision process to estimate the number of layers in each patch using the Minimum

Description Length (MDL) principle. Experiments with synthetic and natural images are provided throughout the thesis to illustrate the method.

Contents

Chapter 1

Introduction

1.1 Motivation

A fundamental problem in the processing of image sequences is the measurement of optical flow. The goal is to compute an approximation of the 2D image velocities, which are the projection of the 3D velocities of surface points onto the image plane. The optical flow field can be used for a wide variety of tasks, for example, video coding, recovery of egomotion and surface structure, scene interpretation and recognition, and robot control. Of these, tasks such as surface reconstruction require that velocity measurements be accurate and dense (i.e., defined at every image pixel).

It is surprisingly difficult to measure optic flow accurately. About 10 years ago, Verri and Poggio [99] suggested that accurate estimates of the 2D motion field were generally inaccessible due to inherent differences between the 2D motion field and intensity variations, while others pointed out that the measurement of optical flow was an ill-posed problem. However, significant progress has been achieved to estimate optical flow robustly and accurately in the past several years. Large numbers of optical flow methods have been developed. These methods are often categorized as gradient-based methods, region-based matching methods, energy-based methods, and phased-based methods. The focus of this thesis is on gradient-based methods, for which the most recent progress has been made.

The gradient-based methods use spatio-temporal derivatives of images as the measurements. To extract motion information, the *data conservation constraint* is applied at every pixel. Due to the widely known *aperture problem*, additional assumptions are required to infer a particular 2D image velocity. According to this extra assumption, gradient-based methods can be further categorized under two schemes, the dense optical flow schemes that use regularization [16, 46, 50, 74, 77, 88], and the parameterized approaches that use regression [4, 18, 93, 103].

Dense optical flow methods, as epitomized by the method of Horn and Schunck [50], require only local image measurements and integrate information over larger areas via regularization. The regularization term enforces spatial smoothness between the motion estimates of neighboring points. To illustrate it, consider the example in Figure 1.1, which shows one black square moving northeast. From the optical flow constraints, only the image motions of the four corners are uniquely defined. Adding the regularization step can result in a dense and smooth flow field defined at every image point. Figure 1.1 shows the ideal smoothed flow field with propagation inside the square. The real method, however, will not produce this but rather a blurred flow. Dense optical flow methods have the advantage of being able to cope with complex and varying flow fields and can be extended to model motion discontinuities in a relatively straightforward fashion [16, 46, 74, 77, 88]. However, despite recent improvements, these methods remain somewhat inaccurate.

Regression approaches, on the other hand, assume that the optical flow within some image region (possibly the entire image) can be modeled by a low-order polynomial [11]. When the model is a good approximation to the image motion these methods are very accurate since one only has to estimate a small number of parameters (for example, six for an affine model) given hundreds, or thousands, of constraints. The problem with these methods is that large image regions are typically not well modeled by a single parametric motion due to the complexity of the motion or the presence of multiple motions. Smaller regions on the other hand may not provide sufficient constraints for estimating the motion. Consider an example sequence shown in Figure 1.2. Three regions varying from large to small are centered at the same image position, where only region 1 contains a single smooth surface. However, this smallest region has relatively little

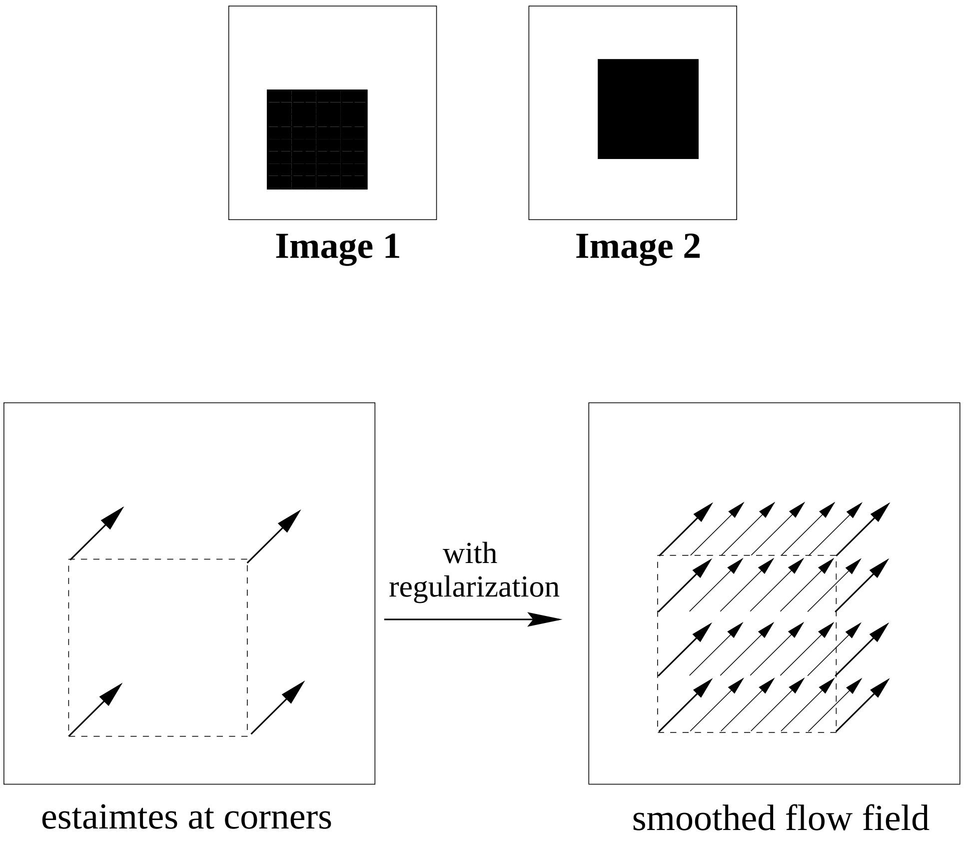


Figure 1.1: Regularized optical flow field of a moving square.

brightness variation, hence, the estimated parametric motion model may be sensitive to noise. On the other hand, the larger regions provide more motion constraints but they contain depth discontinuities, therefore, a single model will not approximate image motion well. Jepson and Black [55] referred to this problem surrounding the appropriate choice of region size as the *generalized aperture problem*.

Generally speaking, the success of parameterized approaches depends critically on the segmentation of the motion constraints. In recent work of motion estimation, people often choose to represent the segmentation information through a *layered* description of the scene [4, 31, 55, 100, 103], in which each layer corresponds to a set of pixels that move over time according to a parametric model of motion [1]. For example, Figure 1.3(a) shows the first image of a sequence, where the camera is panning left to track the little boat,

[1]Wang and Adelson [100] reserve the term motion "layer" to describe the classification of image pixels into distinct motion along with their relative depth ordering. Here we adapt a weaker notion of a layer that does not require knowledge of the relative depth of the layers.

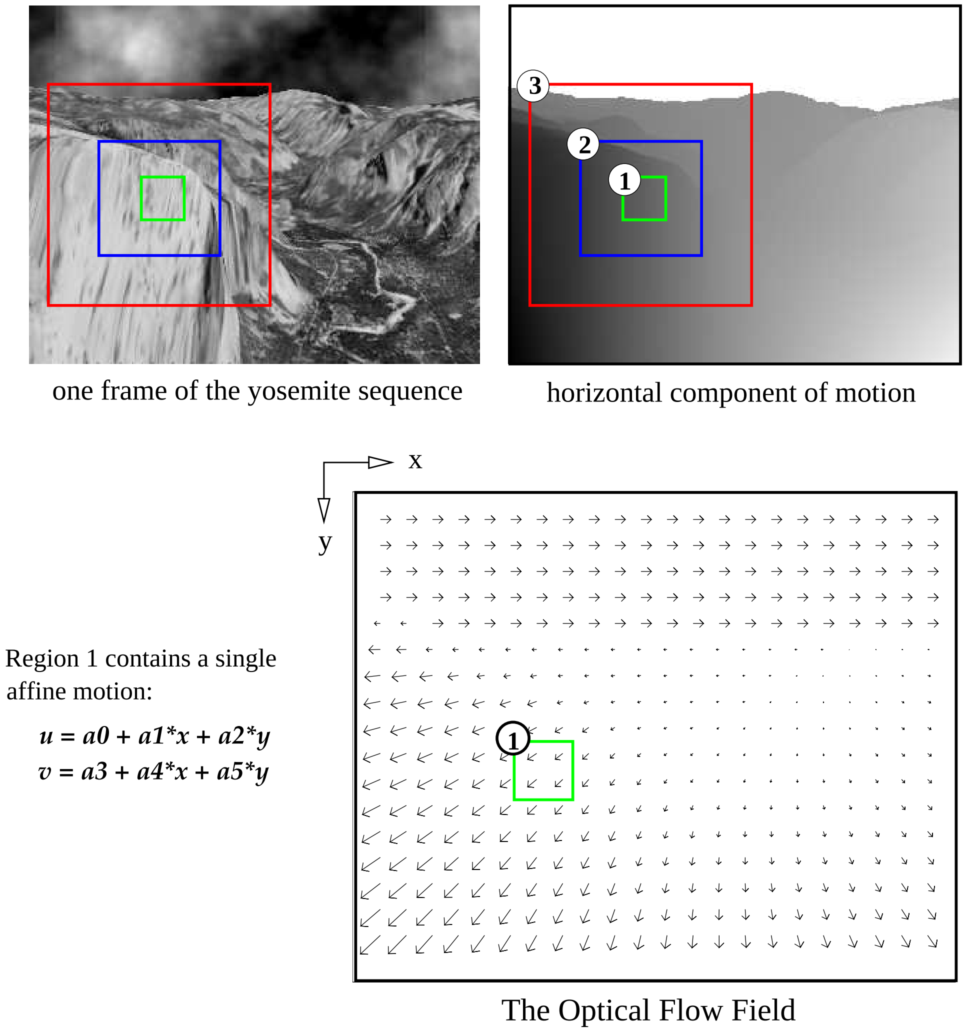


Figure 1.2: Parameterized motion estimation in different regions.

the larger boat is going right. Figure 1.3(b) shows the same image, on top of which are the boundaries of regions that consist of homogeneous 2D motion, as well as the motion field corresponding the affine motion model estimated per region. Figure 1.3 (row 2 to 4) shows the layered representation with three motion layers, each of which is represented by an intensity map, a support map, and a velocity map. Since neither the segmented regions nor the layer maps is know in prior, the current trend in parameterized motion estimation is to find the joint solution of segmentation and motion estimation. So far, this problem remains difficult.

Recent work on optical flow can be seen as trying to find a balance between dense optical flow schemes and parameterized schemes, such as using appropriate image re-

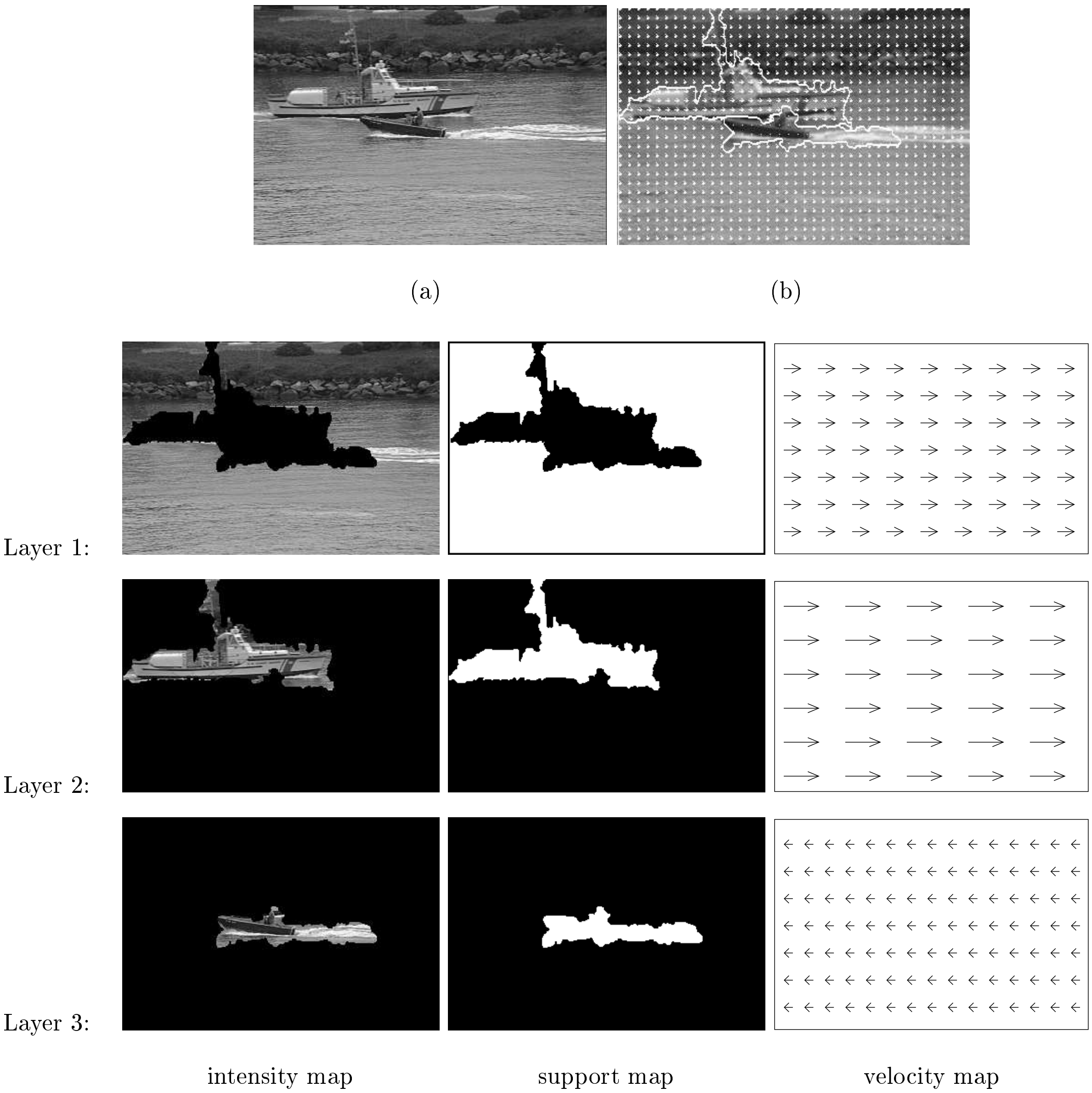


Figure 1.3: **A layered representation of image motion:** Each layer contains three different maps: (1) the intensity map, (or the texture map); (2) the support map, which is unity at the pixel that belongs to the layer, otherwise, zero; (3) the velocity map, which describes the image motion of the layer.

gions that are segmented from the static image [18], learning parametric models which can model more complex motions [22], applying robust motion estimation techniques independently within local image regions [8], and estimating motion models in adaptively selected local patches while image motions between patches are smooth [93].

We have been driven in this work by the desire to develop a method that has the accuracy of the parameterized area-based approaches but which is applied locally. The goal of such a motion estimation algorithm is to recover local parameterized motion models while the models in neighboring patches are smoothly connected. This work shall combine features of both the regularized and parameterized methods to yield a new approach that has the accuracy of the parameterized motion approaches with the generality and flexibility of the regularized approaches.

1.2 The "Skin and Bones" Model

We consider first a basic approach and then extend it to deal with multiple motions. The basic approach tiles the images with a fixed set of local rectangular patches. The motion within each patch is assumed to be affine, though other models could be used. Since, within local patches the affine motion model may be under-constrained due to insufficient spatial variation in the image, we add a regularization term that embodies the assumption that the image motions at the boundaries of neighboring patches should be smooth. We refer to this formulation as "Skin and Bones" where the parameterized patches can be thought of as rigid pieces of "bone" that are connected by a flexible skin. We formulate the problem as an objective function with a data term that enforces the affine flow model within a patch and a prior term that enforces spatial smoothness at the boundaries of patches. The objective function is optimized within a robust estimation framework [16] that accounts for discontinuities at the boundaries and violations of the affine flow model within a region.

The basic method works well when the scene contains a smoothly varying flow field with few motion discontinuities. However, when a given patch contains multiple motions, the basic method will tend to recover only the dominant motion. To generalize the method requires two extensions. First, we must recover multiple motions within a region when they are present. We assume that the motion within a patch can be represented by a small number of affine motions that can be thought of as "layers". Pixels are assigned to layers and the motion of each layer is estimated using a robust mixture model formulation [4, 20, 55, 72] that accounts for "outliers" [45] which cannot be represented by any of the layers. The assignment to layers and the estimation of the motions is achieved using a variant of the Expectation-Maximization (EM) algorithm [72]. This basic approach recovers layered affine motions simultaneously within a patch, and is applied to each image patch independently.

Each image patch may now have multiple motion estimates associated with it and this necessitates the second extension. How can one regularize such a set of layered measurements? If the motions could somehow be grouped into consistent layers (e.g. [56, 100]) then each layer might be regularized independently. But such a segmentation may be difficult or impossible to find if the motions have not already been regularized. Also, instead of the notion of global layers, we prefer a local solution for organizing and smoothing the motion. We take a simple local approach that exploits ideas from robust statistics. Consider a single patch with multiple motion estimates and its four nearest neighboring patches which may also have multiple affine motion estimates. Our approach "connects" every layer in the center patch with every layer in all the neighboring patches. To regularize a particular layer one considers all possible neighboring motions within a robust statistical framework. In such a framework, neighboring layers that have similar motions at the connecting boundary have a strong influence on the solution for the center patch while layers with dissimilar motions will be treated as outliers with little, or no, influence. In a sense, layers have an "affinity" for neighboring layers with similar motions. This "soft" grouping takes place automatically within the robust framework. We call this method *regularization with transparency.*

The "Skin and Bones" model is extended to include an incremental revision process which is used to find the most appropriate number of layers needed. This process is based on the Minimum Description Length principle [83]. At each revision step, a new layer is added. Among two mixture models that have different numbers of layers, we choose the one whose total encoding length is shorter.

In addition, we deal with multiple frames in the "Skin and Bones" model. Intuitively, long image sequences allow us to exploit information over time to improve the estimation of optical flow and the segmentation of the scene. Most of the previous multi-frame approaches presented in the literature assume motion to be constant in time. This assumption has a strong limitation when the image motion or the camera motion is changing over time. Generally speaking, *a priori* knowledge about the evolution of image motion in time is not available. From the layered representation point of view, pixels that are assigned to the same layer are likely to belong to a single layer in the following frames. We describe an incremental approach that takes advantage of such a temporal coherence assumption. The ownership weights from time $t-1$ are used to predict the initial assignments to layers at time t. A prior model of ownership weights based on the estimates from previous frames is integrated in the "Skin and Bones" model to favor consistent grouping over time.

1.3 Thesis Overview

The first portion of the thesis is devoted to the generic "Skin and Bones" model for motion estimation, and the second portion addresses several extensions and a particular application of the original "Skin and Bones" model.

Chapter 2 reviews previous work. First, work related to dense and parameterized optical flow estimation is presented. This forms the background of work on motion estimation. Then, previous methods for selecting motion models and for estimating image motions in long sequences are discussed briefly. Given the diversity of techniques employed, more background and related work is described as the need arises in the following chapters.

Chapter 3 presents the single-layer case of the "Skin and Bones" model. The original technique of robust motion estimation of a rigid object is reviewed at the beginning. Locally affine patches ("bones") are used to estimate image motion, while the skin (the regularization term) is included to improve the stability and accuracy of the approach. Then we discuss the problem caused by the *generalized aperture problem* and show how tiling the image differently can affect the accuracy of the approach. Finally, the limitations of the single-layer "Skin and Bones" model are presented.

Chapter 4 introduces the robust layered affine motion estimation (bones). After introducing mixture models, we formulate the problem using a mixture of affine motions. Then the experimental results of several sequences are shown. We also illustrate the effects of the number of layers used in estimating layered motions. Finally, a spatial constraint is used to enforce the smoothness of the ownership weights between the neighboring points within an image patch.

Chapter 5 presents the "Skin" which is a spatial smoothness term among the multi-layer "Bones". We first propose a general *regularization with transparency* framework, then add the spatial smoothness term (skin) to multi-layer bones. Examples are provided to illustrate the benefits of adding "Skin" to "Bones".

Chapter 6 addresses the problem of selecting the appropriate number of layers within an image patch. In previous formulations of the "Skin and Bones" method, the complexity of the mixture models is fixed, i.e., a two-layer mixture model is used in all the patches. In this chapter, we explore the problem of finding an optimal representation in a Minimum Description Length (MDL) paradigm [83]. We will choose the number of layers that have the minimum encoding cost while explaining the observations best. An incremental approach is developed, which will add a new layer if the revision improves the motion estimates significantly.

Chapter 7 extends the "Skin and Bones" model over time. A temporal coherence constraint on the ownership weights is added to take into account multiple frames. A prior model of the ownership weights is derived from previous estimates. The prediction step allows the coherent grouping of the same motion, and the estimation step provides temporal smoothing of the ownership weights over time. We develop an incremental motion estimation method that applies spatial and temporal prior model in the EM-algorithm to refine the motion estimates and the layered representation of the image.

Chapter 8 presents a particular application that tracks articulated motion over time. The *Cardboard People* model is introduced as a special application of the generic "Skin and Bones" model. To estimate articulated human motion we approximate the limbs as planar regions and recover the motions of these planes ("bones") while constraining the motion of the connected patches to be the same at the points of articulation ("skin"). Experimental results of tracking a walking person are presented.

Chapter 9 presents a summary of the contributions made by this thesis, as well as the open questions and future research directions.

Chapter 2

Background

This chapter presents a review of the methods proposed in the literature for estimating image motion. The organization of the chapter is the following:

In Section 2.1, we first review the two basic assumptions of optical flow estimation, then review the estimation methods based on regularization and area-regression. Section 2.1.1 presents the classic assumptions used for optical flow computation. The first assumption is the *data conservation assumption*, which alone is not always sufficient to recover the image motion. This problem is referred as the *aperture problem*. Therefore the *spatial coherence assumption* is used to regularize the ill-posed problem. Section 2.1.2 presents an overview of the regularization approaches. On the other hand, area-based regression can also solve the problem, however, the *generalized aperture problem* will be encountered. This problem refers to the choice of aperture size and to the dilemma surrounding the choice. Section 2.1.3 describes the parameterized approaches that assume a single motion in a region. We focus on the approaches that can improve the accuracy of the estimates and the robustness of the motion estimation method. In this section, we also describe the methods that recover multiple motions by successively estimating dominant motion models. These methods are categorized as *sequential methods*. Finally in Section 2.1.4, we introduce the approaches that use adaptive window techniques to account for image structure.

Section 2.2 reviews the methods that model multiple motions directly. Section 2.2.1 presents an overview of the recovery of layered motion representations. The methods are usually categorized as *simultaneous methods*. In Section 2.2.2, we also describe some methods that identify multiple motions from pre-computed dense optical flow fields. Section 2.2.3 discusses approaches for estimating the number of motion components, or motion layers, present in the image.

Section 2.2.4 describes the methods that deal with motions which are significantly more complex than simple polynomials (like affine).

Section 2.3 addresses the problem of integrating temporal information over time to improve the motion estimates. In Section 2.3.2, we first discuss the problem caused by aliasing from large motions, then review the methods that can be used to overcome this problem. Temporal smoothing is one common method used to avoid aliasing. Next, we introduce the temporal continuity constraint, which is used to predict and constrain changes in image velocity over time. Section 2.3.3 presents spatio-temporal approaches that model time-varying motion parameters as polynomials of time. Section 2.3.3 reviews incremental approaches, such as those based on Kalman filtering and Bayesian generalizations of Kalman filtering.

The last Section 2.4 summarizes the state of the art of motion estimation techniques, and presents the advantages of the "Skin and Bones" model.

2.1 Optical Flow Methods

When objects move in front of a camera, or when a camera moves through the environment, the relative motion between the objects in the scene and the camera gives rise to corresponding changes in the image sequence. These changes are usually characterized by observing the apparent motion of some brightness patterns in the image. The calculation of apparent motion is only one step towards dynamic scene understanding. In this chapter, we review the previous work of optical flow methods[1].

2.1.1 Constraints on Image Motion

To extract 2D motion information, we often apply the *data conservation constraint*, which states that the image brightness within the region of analysis remains constant in space

[1] *Optical flow* is the 2D field of instantaneous velocities of brightness patterns in the image plane.

and time in the direction of image motion. That is,

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x}), t - 1), \tag{2.1}$$

where I is the image brightness function and t represents time. $\mathbf{x} = (x, y)^T$ denotes an image point, and $\mathbf{u}(\mathbf{x}) = (u, v)^T$ is the two dimensional image motion at $\mathbf{x}$. This equation is commonly linearized by taking the first order Taylor expansion of the right hand side. Simplifying gives rise to the *optical flow constraint equation* [11, 49]

$$\nabla I \cdot \mathbf{u}(\mathbf{x}) + I_t = 0, \tag{2.2}$$

where $\nabla I = (I_x, I_y)$, and the subscripts indicate partial derivatives of image brightness with respect to the spatial dimensions and time at the point $(\mathbf{x}, t)$.

The motion constraint equation (2.2) provides a single constraint on a 2D unknown velocity $\mathbf{u}$, therefore is insufficient to determine a unique solution. This is commonly referred to as the *aperture problem* [49]. Additionally, local image gradients are sensitive to image noise, particularly in areas containing little variation in contrast. Thus, the optical flow problem is ill-posed.

The introduction of a spatial coherence constraint restricts the solution to favor a smoothed optical flow field. This additional constraint can be implemented explicitly using regularization. Regularization involves attaching a *penalty term* to the basic data term (i.e., the optical flow constraint equation), such that the solution not only yields a good fit given the observed data but also keeps the penalty term small. The optical flow estimation methods that use local motion constraints and integrate information over larger areas via regularization are referred to as dense optical flow methods.

On the other hand, one typically assumes that the image motion can be described by a single parametric model within a small region, and estimation of the model parameters can be achieved by regression techniques. The parametric model is commonly taken to be constant, affine, or quadratic. With this approach, the region must be taken to be sufficiently large to yield an accurate result. However, larger regions are more likely to contain multiple motions or depth discontinuities, and hence the image motion is not well approximated by a single parametric model. The optical flow estimation methods that recover parametric models within some image region are referred to as parameterized optical flow methods.

The following two sections reviews these two basic optical flow estimation schemes.

2.1.2 Regularization Approaches

Horn and Schunck [50] proposed the standard approach to constrain the estimated velocity vector $\mathbf{u}(\mathbf{x})$ by combining the optical flow constraint equation with a global smoothness or regularizing term, $E_S(\mathbf{u}(\mathbf{x}))$. This leads to the minimization problem over a region $\mathcal{R}$:

$$\sum_{\mathbf{x}\in\mathcal{R}} [(\nabla I \cdot \mathbf{u}(\mathbf{x}) + I_t)^2 + \lambda E_S(\mathbf{u}(\mathbf{x}))] \tag{2.3}$$

In this expression, λ controls the relative importance of the brightness constancy term (or, the data term) and the regularizing term. The introducing of a spatial coherence constraint restricts the permissible optical flow field, and makes the motion estimation problem well-posed. The most commonly used formulation for the regularizing term is the *membrane* model:

$$E_S(\mathbf{u}) = \|\mathbf{u}_x\|^2 + \|\mathbf{u}_y\|^2, \tag{2.4}$$

where the subscripts indicates partial derivatives in x or y direction. It assumes that the optical flow locally corresponds to a continuously varying motion. As noted by many researchers, this assumption is incorrect at motion boundaries, where the regularizing term results in over-smoothed estimates. Clearly, it reduces the accuracy of the estimated flow field and obscures important structural information about the presence of an object boundary [16, 57, 64, 88]. The regularization problem must therefore be reformulated to allow spatial discontinuities.

The regularization problem involving discontinuities has been studied by many researchers. For example, Marroquin [70] presented a method to reconstruct piecewise smooth surface from sparse and noisy data. They used prior knowledge about the geometry of the discontinuities to prevent the blurring of the boundaries between continuous subregions. Terzopoulos [95] proposed *controlled-continuity* constraints that provided general control over smoothness to visual reconstruction problems. The first-order (step), second-order (crease), and higher-order discontinuities are controlled by adjusting a set of parametric weighting functions.

A number of authors applied the Markov random field (MRF) formulations [40] to cope with spatial discontinuities or estimation and segmentation of optical flow [15, 46, 64, 74]. Most of these approaches have focused on the violation of the spatial coherence assumption by introducing a "line process" [40], or by using *weak continuity constraints* [23, 46]. In the "line process" formulation, a boolean field is used to mark the edges between regions and to prevent smoothing across the edges. The *weak continuity constraints* formulation is the analog version of the binary line process. These methods, however, ignored the violation of the brightness constancy assumption.

In [88], Shulman and Herve pointed out that the brightness constancy assumption was commonly violated, for example at occlusion/disocclusion boundaries. They also pointed out first that spatial discontinuities in optical flow could be treated as statistical outliers. Outliers can have a strong influence on the least-squared estimates, therefore, a *robust* estimator is demanded to reduce the effect of observations that would be highly influential if the least squares method was used. In this context, Shulman and Herve proposed a robust regularization approach based on Huber's minimax estimator.

In order to also take into account the violation of the brightness constancy assumption, Black and Anandan [16] introduced a robust data term. They formulated a robust estimation framework using a robust error function in both the data term and the regularization terms, which can account for violations of both the brightness and smoothness assumptions. Black and Rangarajan [19] have shown that there is a deep relationship between these robust approaches and the traditional "line process" approaches.

2.1.3 Parameterized optical flow methods

Parameterized optical flow methods [2, 11, 35, 37, 62, 66, 101] make explicit assumptions about the spatial variation of the image motion within an image region. These methods typically represent optical flow in a region using a low-order polynomial; for example, affine variation is a common assumption. In the parameterized optical flow estimation framework described in [11], the motion transformation between two frames is modeled as

$$I(\mathbf{x}, t) = I(\mathbf{x} - \mathbf{u}(\mathbf{x}, \theta),\ t - 1) \tag{2.5}$$

where θ denotes the vector of model parameters, $\mathbf{u}(\mathbf{x}, \theta)$ is the flow at $\mathbf{x}$, and $I(\mathbf{x} - \mathbf{u}(\mathbf{x}, \theta), t - 1)$ is the image at time $t - 1$ warped towards t. With this formulation, the goal of motion estimation is to find the value of θ which minimizes an objective function. Hence, the motion is estimated by collecting hundreds or thousands of constraints over a large image region and using regression or a Hough transform to search a relatively low-dimensional parameter space. These approaches can recover accurate motion estimates when the motion model is a good approximation to the image motion.

The problem with this approach is that parametric motion models applied over the entire image or arbitrary, pre-selected, regions are rarely valid in real scenes. There are two primary reasons for this. First, the parametric model may not capture the complexity of the observed motion. Second, large regions are likely to contain motions from multiple surfaces at varying depths or from independently moving objects. So, while simple motion models are typically only valid in small image regions, large regions are desirable as more constraints can result in increased accuracy.

Approaches have been devised which ameliorate some of the problems of the parametric models applied in large image regions. The first attempt is to use a regression method that can robustly estimate the motion parameters corresponding to the dominant motion component. Irani *et al.* [52] used global affine and projective motion models to sequentially separate objects in a coarse-to-fine hierarchical framework. The least squares estimates of motion model parameters were used to create a binary mask of pixels belonging to the model or to the outliers by examining the residuals. This in turn was used to improve the accuracy of the estimated parameters and to derive an updated mask at finer levels. This type of robust process is sensitive to the initial quadratic estimate, which may be arbitrarily bad.

Black and Anandan [16], and Odobez and Bouthemy [79] used robust statistics (M-estimation) [45] to estimate a dominant motion in the scene and then fit additional motions to outlying measurements. Redescending estimators (e.g., M-estimators) have the advantage of decreasing the influence of outliers. Both algorithms applied a continuous method to minimize the non-convex objective function using a fixed annealing scheme to lower the scale parameter gradually. Neither of them addressed the problem of automatically estimating the scale parameter.

Bab-Hadiashar and Suter [8] developed a similar, but more robust, approach using the optimally robust Least Median of Squares technique [86] to estimate the dominant motion in a region. Their method is applied *independently* within image patches centered on each pixel in the image. This simple scheme produces very accurate optical flow estimates. The region size must be specified and any method like this will have problems in areas of the image containing little or no texture. Bab-Hadiashar and Suter provided a confidence measure that could be used to ignore these unreliable estimates.

The next attempt is to obtain multiple motions based on successive estimation of dominant motion. Bergen *et al.* [12] propose an area-regression approach for estimating two motions from three frames. It is based on the fact that if one motion component is known, then an image sequence that does not include this can be constructed by a difference operation. The approach uses an iterative algorithm to estimation one motion, then performs a difference operation to remove the intensity pattern that gives rise to this motion, and finally solve for the second motion. The process is repeated and the motion estimates are refined.

In contrast the robust approach can recover multiple motions from two frames, by successively fitting multiple parametric models to the motion constraint equations. These sequential methods follow the outlier detection/rejection paradigms, and therefore require the process to find pixels where the motion model is valid, and the pixels where it is not. Since all the structure corresponding to the non-dominant motion components are treated as a single outlier structure, a method is also required to extract these structures from the outlier structure. Ayer *et al.* [5] proposed such an approach that integrated the prediction error measurements over intensity-based segmented spatio-temporal regions. That is, the recovered dominant motion was tested for validation in segmented regions instead of at each pixel. The use of intensity constraints helps to overcome the "speckling" effect in the recovered flow field, which is caused by small specks in the region that belong to one surface but are incorrectly classified with pixels of another surface. However, it constrains all pixels within a segmented region to have the same motion, thus the quality of the method may depend on the parameters of the intensity-based segmentation algorithm

Odobez and Bouthemy [80] also estimated 2D motion models robustly for each image partition, then used a statistical regularization approach based on multi-scale MRF to update the partition of image given the current motion estimates. If there are new regions whose motion do not conform to the estimated motion models, the process is repeated by fitting a new model to new regions, until no new region is detected.

There have been a number of recent attempts to apply parameterized motion models to smaller, more local, image regions. For example Black and Jepson [18] first segment an image using brightness information and then hypothesize that these regions correspond to planar patches in the scene. Parametric models are used to recover the motion of the patches and when a good segmentation is available, the motion can be estimated accurately. But brightness information alone cannot be guaranteed to provide a good segmentation.

2.1.4 Adaptive Window Technique

Another set of approaches apply a single parametric motion model to regions that are adaptively determined. One of the method based on adaptive window technique was proposed by Okutomi and Kanade [81] for stereo matching, which adjusted the size of correlation region to minimize the uncertainty in the estimate. Their implementation of the approach is limited by the use of a fixed shape (rectangular) window that can not adapt to irregular surface boundaries.

Szeliski and Shum [93] take a different approach based on "quadtree splines" that treats the image as a set of patches of varying size. These patches are connected in a spline-based representation that enforces smooth motion. This has the advantage of providing reliable motion estimates in areas of low texture. The motion within a patch is determined by a parameterized motion model and the patch size varies based on how well the motion in a region can be approximated by a single flow model. The limitation of this approach is that the spline-based representation does not readily admit spatial discontinuities.

Cohen and Herlin [27] presented a framework for motion computation for oceanographic satellite images. Their method is based on the use of a non-quadratic regularization technique to preserve motion discontinuities between connected patches. They used a finite element method to define a non-uniform multi-grid so that the grid in the neighborhood of moving structures was finer. The computation of the adaptive grid is based on a threshold on the normal flow field, and is done a *priori*.

Recently, Memin and Perez [73] proposed an approach that also relied on adaptive multi-grid minimization. In addition to the robust regularization term, they also use the robust data term. Their formulation is similar to the single-layer "Skin and Bones" model presented in [58], which uses a regular grid. Memin and Perez consider an adaptive grid based on a subdivision criterion, which splits a patch into four regions if the standard deviation of the data outliers on that patch is greater than a certain threshold.

These approaches have a limitation caused by the assumption that only a single motion can be presented within a patch. It would preclude the representation of transparent motion or motion with fragmented occlusion, the region size would have to shrink to a point.

2.2 The Layered Representations of Image Motion

The MRF approach is an estimation framework that allows modeling spatial discontinuities within a regularization framework. This approach has been applied to regularized approaches (see Section 2.1.2, and parameterized approaches [80] that estimated multiple models sequentially. As noted by Darrell and Pentland [30], although the MRF formulation satisfies the need to have a framework that can jointly solve for both segmentation and motion estimation, it has a limitation when recovering multiple motions in cases of fragmented occlusion and motion transparency. Furthermore, with this approach, some extra parameters are needed to model the discontinuity process and weight its importance. It is not clear how to determine these parameters, since they are unobservable from the measurement data.

Area based motion estimation methods (see Section 2.1.3) that use the robust techniques can recover the dominant motion in a region accurately by integrating numerous constraints in the presence of outliers. However, any robust estimator can only tolerate a given percentage of outliers. For example, an estimator that has a *breakdown point* [45] of 50% will fail if no object covers a region that is large enough.

In order to overcome these limitations, one solution is to explicitly model multiple motions present in the region of analysis. As described in this section, a number of approaches have been proposed in the literature that follow this paradigm.

2.2.1 Layered Motion Representation

Darrell and Pentland [31] introduce the idea of estimating global motions in *layers* and present an optimization scheme using ideas from robust statistics. In their framework, images are decomposed into a set of layers corresponding to homogeneous motion. The motion of each layer is approximated by a global model with translation and looming – a situation that arises due to camera pan, tilt rotations and forward translations. The support at a given pixel location is computed for each layer by thresholding the residual error obtained from the corresponding motion model. Given this support map, the motion estimation problem is then formulated as the robust estimation using M-estimators with a truncated quadratic function, which reduces the weight of residuals beyond a threshold to zero.

Another approach, proposed by Jepson and Black [55] that models of multiple motions, uses a probabilistic mixture model to explicitly represent the multiple motions within the region of analysis. A set of support maps which indicate the ownership weights of the pixels to each component of the mixture are obtained given the mixture distribution. The idea is to formulate the probability of a motion constraint at a pixel location in a finite mixture form. By introducing an outlier layer, their approach can cope with outliers, which are viewed as data points that are atypical of all components in the mixture model. Jepson and Black use the EM algorithm to decompose the motion into a fixed number of layers. Yuille et. al [110] also exploit robust statistics, formulate the problem in a statistical physics framework, and use an EM algorithm with deterministic annealing to solve for the motion of each layer. In addition, EM-algorithm is used in [103] to estimate a small number of global motions, which are modeled with smooth flow fields.

Since these methods examine the entire image, distant, and quite unrelated, points can have an influence on the estimated motion of a small localized region. These distant motions can act as "leverage points" [86] that pull the solution away from the desired local motion. Weiss and Adelson [105] add a spatial coherence constraint to the weights that assign pixels to layers. This is likely to reduce the effect of leverage points by encouraging layers to have spatially coherent support.

The above approaches can cope with a small number of motions within a region but not with general flow fields. These area-based approaches do not address how to select appropriate image regions in which to apply the parametric models nor how to select the appropriate number of motions or layers.

2.2.2 Recovery of Layered Representation from Optical Flow Field

Another set of approaches applies parametric models to coarse flow fields by grouping flow vectors into consistent regions. Most of these methods rely on the extraction of homogeneous regions by first analyzing local affine motion models in small patches, then applying a clustering or a stochastic relaxation technique in the affine parameter space. These approaches, like the regression approaches above (see Section 2.2.1), assume that the image motion can be represented by a small number of layers within an image region.

Wang and Adelson [100] assume that an image region is modeled by a set of overlapping layers. They compute initial motion estimates using a least-squares approach within image patches [66]. Only a single translational motion is computed in each patch. They then use K-means clustering to group motion estimates into regions of consistent affine motion.

Similarly, Adiv [2] uses a Hough technique to group flow measurements into regions consistent with the motion of planar surfaces. Although the method is quite robust, it is discrete and computationally expensive. In order to reduce the computational load, Adiv also proposed a multi-scale approach in this paper.

Rognone *et al.* [85] propose a method to identify multiple motions from optical flow as well. In their method, the optical flow is divided into fixed size patches, and the expanding, contracting, rotating, and translating components of vector field are computed in each patch using least-squares approach. Global properties of these estimated components are then extracted by a clustering algorithm. Their results contain a number of clusters and labels for each component. Finally, each patch is assigned to one of the possible labels by means of an iterative relaxation procedure.

There are several drawbacks of these approaches. First, the computation of the reliable optical flow itself is a difficult task, and often requires expensive computations. Second, separating the two processes causes the error associated with estimating the motion to propagate into the segmentation stage.

2.2.3 Estimating the Number of Motion Components

The problem of estimating the number of motion components present in the data is a critical issue, which has not been adequately addressed in the literature. The motion estimation formulation itself implies different ways of approaching the problem of estimating the number of components.

Methods that estimate multiple motions through successive estimation of dominant motion basically depend on preset thresholds to stop fitting new models to the unlabelled areas.

Methods that identify multiple motions from pre-computed optical flow all rely on clustering techniques. The problem of estimating the number of classes in clustering algorithms is known to be a very hard task. In [100], Wang and Adelson started the clustering algorithm with a pre-defined large number of models, then iteratively merged the motion models based on a pre-defined threshold on the distance in the motion parameter space. MacLean *et al.* [67] estimated the number of models by testing for the presence of structures in the outlier process. When a structure is detected, a new component is added to the mixture model.

MRF approaches usually only determine local flow estimates and local spatial discontinuities, thus do not solve for the number of models. However, Bouthemy and Francois [24], and Odobez and Bouthemy [80] gave the MRF formulations whose number of labels is iteratively determined based on an energy term. Their methods implicitly require that regions with the same label must be continuous.

Methods based on layered representations imply that the number of motion components must be known. This number may be considered as a parameter to be estimated, or as *a priori* knowledge [55, 110]. A number of methods have addressed the problem of how to choose the appropriate number of parameterized motions that are necessary to represent the motion in the scene. Both Darrell and Pentland [31] and Ayer and Sawhney [4] address this issue by using a minimum description length encoding principle to strike a balance between accurate encoding of the motion and the number of layers needed to represent it. While these methods provide a segmentation of the image based on the support of pixels for each of the layers, the layers extend over the entire image.

2.2.4 Complex Motions

Layered and robust approaches go a long way towards making parameterized models and area-based regression practical. For large regions, they address how to cope with multiple motions and how to estimate the correct number of layers. What the above methods (described in previous sections) no not address is how to deal with motions that are significantly more complex than simple polynomials (like affine). Examples of complex motion include motion discontinuities, non-rigid motion, articulated motion, etc. Complex motions are often impossible to model by low-order polynomials.

To model complex motions in layers Weiss [103] proposes a layered mixture model in which the motion in each layer is modeled with a smooth flow field. This method shows promise as it combines many of the features of the layered models and regularization approaches. Leverage points may still be a problem and the issue of determining how many layers still needs to be addressed. This approach gives up one of the benefits of parameterized motion models, which is that they provide a concise description of the motion in a region which can be used for recognition [21].

Another attempt [22] is to learn models of complex optical flow from examples. Most of recent work on learning parameterized models of image deformation has been focused in the field of face recognition, where the goal is to model deformation between the faces of different people. Correspondences between difference faces, which were obtained either by hand or by an optical flow method, were used to learn a lower-dimensional model. To model complex motions in natural scenes with concise parameterized models, Black *et. al* [22] propose "learning" these models from examples. They use principle component analysis to learn a set of basis flow fields that can be used to approximate the training data. Individual flow fields are then represented as a linear combination of the basis flows. To compute optical flow with a learned model, they directly estimate the coefficients of the linear combination of basis flows from motion constraints. These coefficients are estimated using a robust, coarse-to-fine, and gradient-based algorithm.

2.3 Spatial-Temporal Methods

Until now, we have only discussed the methods that estimate image motion between two consecutive frames. Methods that use longer image sequences can be categorized according to the scheme of how the temporal support is integrated. One category can be referred to as parametric spatio-temporal methods, which model time-varying image motion as a polynomial function of time. These methods use multiple frames as a global set of observations and estimate one parametric spatio-temporal motion in a batch process. The other category is known to be the *incremental* methods, which apply the *temporal continuity constraint* to predict the image motion at current time based on the optical flow field of previous frames. In the context of methods of dense optical flow estimation, the benefit of using more information is demonstrated by Barron *et al.* in [10], in which methods based on the spatio-temporal approach give generally better results than those using two frames.

2.3.1 Temporal Smoothing

The methods based on the optical flow constraint equation (Equation (2.2)) require estimates of the partial derivatives of the sequence. With two frames, derivatives are estimated using $1^{st}-$order backward differences, which are accurate only when 1) the input is highly over-sampled or 2) intensity structure is nearly linear. Spatio-temporal aliasing arises at pixels when neither of the above is satisfied. One way to overcome the aliasing is to apply differential techniques in a coarse-to-fine framework, for which estimates are first produced at coarse scales where aliasing is assumed to be less severe and velocities are less than 1 pixel/frame. These estimates are then used as initial guesses to warp finer scales to compensate for larger displacements.

To cope with large motion, Wu *et al.* [107] proposed a motion estimation algorithm using a wavelet approximation. Traditional methods that use the coarse-to-fine image pyramid by image blurring may produce incorrect results when the coarse level estimates contain large errors that cannot be corrected at the next finer level. This happens when regions of low texture become uniform or certain patterns result in spatial aliasing due to image blurring. In contrast, their method uses large-to-small full resolution regions without blurring images, and simultaneously optimizes the coarser and finer parts of optical flow so that large and small motions can be estimated correctly. Since their method uses a spline function to enforce smoothness in spatial dimension, it results in over-smoothed estimates at the motion discontinuities.

A number of methods have been proposed to apply temporal smoothing to avoid aliasing. Spatio-temporal derivatives of the image brightness function are often computed by discrete convolution with corresponding partial derivatives of a trivariate spatio-temporal Gaussian distribution (see for example [8, 76]). The common forms of spatio-temporal prefiltering are typically scale-specific. The approaches that makes use of prefiltering based on local filters that are tuned to scale, speed and orientation are referred to as *frequency-based* methods [37, 47]. Heeger [47] suggested a non-linear least squares approach to determine image motion from the outputs of the differently tuned filters in each local patch of the image. Fleet and Jepson [37] used the phase component of band-pass filter outputs, which is more stable than the amplitude component. These methods typically rely on the temporal support from a relatively large number of frames in order to get accurate motion estimation.

2.3.2 Temporal Continuity Constraint

When we consider more than two frames, we have additional information that can be brought into the motion estimation problem. Intuitively, the image motion caused by smooth motion of an observer and by continuously moving objects is predictable over time, which is called *temporal continuity*. The temporal continuity constraint has been applied explicitly through regularization, or implicitly through parametric models (with respect to time), in different frameworks as described in the following subsections.

2.3.3 Parametric Spatio-Temporal Methods

Most of the spatio-temporal approaches initially presented in the literature [1, 37, 47] assume motion to be constant in time. This assumption has a strong limitation in situations where the velocity of the objects in the scene or the velocity of the camera changes over time. In general, image motion is neither constant in space nor in time, space-invariant and/or time-invariant motion models can only be effective when applied locally to small spatio-temporal blocks.

Some later attempts, the parametric spatio-temporal approaches, have been made to overcome this limitation. Chen *et al.* [26] presented a spatio-temporal method to model the time-varying motion parameters as some polynomials of time. They assume the motion transformation between frames is constant in space.

Vasconcelos and Lippman [98] extend the method of Chen to model the global image motion in both spatial and temporal dimensions using low-order polynomials. The image motions of the entire sequence can be estimated using a single batch process. They fit affine motion model in the spatial dimension, and a quadratic model in temporal dimension. Since the time-varying motion model is obtained using least-squares estimation, the method will not provide an accurate motion model when multiple motions are present in the scene.

Ayer *et al.* [6] present a similar parameterized motion model in the spatial dimension as well as in the temporal dimension. Time-varying motion parameters are modeled as a linear combination of orthogonal time functions. Their method is embedded in a direct, multi-resolution, and robust framework. They also propose to use an incremental revision process to automatically determine the degree of freedom of the temporal variations at low spatial resolutions where the noise has been reduced. Their model, however, is only able to recover the dominant motion presented in the entire sequence. Yacoob and

Davis [108] also employed robust techniques to estimate the dominant parametric spatio-temporal motion with the affine motion model in the spatial dimension and the constant acceleration model in the temporal dimension.

Francois and Bouthemy [38] propose the use of multiple frames as a single set of observations for identifying multiple moving objects in a scene. Their approach is embedded in a contextual statistical framework, i.e., Markov random field and Bayesian criterion.

2.3.4 Incremental Methods

The incremental methods focus on exploiting information over time to improve the motion estimation. By using information from a sequence of images, optical flow estimates can be refined as more information becomes available.

Irani *et al.* [53] propose a recursive procedure for building an estimation map over time. For each frame, they compute the best affine motion estimate between the current map estimate and this frame. The map is updated by taking a weighted average of the registered current frame and the old map. The rationale is that, as the sequence progresses, the map locks onto the object of dominant motion and the other objects are blurred out. This, in turn, reinforces the lock and improves motion estimates and segmentations.

Murray and Buxton [74] extend the standard spatial neighborhood systems used in MRF approaches to include neighbors in both space and time. They then define a temporal continuity constraint, which assumes that the flow at a location remains constant over time. Although, they introduce spatial discontinuities using a line-process formulation, they do not allow the discontinuities in the temporal dimension.

Black and Anandan [16] also extend their robust formulation of the two-frame motion estimation framework. They treat temporal continuity as a constraint on image velocity, formulate it robustly, and incorporate it into the robust estimation framework. The temporal continuity constraint is formulated in terms of image motion at each image position. They assume *constant acceleration*, and point out the future study of what constitutes a good temporal model.

The above two methods use only the information from the immediately preceding one or two frames to predict the image motion at current time. Considering a more traditional technique that smoothes and predicts sequential data over time, Kalman filtering [61] is the standard optimal filtering technique for estimating the state of a *linear* system. The Kalman filter formulation of optical flow typically consists of the incremental algorithms, such as those described in [71, 91]. When a new image is acquired, the current measurement and its variance are estimated using a two-frame motion method. This measurement is then integrated with the predicted estimate using the Kalman filter update equation, and the variance estimate is revised. The current estimate is then smoothed using some technique (for example, regularization technique). The smoothed estimate is used to warp the flow field to derive the predicted motion estimate for the next frame.

The recursive linear estimator, Kalman filter, applies only to Gaussian densities. It is a special case of a more general probability propagation process. When measurements have a non-Gaussian, multi-modal conditional distribution, the evolving density requires a more general representation, which is based on a Bayesian model for prediction and estimation [48]. This approach has been successfully applied to hand tracking by Isard and Blake [54]. They propose the *condensation* algorithm, which combines factored sampling with learned dynamic models, to propagate an entire probability distribution for object position and shape over time. They achieve robust tracking in clutter, and the result is superior to approaches that use conventional Kalman filtering. Recently, Yuille *et al.* [109] use the same Bayesian formulation to the problem of motion estimation over time. Their framework is implemented by a parallel network model. They demonstrate several psychophysical experiments on motion occlusion and motion outliers.

2.4 Summary

In the preceding sections, a review of different methods for computing optical flow has been presented, along with the solutions that allow each of these methods to deal with multiple motions within the region of analysis. We note that these methods differ in the choice of aperture (or region) size, which ranges from a local neighborhood around each pixel (such as the regularization approaches), to local image patches (such as the approaches based on adaptive window technique), to the entire image (such as some mixture model based approaches). They also differ in the choice of motion models, which can be the affine flow model, or a smooth flow model. Despite all these differences, we see that the current trend in motion estimation is to recover a *motion representation* that is accurate and robust in the presence of multiple motions. Much of the recent work described in this chapter focuses on finding a balance between dense optical flow schemes and parameterized schemes [8, 18, 73, 93], Thus, methods, which have the accuracy of the parameterized area-based approaches and are applied locally, are needed.

While the previous approaches [73, 93], which apply parameterized area-based approaches in local and adaptive regions, move towards our goal of a local parameterized motion estimate, they have a serious limitation due to the assumption that only a single parametric motion is present in a region. Instead we take fixed sized regions of the image that are sufficiently large to estimate affine flow (these can be overlapping as in [8] or non-overlapping as in our previous experiments [58]). We then estimate multiple affine motions using a layered motion estimation scheme similar to those used by [4, 16, 20, 55, 87, 110] and we automatically estimate the number of layers necessary for each region. In addition, we also formulate the problem of estimating multiple motion with a spatial smoothness prior on the layer assignments at each pixel position.

Since the motion of some patches may be under-constrained, we are faced with the problem of enforcing spatial smoothness between the motions of neighboring patches. We define a spatial smoothness constraint on the image motions at the boundaries of neighboring patches. This is similar in spirit to the constraints used in the oriented particle system of Szeliski and Tonnesen [94]. Unlike their work however, we have fixed regions in the image with multiple motion estimates in each region and we wish to add a prior smoothness model in the same spirit as standard regularization techniques. Madarasmi *et al.* [68] approached a similar problem of regularization with multiple depth measurements at each point using a stochastic minimization framework. We observe here that a straightforward extension of the robust regularization scheme described by Black and Anandan [16] provides a simple and elegant solution to the problem.

Chapter 3

Skin and Bones: Single-Layer Case

The robust estimation of parameterized motion models has been explored by a number of researchers. A review of these techniques has been given in Section 2.1.3. In this chapter, we introduce the "Skin and Bones" model with a simplified case that assumes only a single motion is present within an image region. In the following chapters, this will be extended to the multiple motion case.

We formulate the problem of recovering a dominant parametric motion model using a robust and multi-resolution framework based on the method described by Black and Anandan [16]. The formulation here uses robust error norm functions that enable the automatic rejection of outliers through the computation of a scale parameter. Our experience shows that when applying this parameterized scheme globally, the error in the estimated flow field can be large at the points where the model is not a very good approximation to the true image motion.

To improve the accuracy, we estimate locally affine motions through tiling the image into small patches ("bones"). However, smaller patches are more likely to be under-constrained, hence the solutions may be ill-conditioned. It is therefore useful to regularize the optical flow estimation problem by adding a spatial coherence constraint ("skin"). We define the "Skin and Bones" model in Section 3.3, and illustrate its performance by rigorously comparing the estimated flow field with the true optical flow.

What is the appropriate patch size? Section 3.4 discusses the problem of tiling the image. The *generized aperture problem* is described and illustrated through several experiments.

Our assumption, that only a single parametric motion is present in a patch, is often violated. Consider, for example, the situation that occurs when the patch is centered at a motion boundary. In this case, approximately half of the motion constraints will correspond to one motion and half to the other. The motion estimation approach described in this chapter tends to recover one motion only and considers the points belong to the other motion as outliers. Section 3.5 illustrates the limitations of the single-layer "Skin and Bones" model and discusses what is needed to improve it.

3.1 Robust Parameterized Motion Estimation

In this section, we review the recovery of the parameterized image motion based on a framework of robust regression.

3.1.1 Parameterized Motion Models

Parametric models of image motion make explicit assumptions that the image flow can be represented by a low-order polynomial. Typically models include constant flow (translational motion only), rational flow (allows translation and scaling), affine flow (or linear model), and planar flow.

Affine model

When the image region is small, an affine (linear) transformation can well approximate the image flow for a smooth surface [63]. The affine model of image motion given an image region, $\mathcal{R}$, is defined as

$$u(x,y) = a_0 + a_1 x + a_2 y, \tag{3.1}$$

$$v(x,y) = a_3 + a_4 x + a_5 y, \tag{3.2}$$

Using vector notation this can be rewritten as

$$\mathbf{u}(\mathbf{x};\mathbf{a}) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \mathbf{a}, \tag{3.3}$$

where boldface lowercase letters denote vectors and, in particular, $\mathbf{x} = (x,y)^T$ and $\mathbf{a}$ denotes the vector $(a_0, a_1, a_2, a_3, a_4, a_5)^T$. $\mathbf{u}(\mathbf{x},\mathbf{a}) = (u(x,y), v(x,y))^T$ are the horizontal and vertical components of the flow at image point $\mathbf{x}$, whose coordinates (x, y) are defined relative to a particular point. Here this is taken to be the center of the patch (x_c, y_c). The goal of motion estimation is to estimate the vector of coefficients, $\mathbf{a}$.

Planar model

Another appropriate parameterized model would assume that the rigid object is a plane viewed under perspective projection. For small motions, the image motion of a rigid planar patch of the scene can be approximated by the following eight-parameter model:

$$u(x,y) = a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 xy, \tag{3.4}$$

$$v(x,y) = a_3 + a_4 x + a_5 y + a_6 xy + a_7 y^2, \tag{3.5}$$

Using vector notation this can also be rewritten as

$$\mathbf{u}(\mathbf{x};\mathbf{a}) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix} \mathbf{a}, \tag{3.6}$$

where $\mathbf{a}$ denotes the vector $(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)^T$.

3.1.2 Motion Estimation Using Robust Regression

We use the robust framework presented by Black and Anandan [16] to estimate the parameters of motion models. The framework is briefly reviewed in this section.

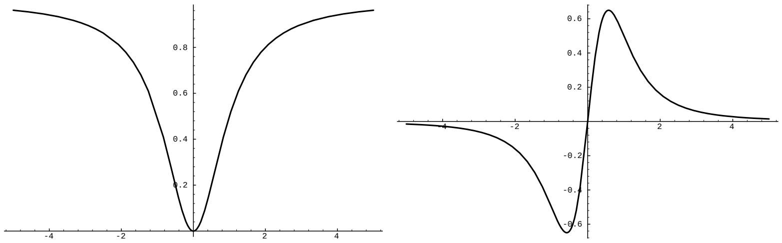
Data Conservation

The assumption of brightness constancy for a given region and a particular parameterized flow model is

$$I(\mathbf{x},t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x};\mathbf{a}(s)), t-1), \quad \forall \mathbf{x} \in \mathcal{R}(s) \tag{3.7}$$

where $\mathcal{R}(\int)$ denotes the pixels in some region s, I is the image brightness function and t represents time. This simply states that the image at time t is the same as the image at time $t-1$ warped by the optical flow $\mathbf{u}(\mathbf{x};\mathbf{a}(s))$ in region s. Recall that this equation is commonly linearized by taking the first order Taylor expansion of the right hand side. Simplifying gives rise to the optical flow constraint equation [11, 49]:

$$\nabla I \cdot \mathbf{u}(\mathbf{x};\mathbf{a}(s)) + I_t = 0, \quad \forall \mathbf{x} \in \mathcal{R}(s) \tag{3.8}$$



$\rho(r, \sigma)$ $\psi(r, \sigma)$

Figure 3.1: A robust error norm and its influence function.

where $\mathbf{a}(s)$ denotes the parameterized motion model for region s, $\nabla I = [I_x, I_y]$, and the subscripts indicate partial derivatives of image brightness with respect to the spatial dimensions and time at the point $(\mathbf{x}, t)$. Since the brightness constancy assumption is expected to be violated due to motion boundaries, shadows, specular reflections, etc., it is important that the estimation of the motion parameters be performed robustly.

Robust Regression

To estimate the motion coefficients within a region, the robust regression approach described in [16] uses an M-estimation technique [45] to recover the dominant motion in the region while treating the motions of inconsistent pixels as outliers. These outliers occur when multiple motions are present in a region. To estimate the parameters $\mathbf{a}(s)$, we minimize

$$E(s) = \sum_{\mathbf{x} \in \mathcal{R}(s)} \rho(\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}(s)) + I_t,\ \sigma(s)), \tag{3.9}$$

with respect to the coefficients $\mathbf{a}(s)$. The value, σ, is a scale parameter and ρ is some robust error function. For the examples in this thesis, ρ is taken to be

$$\rho(r, \sigma) = \frac{r^2}{\sigma^2 + r^2} \tag{3.10}$$

which is used in [16, 20, 41] and is shown in Figure 3.1(a). The shape of the ρ function in Figure 3.1 is such that it "rejects", or down-weights, large residual errors. The function $\psi(r, \sigma)$, also shown in Figure 3.1, is the derivative of ρ and characterizes the influence of the residuals [45]. As the magnitudes of residuals, $|\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}(s)) + I_t|$, grow beyond a point their influence on the solution begins to decrease and the value of $\rho(\cdot)$ approaches a constant, while the influence goes to zero. The value σ is a scale parameter that effects the point at which the influence of outliers begins to decrease.

3.1.3 Estimating the Scale Parameter

One key problem with the use of the robust M-estimator is the estimation of the scale parameter σ. Black and Anandan [16], and Odobez and Bouthemy [79] used pre-specified values of an initial scale, and a pre-determined annealing schedule in each iterative step. Sawhney and Ayer [87] proposed a method to automatically estimate scale parameters. We describe an approach that combined these two schemes; that is, the scale parameter is not only estimated but also controlled by an annealing schedule.

In the use of $M-$estimators for motion estimation, a possibility is to use the assumption that the underlying distribution of residuals can be modeled by an error distribution, and compute the scale, σ, corresponding to this distribution. Like Sawhney and Ayer [87], we also model the residuals using a contaminated Gaussian distribution, where the residuals for the outliers are the contaminants. Given random simples r_i from this distribution which has zero mean, the most commonly used scale estimator is the median absolute deviation, which yields

$$\tilde{\sigma} = 1.4826 \, median_i \, |r_i|, \tag{3.11}$$

Where the factor $1.4826 = \frac{1}{\Phi^{-1}(0.75)} = \frac{1}{0.6745}$ (cf. [86], page 202), with Φ^{-1} denoting the inverse of the Gaussian distribution function.

Finding the global minimum of Equation (3.9) with the estimated scale parameter is complicated by the existence of local minima. We solve the minimization problem using a continuation method developed by Blake and Zisserman [23]. The idea is to begin with a high value of σ and lowers it gradually during the minimization until it reaches the desired value. Initially, σ is experimentally determined so that the objective function is likely to be convex, and no data are treated as outliers. Then as σ decreases the influence of outliers is gradually reduced.

Let r_i denotes the residuals, $\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}(s)) + I_t$, obtained by the robust regression method, and $r_i/\tilde{\sigma}$ denote the standardized residuals. Equation (3.10) can be rewritten as

$$\rho(r,\sigma) = \frac{(r/\tilde{\sigma})^2}{(\hat{\sigma})^2 + (r/\tilde{\sigma})^2} = \frac{r^2}{(\hat{\sigma} * \tilde{\sigma})^2 + r^2} \tag{3.12}$$

where the parameter σ in Equation (3.10) is divided into two parts: the annealing parameter $\hat{\sigma}$ and the scale parameter $\tilde{\sigma}$. We estimate $\tilde{\sigma}$ robustly as described in Equation (3.11), so that the standardized residuals are normally distributed with unit variance. Moreover, an annealing schedule is required by the continuation method, such that the annealing parameter $\hat{\sigma}$ starts at a large value in the first iteration and decreases by a fixed rate 0.95 at each iteration, while its final value in the last iteration is the unit 1.0.

3.1.4 Minimization

Since the robust function is twice differentiable, local minima of Equation (3.9) can be found by using a gradient descent scheme (e.g., Successive Over-Relaxation (SOR)). Also, the continuation method [16] allows the global minima of the final objective function to be found by solving a sequence of robust functions.

Successive Over-Relaxation

Successive Over-Relaxation (SOR) is a relaxation method that involves an iterative process. Comparing to the classical relaxation methods, such as *Jacobi's method* and *Gauss-Seidel* method, SOR reduces the number of iterations required. Here we consider the minimization of the objective function $E(s)$ (Equation (3.9)) with respect to one of the affine parameter $a(s)_i$, the update equation at step $n+1$ and at region s is [23]

$$a(s)_i^{(n+1)} = a(s)_i^{(n)} - \frac{\omega}{T(a(s)_i)} \frac{\partial E(s)}{\partial a(s)_i}, \tag{3.13}$$

where $0 < \omega < 2$ is the *overrelaxation parameter* which is used to overcorrect the value at step n and anticipate future corrections. When $1 < \omega < 2$, overrelaxation can give faster convergence than Gauss-Seidel method. The term $T(a(s)_i)$ is an upper bound on the second partial derivative of $E(s)$. For notational simplicity, we will omit the explicit dependence on region s in this section (3.1.4). We take

$$T(a_i) \;=\; \frac{2}{\sigma^2} K_i \;\geq\; \frac{\partial^2 E}{\partial a_i^2}, \tag{3.14}$$

$$\begin{aligned} K_0 &= \sum_{\mathbf{x}\in\mathcal{R}} I_x^2, \quad K_1 = \sum_{\mathbf{x}\in\mathcal{R}} I_x^2 x^2, \quad K_2 = \sum_{\mathbf{x}\in\mathcal{R}} I_x^2 y^2, \\ K_3 &= \sum_{\mathbf{x}\in\mathcal{R}} I_y^2, \quad K_4 = \sum_{\mathbf{x}\in\mathcal{R}} I_y^2 x^2, \quad K_5 = \sum_{\mathbf{x}\in\mathcal{R}} I_y^2 y^2, \end{aligned}$$

where $\mathbf{x} = (x, y)$ denotes the coordinate of a pixel with respect to a reference point (x_c, y_c), and (I_x, I_y) indicates partial derivatives of image brightness at the point $(\mathbf{x}, t)$. The detailed updated equation (3.13) for each affine parameter is

$$\begin{aligned} a_i^{(n+1)} &= a_i^{(n)} - \frac{\omega}{T(a_i)} \left[\sum_{\mathbf{x}\in\mathcal{R}} P_{\mathbf{x},i} \frac{2r\sigma^2}{(\sigma^2 + r^2)^2} \right], \qquad (3.15) \\ P_{\mathbf{x},0} &= I_x, \quad P_{\mathbf{x},1} = I_x x, \quad P_{\mathbf{x},2} = I_x y, \\ P_{\mathbf{x},3} &= I_y, \quad P_{\mathbf{x},4} = I_y x, \quad P_{\mathbf{x},5} = I_y y, \end{aligned}$$

where r denotes the residual, $\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}(s)) + I_t$, at the point $(\mathbf{x}, t)$. The overrelaxation parameter ω is taken to be 1.9 for all experiments. The spatial and temporal derivatives (I_x, I_y, I_t) are estimated using the simple technique described by Horn [49].

Large Motions

To cope with large motions a hierarchical coarse-to-fine strategy is used. A Gaussian image pyramid is constructed and, starting at the coarsest level, giving an initial guess of $\mathbf{a}(s)$, typically chosen to be zero, the robust regression method is applied to solve the minimization of Equation (3.9). Then the estimated parameters are projected to the next finer level and used as initial estimates, and the process repeated until it converges at the finest level.

The Algorithm

The overview of the algorithm is summarized in Figure 3.2. The algorithm begins by constructing a Gaussian pyramid. At the coarse level (the first row in Figure 3.2), the initial affine motion $a^{(init)}$ is set to zero. The spatial and temporal derivatives (I_x, I_y, I_t) at all image positions are computed using the two images at the coarse level. Then we minimize Equation (3.9) and compute an incremental affine motion estimate $da^{(n)}$, where n denotes the pyramid level. Thus the affine motion model estimated at this level is $a^{(n)} = a^{(init)} + da^{(n)}$, which is projected into the next level. The projected affine motion is $(2 * a_0^{(n)}, a_1^{(n)}, a_2^{(n)}, 2 * a_3^{(n)}, a_4^{(n)}, a_5^{(n)})$.

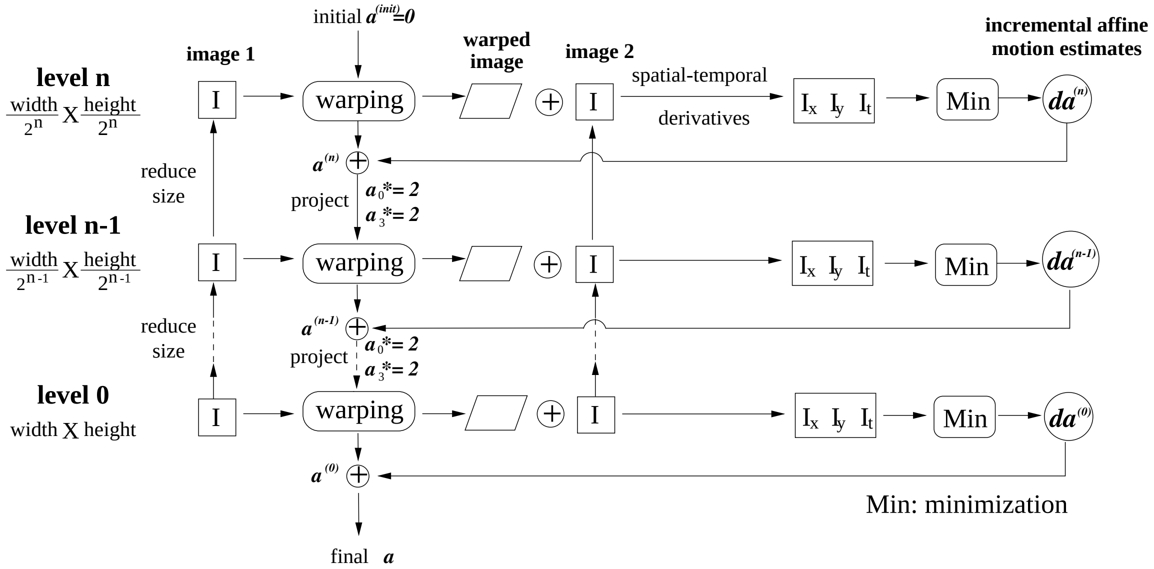


Figure 3.2: The general framework of the robust motion estimation algorithm.

The next level in the pyramid is them processed. The initial motion estimate at this level is the projected affine motion of the previous level. It is used to warp the first image toward the second. Similar to the process in the previous level, the spatial and temporal derivatives are computed, then used to minimize Equation (3.9). The process is repeated until the finest level is reached. This general framework depicted in Figure 3.2 will be implemented for all the methods described in the thesis, while each method has a different minimization process.

The minimization process used in each level iteratively update the change of the affine motion, which is illustrated in Figure 3.3. At the start of the process, the change of the affine motion, da, is set to zero, the annealing scale parameter begins at 5.0, and the number of iterations, n, is set to zero. Then the current residuals at all image positions are calculated and used to estimate the scale parameter $\tilde{\sigma}$ using Equation (3.11). We combine the two parts, $\tilde{\sigma}$ and $\hat{\sigma}$, to get the current scale parameter σ. At each iteration, annealing parameter $\hat{\sigma}$ is lowered according to the schedule $\hat{\sigma} = 0.95\hat{\sigma}$. Once the scale parameter is determined, we update the affine motion estimate according to Equation (3.13), (3.14), and (3.15). The number of iterations used in minimization is 30. The process will also stop if the change in the affine parameters is less than 10^{-6}. The parameters described in this section remain fixed for the experiments in the remainder

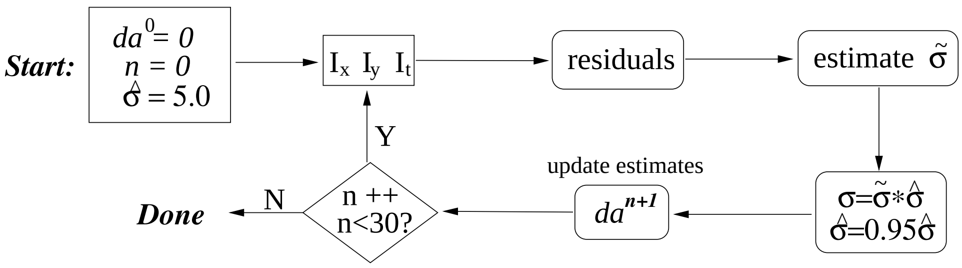


Figure 3.3: The iterative minimization process: robust motion estimation

of this chapter.

3.1.5 Examples

To evaluate the method, experiments are performed using both synthetic and real image sequences. The main advantage of synthetic inputs is that the estimated optical flow field can be rigorously compared with the ground truth data and the results from other published algorithms. We apply the method to two consecutive images in each sequence. An affine motion is estimated over the entire image region. A three level pyramid is used except for the Yosemite sequence, which uses four levels.

The experimental results are demonstrated by the optical flow field, or the horizontal and vertical component of flow, and the outlier mask which shows the pixels whose residuals are larger than a threshold. By examining the ψ-function we see that the influence begins to decrease where the second derivative of ρ is zero. For the error function used in Equation (3.10), this means that when residual is equal to $\sigma/\sqrt{3}$, the maximum influence will be

$$\psi(\sigma/\sqrt{3},\sigma) = \frac{2\sigma^2(\frac{\sigma}{\sqrt{3}})}{(\sigma^2+(\frac{\sigma}{\sqrt{3}})^2)^2} = \frac{3\sqrt{3}}{8}\sigma \tag{3.16}$$

The point when the residual is equal to 2.5σ has an influence that is about one seventh of the maximum influence, and we use this point as the threshold for the outliers. The factor 2.5 was first used by Ayer and Sawhney [4], and is a standard choice.

For the synthetic sequences presented in the following, we can also compute the error in the flow using the angular error measure of Barron *et al.* [10]. They represent image velocities as 3-D unit direction vectors $\mathbf{v} \equiv \frac{1}{\sqrt{u^2+v^2+1}}(u,v,1)^T$. The error between the true velocity $\mathbf{v}_t$ and the estimated velocity $\mathbf{v}_e$ is given by $\arccos(\mathbf{v}_t \cdot \mathbf{v}_e)$. An angular measure of error is convenient because it handles large and very small speeds without the amplification inherent in a relative measure of vector differences. Otte and Nagel [82] pointed out the problem of angular measures, i.e., symmetrical deviations of estimated vectors from the true value result in different angular errors. For example, let $\mathbf{u} = (1.5, 0, 1)^T$ be the true displacement, $\hat{\mathbf{u}}_1 = (2.0, 0, 1)^T$, and $\hat{\mathbf{u}}_2 = (1.0, 0, 1)^T$ are two estimated optical flow vectors. The two angular errors in this example are $7.12°$ and $11.3°$. Therefore, the magnitude of difference vectors, $\|\mathbf{u} - \hat{\mathbf{u}}\|$, can also be used as an error measure. The performance of the our algorithm is quantified, and compared with other published results using both measures.

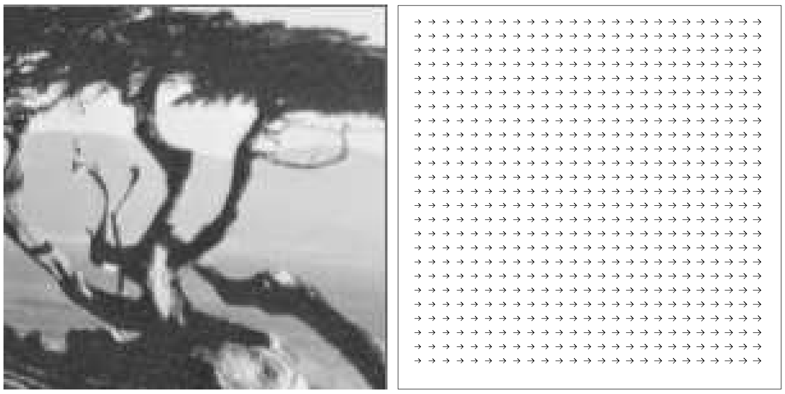
Translating Tree Sequence

The Translating Tree sequence, created by David Fleet, simulates translational camera motion with respect to a textured planar surface (see Figure 3.4(a)). The camera moves normal to its line of sight along its X-axis, with velocities all parallel with the image x-axis, with speed around 2 pixels/frame. Due to the non-zero surface gradient, the sequence consists of a speed gradient in the direction of image velocity (see Figure 3.4(b)). An affine motion model with non-zero divergence and deformation components is sufficient to approximate the image motion of Translating Tree sequence over the entire image region. Therefore, the results of robust motion estimation shown in Figure 3.5 and the error statistics shown in Table 3.1 indicate accurate motion estimates with very little angular error. "Pixel Error" shown in the table is the mean of the Euclidean distance between the estimated flow vector and the true flow vector, while "Average Error" refers to the mean angular error. The results are also compared with other published results [10] in Table 3.2, which shows that our method produced the most accurate result.

Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
			$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
0.013	0.24°	0.05°	100%	100%	100%	100%	100%

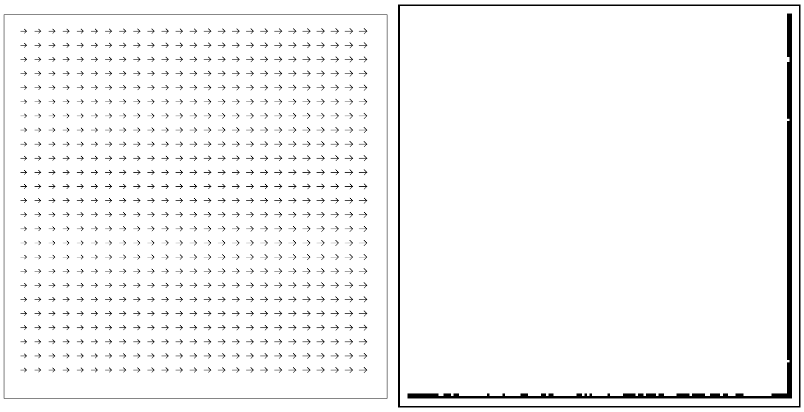
Table 3.1: **Translating Tree Sequence: robust motion estimation;** error statistics.

Additive Noise: We now consider the robustness of motion estimation when a significant amount of noise is present. The noise signal is added to all pixels in the second



(a) (b)

Figure 3.4: **Translating Tree Sequence: ground truth;** (a) image one in the sequence; (b) optical flow field.

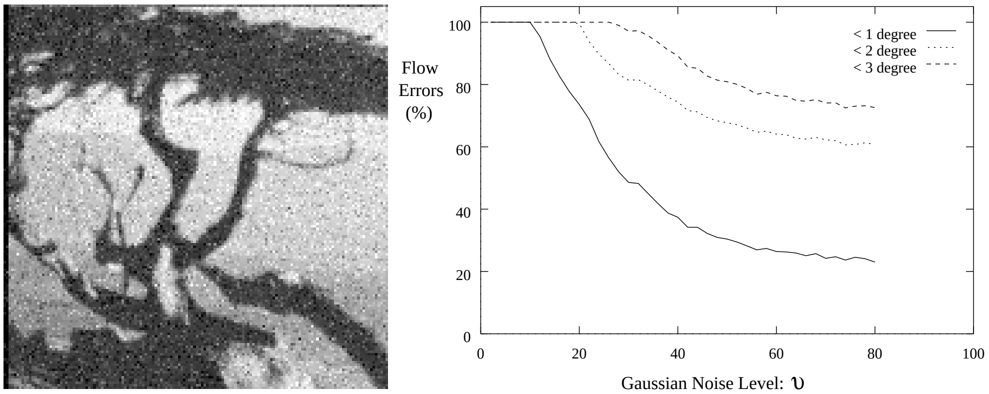


(a) (b)

Figure 3.5: **Translating Tree Sequence: robust motion estimation;** (a) estimated optical flow field; (b) the outlier mask (black stands for outliers).

Technique	Average Error	Standard Deviation	Density
Anandan [3]	4.54°	3.1°	100%
Singh [90]	1.64°	2.44°	100%
Nagel [78]	2.44°	3.06°	100%
Horn and Schunck (modified) [50]	2.02°	2.27°	100%
Uras *et al.* [96]	0.62°	0.52°	100%
Szeliski and Coughlan [92]	0.59°	N/A	100%
Wu *et al.* [107]	0.45°	N/A	100%
Fleet and Jepson [37]	0.32°	0.38°	74.5%
Lucas and Kanade [66]	0.66°	0.67°	39.8%
Giachette and Torre [42]	0.25°	0.23°	95.0%
global robust affine model	0.24°	0.05°	100%

Table 3.2: **Translating Tree Sequence: robust motion estimation;** comparison of various optical flow algorithms.



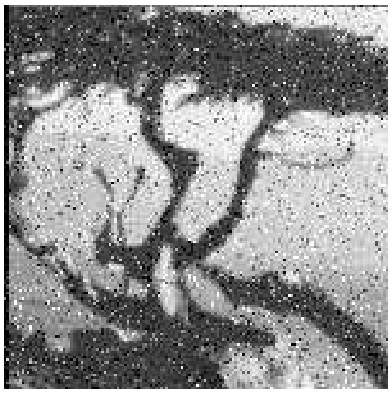
(a) (b)

Figure 3.6: **Effective of Gaussian Noise:** (a) noisy second image with Gaussian noise level $\upsilon = 20$; (b) plot of angular errors.

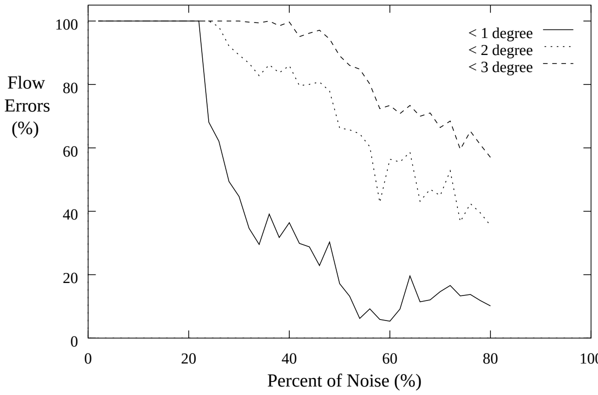
frame, while the first frame is noise-free. The algorithm was tested on three types of noise. The first is Gaussian noise with zero mean and a standard deviation υ, which varies from 0 to 80. The second is uniform random noise whose range is from -128 to 128 (the intensity values of image pixels are between 0 and 255). The signal is added to some percent of pixels, which are selected randomly. The percent varies from 2% to 80%. The third is a mixture of Gaussian noise and random white noise, where the ratio of two mixture signals is $7 : 3$. Gaussian noise is zero mean with standard deviations $\upsilon = 10$ and the uniform noise ranged from -128 to 128. The mixture signal is also added to 2% to 80% of pixels in the second image.

Figure 3.6(a) shows the noisy second frame, where the noise signal has a Gaussian distribution with zero mean and the standard deviation $\upsilon = 20$. Figure 3.6(b) shows the proportions of estimates with errors below 1, 2, and 3 degrees as a function of υ, where υ is up to 80. Comparing with the similar experiment shown in [36], the error of the robust motion estimation method starts to increase when $\upsilon > 10$, while using Fleet's phase method, errors increase almost linearly with υ. Figure 3.9 shows the mean error of the motion estimates, which is about $0.25°$ persistently when $\upsilon <= 10$.

Figure 3.7(a) shows the noisy second frame with uniform random noise added to 20% of the pixels. Figure 3.7(b) shows the proportions of estimates with errors below 1, 2, and 3 degrees as the proportion of noisy pixels varies from 2% to 80%. The algorithm



(a)

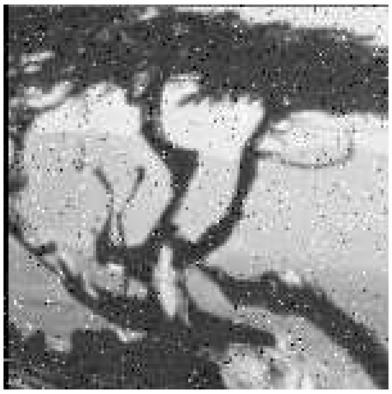


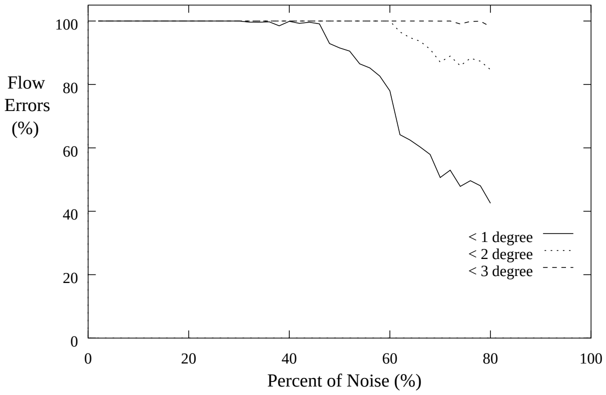
(b)

Figure 3.7: **Effective of Uniform Random Noise:** (a) noisy second image with noise added to 20% pixels; (b) plot of angular errors.

is robust to the presence of about 20% outliers. In previous implementations of robust motion estimation [87], it was shown that the $M - estimator$ can tolerate 50% outliers, which come from the non-dominant motion in the scene. In that case, the outliers are only presented in a coherent region in the image, therefore, at least 50% of image gradients are still correct. We test the algorithm with a noise signal added to randomly selected points, hence one outlier affects the gradients of four neighboring pixels. Considering this spreading effect due to the computation of image derivatives, the 20% outlier resistance in our experiments also demonstrates that the $M - estimator$ can tolerate a relatively high percentage of outliers.

Figure 3.8(a) shows the noisy second frame with additive mixture noise added to 20% of pixels. Figure 3.7(b) shows the proportions of estimates with errors below 1, 2, and 3 degrees as the proportion of noisy pixels varies from 2% to 80%. Figure 3.9 also illustrates the same result by showing plots of mean errors given different noise levels. It shows that the result with mixture noise is similar to those with random noise. Consistent motion models are estimated when less than 20% pixels have noise. The errors increase almost linearly when the noise level is larger than 20%. Figure 3.9 also illustrates the same result by showing plots of mean errors given different noise levels.





(a) (b)

Figure 3.8: **Effective of Mixture Noise:** (a) noisy second image the mixture noise added to 20% pixels; (c) plot of angular errors.

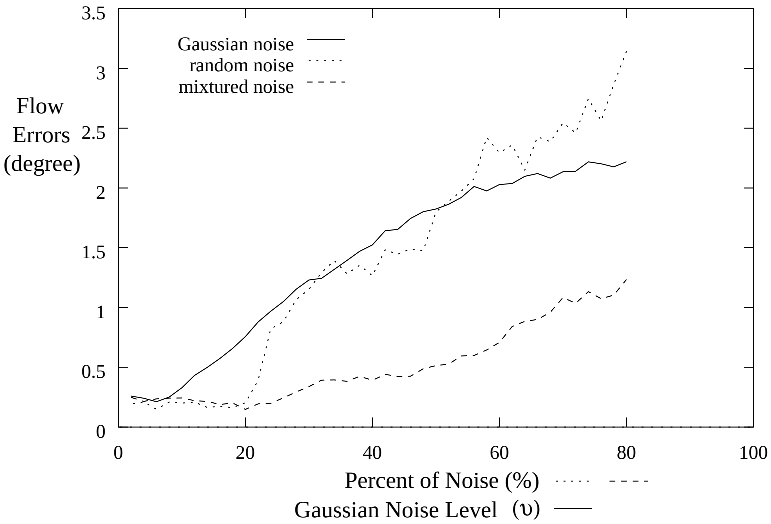
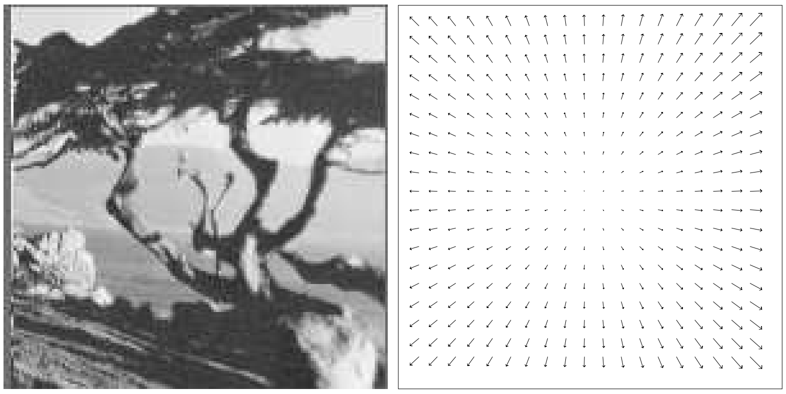


Figure 3.9: **Mean errors of three type of noise.**



(a) (b)

Figure 3.10: **Diverging Tree Sequence: ground truth;** (a) image one in the sequence; (b) optical flow field.

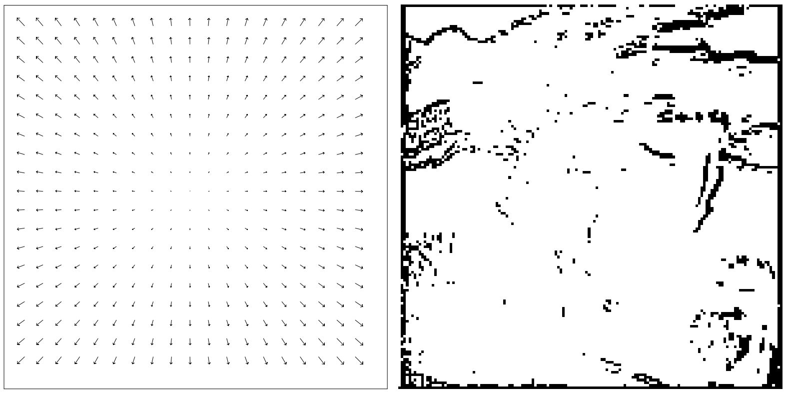
Diverging Tree Sequence

The diverging tree sequence, also created by David Fleet, is similar to the translating sequence except that the camera moves along its line of sight. The focus of expansion is at the center of the image (see Figure 3.10). The tree surface is also not perpendicular to the line of sight, thus not only image divergence and deformation but also image yaw are non-zero.

The error statistics shown in Table 3.3 indicate that the estimated motion model is not very accurate. Only 5.3% of pixels have less than $< 1°$ angular error. We assume a single affine motion in the entire image region, however the motion of the diverging tree is not well approximated by it. A more complex model, such as the planar motion model, is required. Figure 3.11(b) shows the outliers given the estimated flow field (see Figure 3.11(a)), where motion of the tree is not estimated correctly. In Section 3.2, we show how locally affine models improve the motion estimation.

Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
			$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
0.093	2.84°	1.47°	5.3%	31.5%	67.1%	88.3%	100%

Table 3.3: **Diverging Tree Sequence: robust motion estimation;** error statistics.



(a) (b)

Figure 3.11: **Diverging Tree Sequence: robust motion estimation;** (a) estimated optical flow field; (b) the outlier mask (black stands for outliers).

Yosemite Sequence

The Yosemite sequence (without clouds) was generated by Lynn Quam and provided by David Heeger. This sequence is more challenging than the previous two sequences because of the occluding edges between the mountains and at the horizon. The first frame is shown in Figure 3.12(a). Figure 3.12(b) and (c) show the true horizontal and vertical motion respectively while Figure 3.12(d) shows the optical flow field.

Yosemite sequence has a maximum speed close to 4 pixels per frame, which is about twice that of the previous two Tree sequences, thus we use a four-level Gaussian pyramid in the coarse-to-fine processing. Figure 3.13(a) shows the estimated optical flow field, which only captures the main divergent motion in the scene. Figure 3.13(b) shows the outliers given the estimated affine motion (see Figure 3.13(a)).

For the yosemite sequence, we do not compute flow errors in the sky area where no ground truth motion exists. Therefore, 71 rows are clipped from the top of the image and 5 pixels from other boundaries. The numbers used for clipping remain fixed for all the experiments of Yosemite sequence in this thesis. The error statistics are shown in Table 3.4[1]. As we expected, the estimates are poor. Since the scene contains significant depth variation, the assumption that a single affine or even planar motion model can approximate the motion of entire image is violated. Typically, these parametric models

[1]Flow errors were not computed in the sky area where no ground truth motion exists.

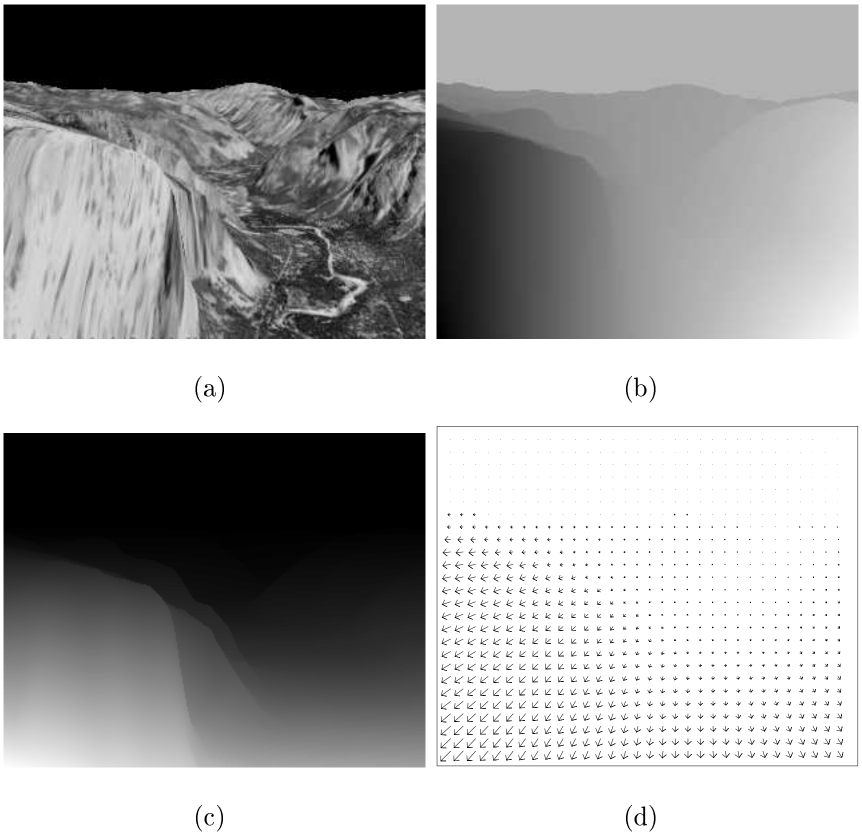


Figure 3.12: **Yosemite Sequence: the ground truth;** (a) image 11 in the sequence; (b) horizontal component of flow (leftward motion = black; rightward motion = white); (c) vertical component of flow (upward motion = black; downward motion = white); (d) optical flow field.

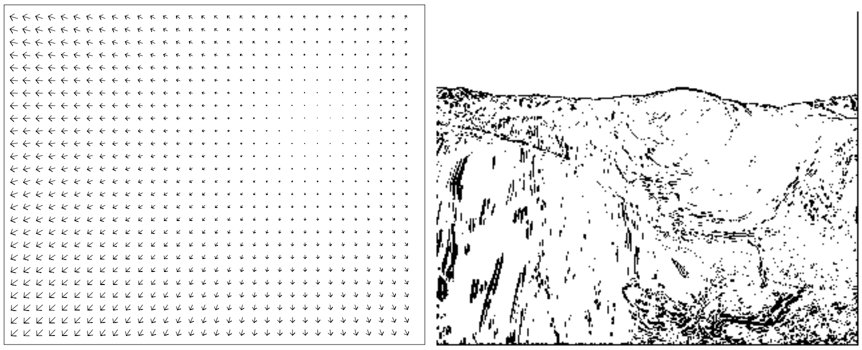
are good approximations to the image motion only locally. Thus what is need is to estimate parametric motion models in smaller image regions.

Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
			$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
0.565	8.95°	3.38°	0.5%	2.1%	4.7%	12.3%	62.9%

Table 3.4: **Yosemite Sequence: robust motion estimation;** error statistics.

3.2 Locally Affine Motion ("Bones")

In this section, we consider applying the robust motion estimation framework in small patches over the image. Following the notation described in Section 3.1, our goal is to estimate the parametric motion models of all the patches. We minimize this energy



(a) (b)

Figure 3.13: **Yosemite Sequence: robust motion estimation;** (a) estimated optical flow field; (b) the outlier mask (black stands for outliers).

function, where $E(s)$ is defined in Equation (3.9)

$$E = \sum_{s=0}^{n} E(s) = \sum_{s=0}^{n} \sum_{\mathbf{x} \in \mathcal{R}(s)} \rho(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}(s)) + I_t, \sigma(s)) \tag{3.17}$$

where $\mathbf{a}(s)$ is an affine motion model for patch s defined in Equation (3.2). Since minimizing function $\sum_{s=0}^{n} E(s)$ is equivalent to minimizing each $E(s)$, affine motion of each patch can be estimated independently as described in Section 3.1. We think of the parameterized patches as rigid pieces of "bone". The motion of each patch is approximated by one affine model. To illustrate the behavior of locally affine patches, we re-visit the Diverging Tree sequence and the Yosemite sequence below.

3.2.1 Examples

The image is divided into equally sized patches that are 32 pixels on each side. However, patches at the boundaries of the image may be larger or smaller, and may range from 16 pixels to 40 pixels one side. All the parameters used for motion estimation are unchanged as described in Section 3.1. The affine motion $\mathbf{a}(s)$, for each region s, specifies the motion of every pixel $\mathbf{x} \in \mathcal{R}(s)$ and we can use this recovered affine motion to produce a dense flow field, with a flow vector at every pixel. We apply the method to two sequences for which the robust motion estimation method using a global affine model did not work well.

Diverging Tree Sequence

Figure 3.14(a) shows the first image of Diverging Tree sequence, which is segmented into small, square, and non-overlapped patches. The horizontal and vertical components of the estimated flow are shown in Figure 3.14 (c) and (d). The pixels that were treated as outliers during the robust estimation are shown in Figure 3.14(b); these are points where the affine model was not a very good approximation to the true image motion. Figure 3.14 (e) shows the optical flow field. By visual inspection, it is clear that the estimated motion field is not as smooth as the actual flow (see Figure 3.10 (b)) and shows a block structure (only slightly except at the boundary regions). In some regions, most notably at the smaller patches near the boundaries, the estimated motion is incorrect. The brightness variations in these regions are either in one orientation only or very small, therefore the motion estimation problem is under-constrained.

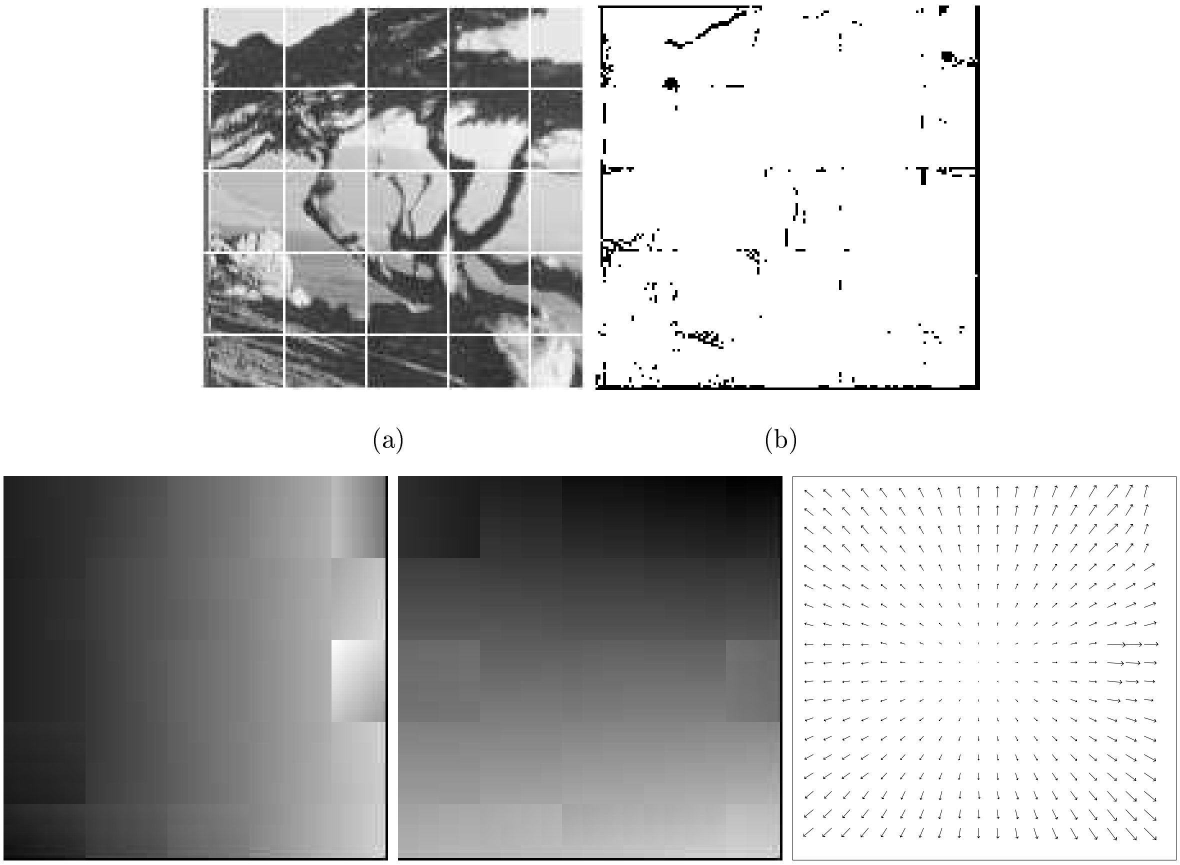
To compare the results of robust motion estimation methods using locally affine models described here and a globally affine model used in section 3.1, Table 3.5 shows the error statistics of both methods. There is a significant improvement with respect to the estimates that have errors less than 1°. This indicates that locally affine models approximate image motion better than a single affine motion in general. However the standard deviation increases, which means the flow field is not smooth, and has large errors in some regions.

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
globally affine	0.093	2.84°	1.47°	5.3%	31.5%	67.1%	88.3%	100%
locally affine	0.061	2.0°	3.12°	50.3%	73.0%	83.1%	93.0%	97.3%

Table 3.5: **Diverging Tree Sequence: locally affine motion;** error statistics.

Yosemite Sequence

Since a four-level Gaussian pyramid is used for Yosemite sequence, At the coarsest level, the size of each region is no more than 4×4 pixels which is not sufficient to reliably fit an affine flow model. Therefore, for regions at this size, we fit only a translational model (the parameters a_0 and a_3), while affine models are used at finer levels. Figure 3.15(a)



(a) (b)

(c) (d) (e)

Figure 3.14: **Diverging Tree Sequence: locally affine motion;** (a) image with segmented region shown; (b) outliers (black); (c) horizontal component of flow; (d) vertical component of flow; (e) optical flow field.

shows the first image of Yosemite sequence with segmented regions. The pixels that were treated as outliers are shown in Figure 3.15(e). The horizontal and vertical components of the estimated flow is shown in Figure 3.15 (c) and (d), which is first truncated to be in the range of $[-5, 5]$ then scaled so that the minimum value is 0 and the maximum value is 255 for display. Figure 3.15 (b) shows the vector field of optical flow for comparison with Figure 3.12 (d).

Table 3.6 shows the error statistics of both methods recovering a global affine motion and locally affine motions. There is significant improvement with respect to the estimates that have errors less than $1°$ to $5°$. Close to two thirds of the flow vectors have an angular error less than $3°$. Recall that "Average Error" refers to the mean angular error over the non-sky portion of the image.

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
globally affine	0.565	8.65°	3.55°	0.4%	1.7%	4.4%	13.0%	73.2%
locally affine	0.153	2.94°	2.58°	15.8%	44.7%	65.2%	85.8%	97.6%

Table 3.6: **Yosemite Sequence: locally affine motion;** error statistics.

Like the Diverging Tree sequence, the motion field is not very smooth and shows a clear block structure. Particularly, in the sky regions which contain no texture, any affine model can approximate the motion well. The coarse-to-fine method reduces the size of patches by first a blurring process then a sub-sampling process. The method will cause spatial aliasing at the coarse level in those patches which have no texture at the fine level (e.g., the patches at the lower part of the sky areas, which are adjacent to textured patches). Therefore, large errors will occur in coarser estimates. However, the estimates can not be corrected at the finer level due to the lack of texture in the region.

The above indicates that the motion estimation problem may be ill-conditioned in some local patches. Dividing the image into fixed regions is not sufficient to solve the problem. Therefore, Bab-Hadiashar and Suter [8] use a measure of reliability and do not produce any estimate if the result is judged to be unreliable. This scheme is not suitable if a dense flow field is required. Szeliski and Shum [93] treat the image as a set of patches, whose sizes are adaptively varied. These patches are connected in a spline-based representation that enforces smooth motion. Another solution is to keep

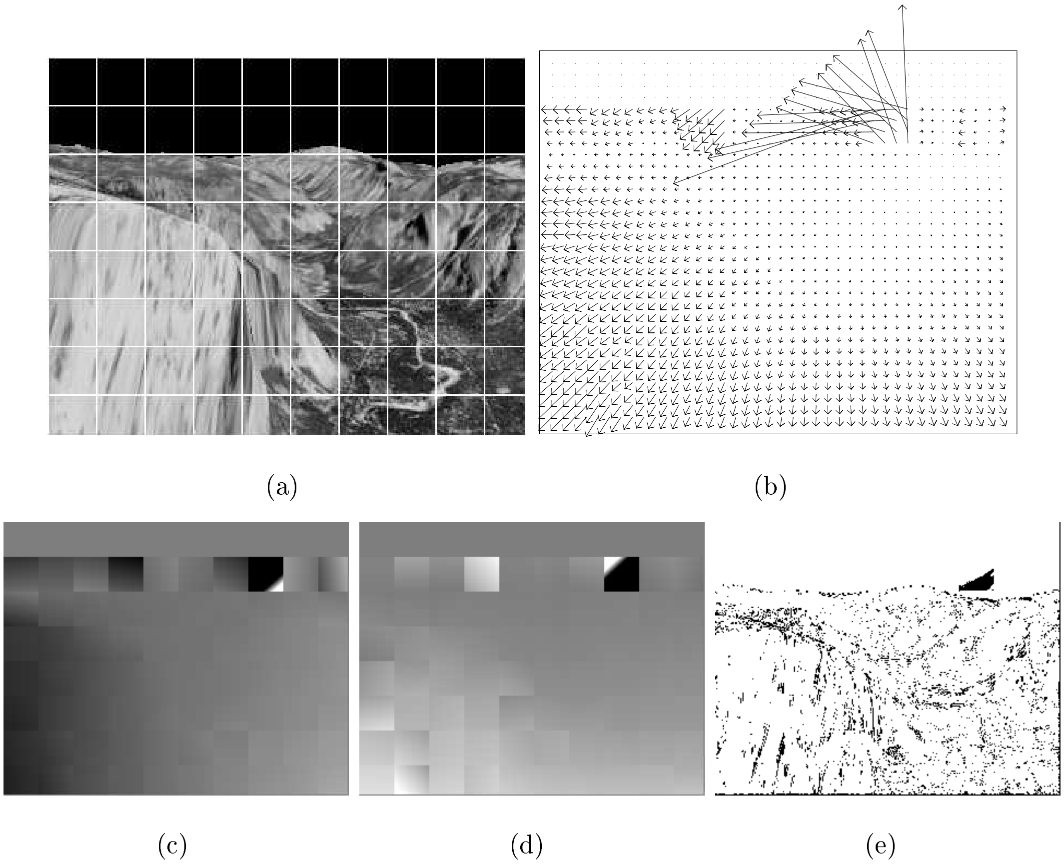


Figure 3.15: **Yosemite Sequence: locally affine motion;** (a) image with segmented region shown; (b) optical flow field. (c) horizontal component of flow; (d) vertical component of flow; (e) outliers (black).

the arbitrary fixed patches but regularize the motion estimates to make the problem well-posed [58, 73]. In the following section we will illustrate how a regularization term (skin) improves on these locally affine estimates (bones).

3.3 Regularization (Skin)

Regardless of the region size chosen for optical flow estimation, there is the possibility that the solution will be ill-conditioned due to the lack of sufficient brightness variation within the region. It is therefore useful to regularize the optical flow estimation problem by adding a spatial coherence constraint that favors solutions which are "smooth"; that is, where the spatial variation of the flow field is small. We refer to this formulation as "Skin and Bones" where the parameterized patches can be thought of as rigid pieces of "bone" that are connected by a flexible skin. In previous formulations [58], this constraint is formulated to minimize the difference between the parameters of neighboring affine patches. Here, we apply the spatial coherence constraint on the motion of the pixels at the boundaries of the patches.

3.3.1 The Smoothness Constraint

We define the Skin & Bones model by adding a spatial coherence term to the to Equation (3.17), such that the image motions at the patch boundaries are smooth.

$$\begin{aligned} E(s) \quad = \quad & \frac{1}{|\mathcal{R}(s)|}\left[\sum_{\mathbf{x}\in\mathcal{R}(\mathrm{s})} \rho(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}(s)) + I_t,\ \sigma(s)_D)\right] + \\ & \frac{\lambda}{\sum_{\mathbf{y}\in\mathcal{G}(\mathrm{s})} |\mathcal{N}(\mathrm{s},\mathbf{y})|}\left[\sum_{\mathbf{y}\in\mathcal{G}(\mathrm{s})}\sum_{\mathrm{t}\in\mathcal{N}(\mathrm{s},\mathbf{y})} \rho(\mathbf{u}(\mathbf{y}, \mathbf{a}(s)) - \mathbf{u}(\mathbf{y}, \mathbf{a}(t)), \sigma(s)_S)\right] \end{aligned} \tag{3.18}$$

where s is an image region, λ controls the relative importance of the two terms, $\mathcal{R}(s)$ and $\mathbf{a}(s)$ are the pixels and the affine parameters of region s respectively, $\mathcal{G}(s)$ is the set that contains the pixels at the boundaries of patch s, and $\mathcal{N}(s, \mathbf{y})$ are the neighboring patches connected to patch s at pixel $\mathbf{y}$. The two terms of E (data and spatial) are normalized with respect to the size of $\mathcal{R}(s), \mathcal{N}(s, \mathbf{y})$ and $\mathcal{G}(s)$ respectively and each has its own scale parameter. Note that the use of a robust error norm, ρ, allows spatial discontinuities at the boundary of the region. In this thesis, ρ is taken to be the function given in Equation (3.10).

The first term of Equation (3.18) is simply the single-layer bone from Equation (3.17). The smoothness term is formulated to minimize the difference between optical flow vectors at the boundary of the region for *all* neighboring patches. Motions that are similar will tend to reinforce each other while dissimilar motions will be ignored as outliers.

Unlike traditional parametric motion estimation schemes, the addition of the spatial coherence constraint on the affine parameters means that each step in the non-linear optimization takes into account both the optical flow constraints within the region and the parameters of the neighboring regions. This results in more accurate motion estimates and a more stable optimization problem.

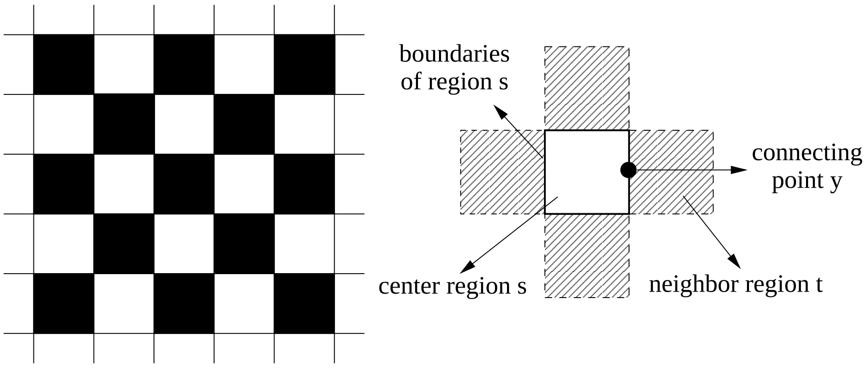


Figure 3.16: The partition of regions for first-order smoothness.

Minimization

We minimize this function using the same gradient descent scheme and continuation method described in Section 3.1.4. This involves taking derivatives of Equation (3.18) with respect to each of the affine parameters. We consider the first-order smoothness constraint, where each center region in Figure 3.16 is dependent on its four neighboring regions with a different color. Each iteration of the minimization process described in Figure 3.17 is implemented sequentially by first updating each black region in Figure 3.16 while the motion estimates in all the white regions remain fixed, then updating each white region.

The parameters that control the annealing of $\sigma(s)_D$, iterations, number of levels in the pyramid, and the size of patches were the same as those used in the previous section. The new scale parameter in the skin term, $\sigma(s)_S$, is estimated in the similar way as described in Section 3.1.3. The median absolute deviation of all difference vectors is used to estimate the scale part of $\sigma(s)_S$. For the annealing part, we use a faster decreasing rate, which is 0.9. The final value of the annealing sigma, $\hat{\sigma}(s)_S$, is also the unit 1.0. Another new parameter, λ, is taken to be $\frac{\sigma(s)_S^2}{\sigma(s)_D^2}10.0$. All these values remain fixed in all remaining experiments in this chapter.

The iterative update equations for minimizing $E(s)$ at step $n + 1$ are defined in Equation (3.13). Considering the regularization term in the objective function (3.18),

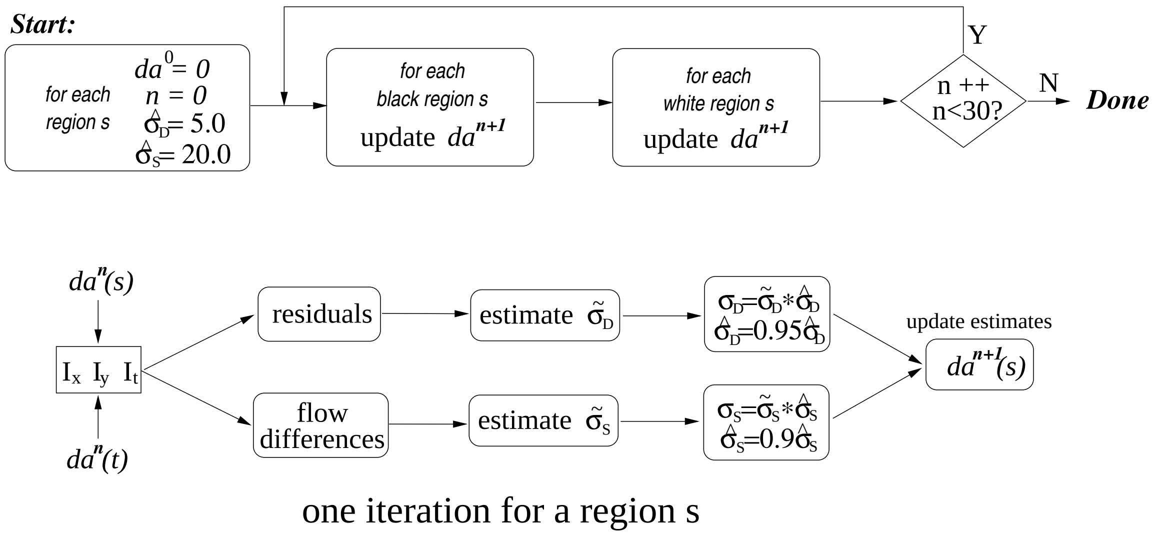


Figure 3.17: The iterative minimization process

the detailed update equation for each affine parameter is:

$$
\begin{aligned}
a_i^{(n+1)} &= a_i^{(n)} - \frac{\omega}{T(a_i)}\left[\frac{1}{|\mathcal{R}|}\sum_{\mathbf{x}\in\mathcal{R}}\frac{2r\sigma_D^2 P_{\mathbf{x},i}}{(\sigma_D^2+r^2)^2} + \frac{1}{\sum_{\mathbf{y}}|\mathcal{N}(s,\mathbf{y})|}\sum_{\mathbf{y},t}\frac{2d\sigma_S^2 H_{\mathbf{x},i}}{(\sigma_S^2+d^2)^2}\right], \quad (3.19)\\
T(a_i) &= \frac{2K_i}{\sigma_D^2|\mathcal{R}|} + \frac{2\lambda\sum_{\mathbf{y},t}H_{\mathbf{x},i}^2}{\sigma_S^2\sum_{\mathbf{y}}|\mathcal{N}(s,\mathbf{y})|},\\
H_{\mathbf{x},0} &= H_{\mathbf{x},3} = 1, \quad H_{\mathbf{x},1} = H_{\mathbf{x},4} = x, \quad H_{\mathbf{x},2} = H_{\mathbf{x},5} = y,
\end{aligned}
$$

where d denotes the difference between optical flow vectors at a boundary point $\mathbf{y}$ generated by estimated affine motions of region s and t. $P_{\mathbf{x},i}$ and K_i are defined in (3.15) and (3.14) respectively. The dependence on region s in Equation (3.19) is omitted.

3.3.2 Examples: Synthetic Sequences

We revisit the Diverging Tree sequence and the Yosemite sequence using the “Skin and Bones” model.

Diverging Tree Sequence

To illustrate the effect of regularization we add skin to the Diverging Tree sequence example from the previous section. The horizontal and vertical components of the estimated flow are shown in Figure 3.18 (a) and (b). The pixels that were treated as outliers during the robust estimation are shown in Figure 3.18(c). Figure 3.18 (d) shows the optical flow field. By visual inspection, it is clear that the estimated motion field is much smoother than the flow field estimated without the "skin" term (see Figure 3.14 (c)). The unstable patches near the boundaries in Figure 3.14 are gone.

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Bones	0.061	2.0°	3.12°	50.3%	73.0%	83.1%	93.0%	97.3%
Skin&Bones	0.023	0.81°	0.72°	73.9%	95.7%	98.2%	99.4%	100%

Table 3.7: **Diverging Tree Sequence: Skin&Bones;** error statistics.

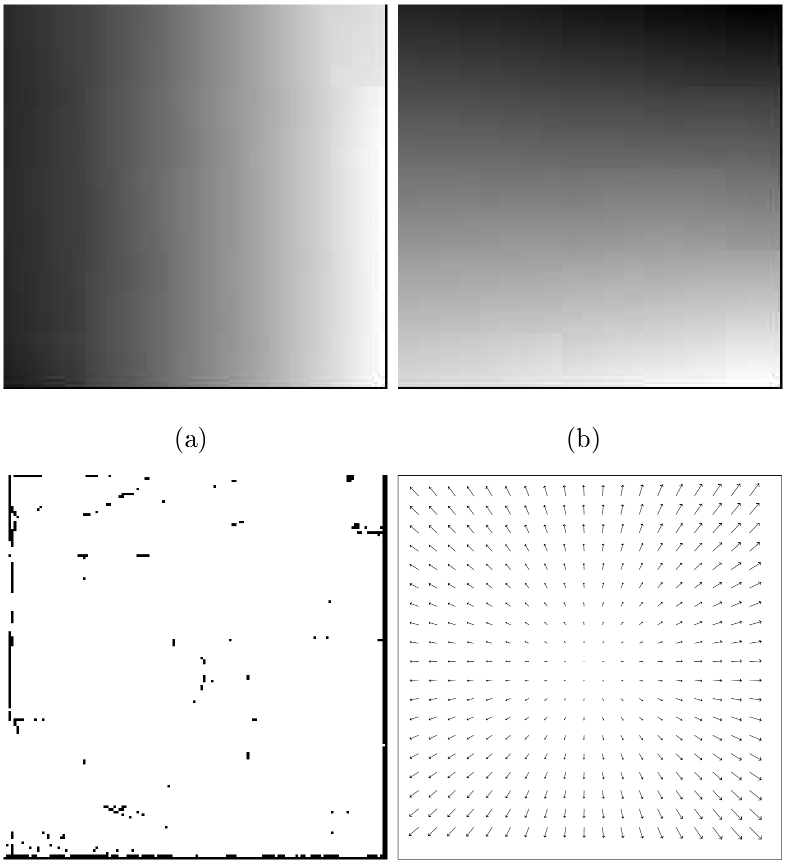
The improvement is also compared quantitatively in Table 3.5. From the table we see that only about a quarter of pixels have more than 1° angular error, and the addition of "skin" improves the average angular error by 60%. Another notable improvement is that there are no pixels with large error, which indicates that the "Skin and Bones" model results in a more stable optimization procedure. The results are also compared with other published results [10] in Table 3.8, which shows that the "Skin and Bones" method generates the most accurate motion estimates.

Yosemite Sequence

We also apply the "Skin and Bones" method to Yosemite sequence. The recovered optical flow using Equation (3.18) is shown in Figure 3.19. Comparing the results to those in Figure 3.15 one can see that the results are much smoother, and that the estimates in the sky area are more stable. The improvement also is compared quantitatively in Table

Technique	Average Error	Standard Deviation	Density
Anandan [3]	7.64°	4.96°	100%
Singh [90]	8.6°	4.78°	100%
Nagel [78]	2.94°	3.23°	100%
Horn and Schunck (modified) [50]	2.55°	3.67°	100%
Uras *et al.* [96]	4.64°	3.48°	100%
Szeliski and Coughlan [92]	0.98°	0.74°	100%
Wu *et al.* [107]	1.33°	N/A	100%
Fleet and Jepson [37]	0.99°	0.78°	61.0%
Lucas and Kanade [66]	1.94°	2.06°	48.2%
Giachette and Torre [42]	2.07°	1.37°	95.0%
single-layer Skin&Bones	0.81°	0.72°	100%

Table 3.8: **Diverging Tree Sequence:** comparison of various optical flow algorithms.



(a) (b)

(c) (d)

Figure 3.18: **Diverging Tree Sequence: Skin & Bones:** (a) horizontal component of flow; (b) vertical component of flow; (c) outliers (black); (d) flow field.

3.9. From the table see that the addition of "skin" improves the average angular error by 28%, and that close to one third of pixels have less than 1° angular error (twice as much as the result without the skin term).

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Bones	0.153	2.94°	2.58°	15.8%	44.7%	65.2%	85.8%	97.6%
Sin&Bones	0.098	2.11°	1.84°	30.5%	61.7%	78.7%	92.8%	99.4%

Table 3.9: **Yosemite Sequence: Skin&Bones;** error statistics.

The results of the Skin & Bones approach are compared with other published results for the Yosemite sequence in Table 3.10 [10]. In [10], when the sky is omitted, the errors for the Lucas and Kanade [66] and Fleet and Jepson [37] methods improve to 3.37° and 2.97° respectively, although the density remains low. The accuracy of the other approaches in [10] might also be expected to improve in accuracy by approximately 25% if the sky is ignored but this is still well below the accuracy of the Skin & Bones model which also provides 100% density (not counting the sky). Comparing to other recent methods proposed by Black and Jepson [18], and Bab-Hadiashar and Suter [8][2], the performance of the "Skin and Bones" method is also good. In [18], Black and Jepson perform a similar parametrized fit, but do so in regions obtained by segmenting the brightness images. They allow deformations from the fitted motions using a robust regularization scheme in which the motion of the patches is treated as a prior. In [58], we allowed similar local deformations from the Skin & Bones fit, the average angular error decreased to 1.82° with as standard deviation of 1.58° and 100% density.

3.3.3 Examples: Real Image Sequences

The "Skin and Bones" method is applied to three real image sequences in this section.

[2]The numbers, which are different from those listed in their original paper, are calculated using our error computation program after running their motion estimation algorithm. We will explain it in greater details in Section 5.4. The best results of the Yosemite sequence with clouds shown in [8] have 2.11° average angular error and 1.75° standard deviation with 58.2% density.

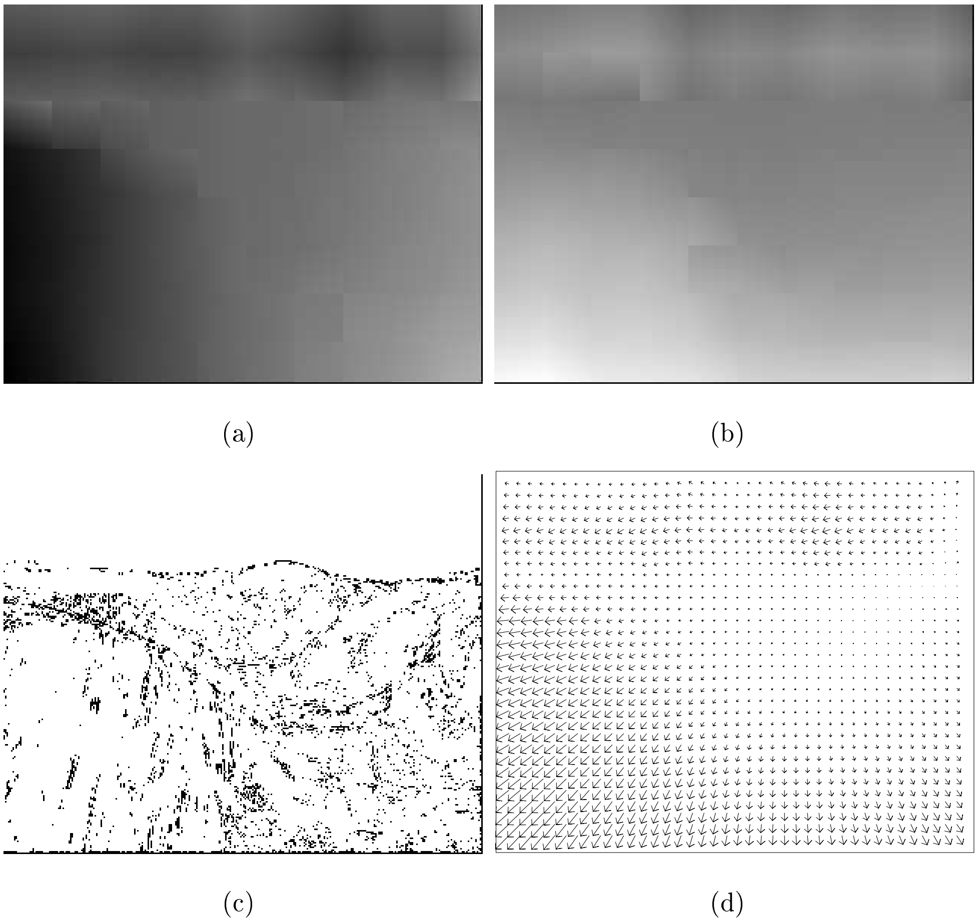


Figure 3.19: **Yosemite Sequence: Skin & Bones;** (a) horizontal component of flow; (b) vertical component of flow; (c) outliers (black); (d) flow field.

Technique	Average Error	Standard Deviation	Density
Anandan [3]	15.84°	13.46°	100%
Singh [90]	13.16°	12.07°	100%
Nagel [78]	11.71°	10.59°	100%
Horn and Schunck (modified) [50]	11.26°	16.41°	100%
Uras *et al.* [96]	10.44°	15.00°	100%
Fleet and Jepson [37]	4.29°	11.24°	34.1%
Lucas and Kanade [66]	4.10°	9.58°	35.1%
Weber and Malik [102]	3.42°	5.35°	45.2%
Giachette and Torre [42]	2.82°	6.98°	70.9%
Szeliski and Coughlan [92]	3.09°	7.59°	39.6%
Szeliski and Shum [93]	3.00°	7.08°	39.4%
Wu *et al.* [107]	3.54°	N/A	100%
Memin and Perez [73]	4.75°	6.89°	100%
Black and Anandan [16] *	4.47°	3.90°	100%
Black [14] *	3.52°	3.25°	100%
Black and Jepson [18] *	2.29°	2.25°	100%
Bab-Hadiashar and Suter [8] *	2.51°	2.58°	100%
Skin & Bones *	2.11°	1.84°	100%

Table 3.10: **Yosemite Sequence:** comparison of various optical flow algorithms. The "*" indicates those results computed without the sky;.

Marbled Block Sequence

The sequence, which contains four columns and one moving marbled block, was created and used by Otte and Nagel in [82][3]. It is a real image sequence prepared with a camera mounted on the moving arm of a robot. The camera moves with pure translation towards the scene. The marbled light block translates to the left, while other objects are stationary. The camera is calibrated, and the true motion field is provided by the author. Figure 3.20(a) to (c) show the first frame of the sequence, the pixels that have no motion information (due to occlusion and disocclusion), and the ground truth vector field.

For Marbled Block sequence, very few results of angular errors have been published. In Table 3.11, we show the performance of the "Skin and Bones" method, the locally affine motion method (Bones), and two of the best dense optical flow methods proposed by Black and Anandan [16], and Bab-Hadiashar and Suter [8]. From the table, we see that the addition of skin improves the accuracy by 15.7%, and that the performance

[3]The sequence is available online.

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Black & Anandan	0.145	4.04°	4.38°	9.4%	29.0%	56.1%	85.0%	90.4%
Bab-Hadiashar [8]	0.123	3.36°	4.28°	2.9%	30.9%	78.6%	90.6%	94.5%
Bones	0.147	4.08°	4.96°	10.5%	33.8%	60.6%	82.4%	90.9%
Skin&Bones	0.130	3.44°	4.00°	11.8%	37.4%	67.7%	88.2%	92.6%

Table 3.11: **Marbled Block Sequence: Skin&Bones;** error statistics.

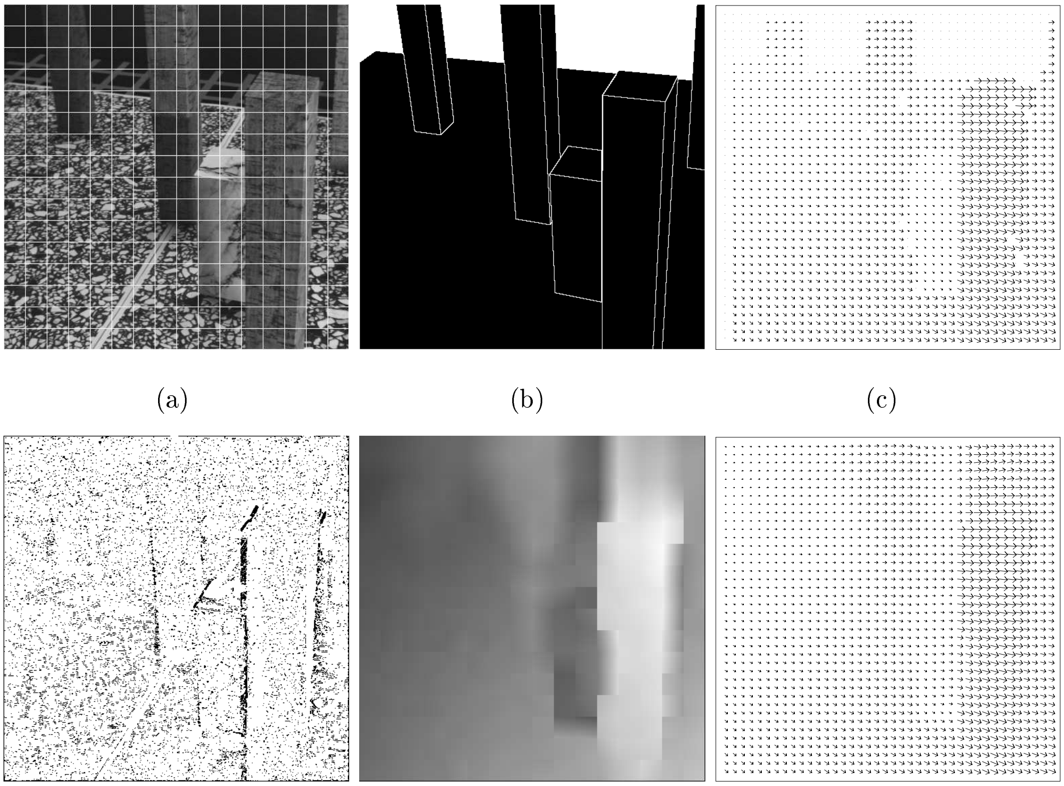
Technique	Average Error	Percent of vectors used
Werkhoven & Koenderink [106]	0.369	43.6%
Otte & Nagel: RC [82]	0.127	50.5%
Otte & Nagel: SC [82]	0.128	51.3%
Campani & Verri: [25]	0.107	51.4%
Black & Anandan [16]	0.145	100%
Bab-Hadiashar [8]	0.123	100%
Bones	0.147	100%
Skin & Bones	0.13	100%

Table 3.12: **Marbled Block Sequence:** comparison of various optical flow algorithms.

of the "Skin and Bones" method is slightly better than those of Black and Anandan's method, or comparable with Bab-Hadiashar and Suter's method. Comparing with the Yosemite sequence, the overall performance of the algorithm dropped because of several reasons. First, there are significant motion discontinuities presented in the scene. The single-layer "Skin and Bones" model has a limitation in handling multiple motions. We will discuss this limitation in details in Section 3.5. Second, the true flow provided also contains errors[4]. The true flow is computed from the world point coordinates of the 3D scene with a precision of $1/10mm$, furthermore, the calibrated camera is not very accurate either.

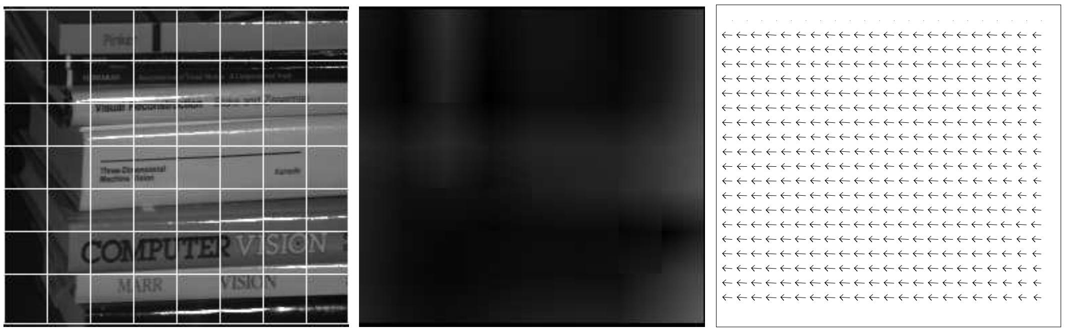
Otte and Nagel [82] used the absolute magnitude of difference vectors between the true image velocity and the estimated one as the error measure. Table 3.12 shows the quantitative comparison between our methods and other different algorithms. Although the results by Otte and Nagel [82], and by Campani and Verri [25] are more accurate, they used only the better half of the motion constraints.

[4]Otte mentioned this in the calibration data file: "Although the trajectory for the robot was given as a pure translational movement, the obtained calibration data shows, that the resulting trajectory differs slightly from the given one."



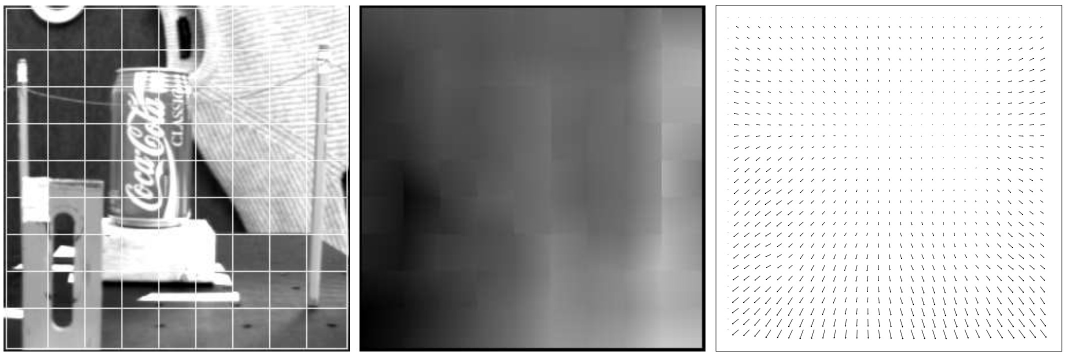
(a) (b) (c) (d) (e) (f)

Figure 3.20: **Marbled Block Sequence: Skin & Bones:** (a) image with segmented regions shown; (b) pixels without true data (white); (c) true flow field; (d) outliers (black); (e) horizontal component of flow; (f) estimated flow field.



(a) (b) (c)

Figure 3.21: **Book Sequence: Skin & Bones:** (a) image with segmented regions shown; (b) horizontal component of flow; (c) flow field.



(a) (b) (c)

Figure 3.22: **NASA Coke Sequence: Skin & Bones:** (a) image with segmented regions shown; (b) horizontal component of flow; (c) flow field.

Book Sequence

The book sequence shows a pile of books, while the camera is translating to the right. The sequence is challenging because of the specular reflections presented in the scene. Figure 3.21 shows the estimated results, and the translational motion in the scene is recovered correctly.

NASA Coke Sequence

The NASA Coke sequence is primarily dilational, where the camera moves toward the Coke can near the center of the image. Figure 3.22 shows the estimated flow field by the "Skin and Bones" method, which is smooth and recovers the correct dilational motion in the scene.

3.4 Tiling the Image

Recall that the area-based optical flow approaches suffer from the *generalized aperture problem* [55], which refers to the dilemma surrounding the choice for the appropriate size of region. We assume the spatial variation of the motion within a patch can be modeled by a single affine model. Generally, the assumption is valid only locally. Hence, the patch should be small. On the other hand, a small region may cause the motion estimation problem to be under-constrained. Firstly, as real images are noisy and derivative operators can enhance the noise, a large region is needed such that the process is insensitive to noise. Secondly, the region must be taken to be sufficiently large to include several constraints having different orientations. For these two reasons, the patch should be large, which is in contradiction with the previous requirement. This dilemma is common to all optical flow techniques. In this section, we discuss the problem caused by tiling the image. The single-layer "Skin and Bones" method is applied to synthetic image sequences with different sizes of patches, and the results are compared quantitatively.

We use the Diverging Tree sequence and the Yosemite sequence to demonstrate the effect of tiling the image with different grids. In these experiments, the size of patches varies from 16 pixels to 64 pixels at one side. Figure 3.23 and Figure 3.24 plot the means and standard deviations of angular errors of the Diverging Tree sequence and the Yosemite sequence respectively. The results shown in both figures are consistent. Patches that are smaller than 24×24 will result in unsteady estimates. Between 32×32 and 42×42, the best performance is observed in both sequences. When the region gets larger than 46×46, the errors increase gradually. In summary, 32×32 is an appropriate aperture size in general. However, real image sequences often contain various amounts of noise, thus a larger aperture size, such as 42×42, may result in more stable estimates for some real image sequences.

3.5 Limitations of the Single-Layer Model

The "Skin and Bones" model exploits the accuracy of area-based regression techniques locally and does so reliably through the use of a regularizing term. When the affine flow

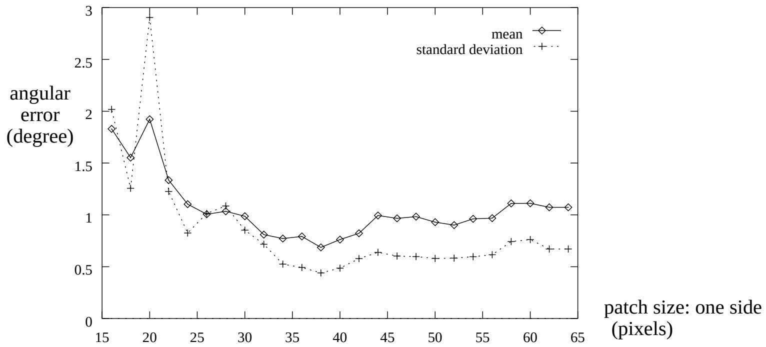


Figure 3.23: **Tiling the Image:** Diverging Tree sequence errors.

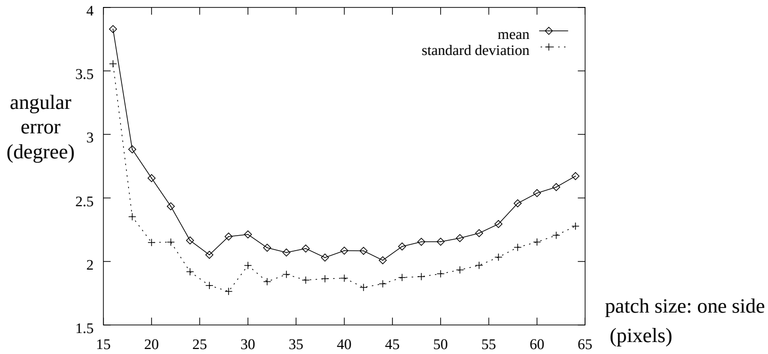


Figure 3.24: **Tiling the Image:** Yosemite sequence errors.

model is a reasonable approximation for the motion in a region, minimizing Equation (3.18) results in very accurate motion estimates. In practice however, flow fields are rarely smoothly varying but rather, typically contain discontinuities.

Consider the well known Flower Garden sequence shown in Figure 3.25 (a). The scene is static but the camera motion induces parallax motion on the image plane due to the different depths in the scene. The 32×32 pixel regions in the figure span surfaces at a number of depths. In this case the robust motion estimation technique used in the "Skin and Bones" model will tend to recover the dominant motion in a region. This can be seen in the horizontal flow estimates in Figure 3.25 (a) (there is very little vertical motion). Notice that in regions which span the tree boundary, one of two things occurs. In some cases, the algorithm chooses one of the two motions in the region. Where this occurs, the other motion is treated as an outlier and pixels associated with that motion appear as black in Figure 3.25 (b). In the other cases, the algorithm attempts to fit both the foreground and background regions. For these cases, the motion constraints from the

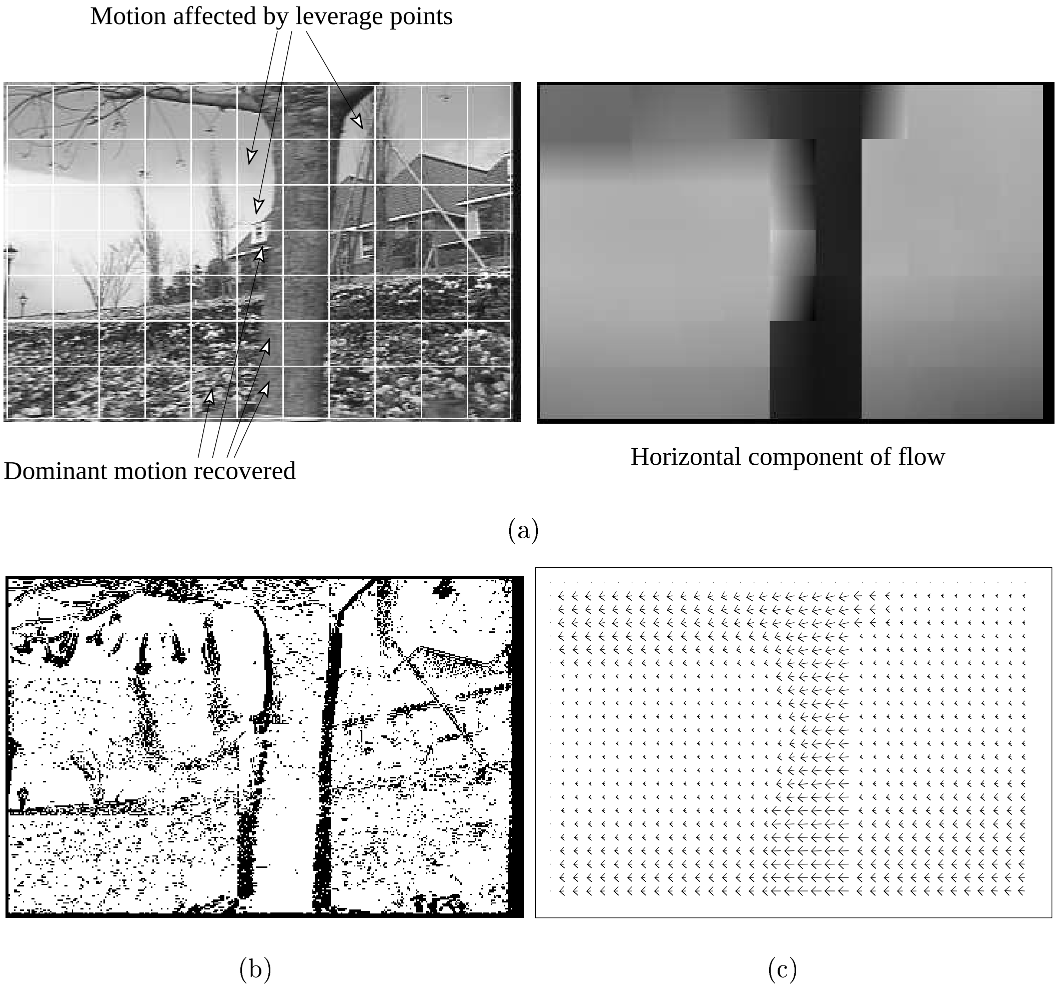


Figure 3.25: **Flower Garden Sequence: Skin & Bones:** (a) image with segmented regions shown, and the horizontal component of the flow; (b) outliers (black); (d) flow field.

non-dominant motions can be thought of as leverage points that pull the solution away from the dominant motion (Figure 3.25 (a)).

Similar behavior can be seen in the Hamburg taxi sequence shown in Figure 3.26 (a). In this sequence, the camera is static but several objects (three vehicles and a pedestrian) are moving independently in the scene. Again, the 32×32 patches in the figure span regions containing both moving vehicles and the still background. The motion of one of these surfaces will dominate causing a violation of the optical flow constraint equation for the other motion. Figure 3.26(c) shows the horizontal component of the motion field. The background motion is recovered in all but four patches. In patch 1 and 3 (see Figure 3.26(a)), the motions of moving vehicles are recovered, which also affect the estimated

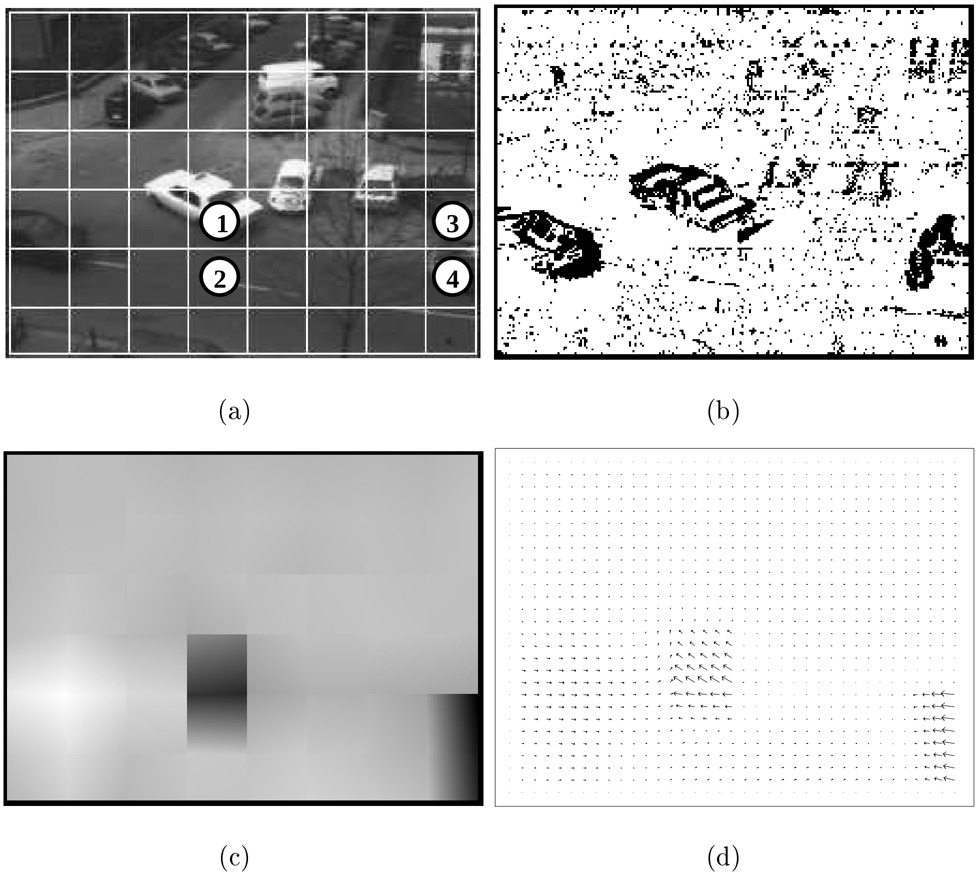


Figure 3.26: **Hamburg Taxi Sequence: Skin & Bones** (a) image with segmented regions shown; (b) outliers (black); (c) horizontal component of flow; (d) flow field.

motion of the patches that are directly underneath them.

To cope with common situations like this, tiling the image with a finer grid is not a suitable solution, since smaller patches may result in unstable motion estimation. Furthermore, regardless of the region size chosen for optical flow estimation, there is the possibility that multiple motions are present in the region (e.g., when there is a transparent motion in the scene). Thus the single motion assumption is often invalid. What is required is an extension of the model to allow multiple motions within a region as described in the following chapters.

Chapter 4

Mixtures of Locally Affine Motions

The main problem addressed in this chapter is that of computing multiple affine motion models simultaneously. In order to allow multiple models in the description of image motion, we model the likelihood function for change in intensity of a pixel, conditioned on the motion parameters, as an additive mixture of some density functions. This is called *mixture models*. We present an approach for motion estimation based on the maximum likelihood estimation of mixture models.

Section 4.1 reviews the mixture likelihood approach to clustering [72], and the previous implementations of mixture models in the field of motion estimation.

Section 4.2 presents a layered representation of locally affine motions based on mixture models. We think of multiple motions as corresponding to *layers*, which contain a single consistent parametric motion. The number of layers presented in each patch is given in advance. In addition, an outlier layer is used to identify the atypical constraints which resulte from noise or occlusion/disocclusion boundaries. Our formulation is solved by using the Expectation-Maximization(EM) algorithm.

Section 4.3 describes the problem of spatial coherence within a patch. In the formulation described in Section 4.2, pixels are assigned to models based on their residuals only. Therefore, estimated flow is often "speckled". We assume that neighboring points are likely to be from the same object, which is a smoothness constraint on the ownership weights. A smoothness prior on the ownership weights is added to the formulation in Section 4.3 to consider this spatial coherence constraint within patches.

Section 4.4 addresses the issue of how the number of layers affects the accuracy of the recovered flow. Experimental results are shown with two, three, and four motion layers estimated in each patch. In Chapter 6, we also present an approach to automatically select the appropriate number of layers using a Minimum Description Length principle.

Finally, Section 4.6 demonstrates the experimental results. Note that we estimate multiple motions within each patch *independently* in this chapter. The regularization term will be included in the formula in the next chapter.

4.1 Mixture Models and EM Algorithm

In the mixture likelihood approach, observations are assumed to be from a mixture of an given number of populations or groups. The likelihood is formed in terms of the mixture of densities, given density functions of each group. The probability of selecting group k is obtained by estimating posterior probabilities of group ownership. In this section, we first review the likelihood estimation for finite mixture models, then discuss recent achievements of using mixture models in motion estimation.

4.1.1 Mixture Likelihood Approach

Following the notation used in [72], a superpopulation G is a mixture of a finite number, say g, of populations $G_1, \cdots, G_g$ in some proportions $\pi_1, \cdots, \pi_g$, where

$$\sum_{i=1}^{g} \pi_i = 1 \qquad lrm \qquad \pi_i \geq 0 \tag{4.1}$$

The probability density function (pdf) of an observation $\mathbf{x}$[1] in G can therefore be represented in the finite mixture form,

$$f(\mathbf{x}; \phi) = \sum_{i=1}^{g} \pi_i f_i(\mathbf{x}; \mathbf{a}_i), \tag{4.2}$$

where $f_i(\mathbf{x}; \mathbf{a}_i)$ is the pdf corresponding to G_i, and $\phi = (\pi_1, \cdots, \pi_g, \mathbf{a}_1, \cdots, \mathbf{a}_g)$ denotes the vector of all unknown parameters. Assume $\mathbf{x}_1, \cdots, \mathbf{x}_n$ are the observed values of n independently and identically distributed random variables with a common distribution function, an estimate of ϕ can be obtained as a solution of the likelihood equation,

$$\partial L(\phi)/\partial \phi = 0, \tag{4.3}$$

[1]Boldface letters are used to represent a vector of $p-$dimensions in this thesis.

where $L(\phi)$ denotes the log likelihood. The posterior probability that $\mathbf{x}_j$ belongs to G_i is given by

$$\tau_{ij} = \pi_i f_i(\mathbf{x}_j; \mathbf{a}_i) / \sum_{t=1}^{g} \pi_t f_t(\mathbf{x}_j; \mathbf{a}_t) \tag{4.4}$$

Here the quantities τ_{ij} will be called "ownership probabilities". At a local extrema, it can be shown that the parameters ϕ must satisfy

$$\pi_i = \frac{1}{n} \sum_{j=1}^{n} \tau_{ij} \tag{4.5}$$

$$\sum_{i=1}^{g} \sum_{j=1}^{n} \tau_{ij} \partial \log f_i(\mathbf{x}_j; \mathbf{a}_i) / \partial \mathbf{a}_i = 0. \tag{4.6}$$

Equation (4.5) and (4.6) are used to solve the likelihood equation for mixture models (Equation (4.3)) with specific component densities. An iterative computation of the solution using the EM algorithm was suggested in [33]. The EM algorithm proceeds iteratively in two steps, E (expectation) step and M (maximization) step. Using some initial value for ϕ, say $\phi^{(0)} = (\pi_i^{(0)}, \mathbf{a}_i^{(0)}), i = 1, \cdots, g$, the E step requires the calculation of the expectation of ownership probabilities $\tau_{ij}^{(1)}$ using Equation (4.4) with π_i and $\mathbf{a}_i$ replaced by $\pi_i^{(0)}$ and $\mathbf{a}_i^{(0)}$. On the M step first the time through, updated $\phi^{(1)}$ is obtained by solving Equation (4.5) and (4.6) with τ_{ij} replaced by $\tau_{ij}^{(1)}$. The solution to the M step may exist in closed form, such as when the component densities of the mixture are taken to be normal. The E and M steps are alternated repeatedly.

The EM algorithm is guaranteed to increase the log likelihood with each iteration. However, the convergence may be quite slow, and the situation will be exacerbated by a poor choice of $\phi^{(0)}$. Also the EM algorithm does not guarantee convergence to the global maximum when there are multiple maxima, and in this case, the estimate obtained depends upon the initial value.

The E and M steps bear a resemblance to a motion segmentation framework proposed by Hsu *et al.* [51]. They observed that many motion segmentation algorithms can be characterized as iterating the two steps of *segmentation* and *motion modeling*. In the segmentation step, regions are assigned to models by measuring deviation from prediction, and in motion modeling step, motion is estimated using the assignment of the regions. However, the segmentation achieved by the EM algorithm used for mixture estimation is "soft", for a pixel can be assigned in varying proportions to multiple populations.

4.1.2 Related Work

Mixtures of distributions have been used extensively as models in the field of cluster analysis where data can be viewed as arising from two or more populations mixed in varying proportions. Jepson and Black [55] were the first to use a mixture model formulation for the problem of optical flow computation in the presence of multiple motions. In their approach, the motion constraint at a given pixel position is defined to be modeled by a probability function of Gaussian mixtures of two constant velocity models. By introducing an outlier process, their approach can also cope with outliers, which in the mixture model case can be viewed as data points that are atypical of all components in the mixture. Since we use a straightforward extension of their approach, we will review their formulation briefly in the following.

Jepson and Black [55] assumed the flow in an image region can be treated as constant velocity plus noise. They allowed two different constant velocity layers within a region. Each layer was modeled by a Gaussian distribution

$$f_i(\mathbf{x}, \mathbf{c}(\mathbf{x}); \mathbf{a}_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp - \frac{d^2(\mathbf{c}(\mathbf{x}), \mathbf{a}_i(\mathbf{x}))}{2\sigma_i^2}$$

where $\mathbf{c}(\mathbf{x})$ is the observed motion constraint at pixel $\mathbf{x}$, $d^2(\mathbf{c}(\mathbf{x}), \mathbf{a}_i(\mathbf{x}))$ is the sine of angular error between the motion constraint $\mathbf{c}(\mathbf{x})$ and the image velocity $\mathbf{a}_i(\mathbf{x})$, σ_i is an estimate for the variance of the angular errors. In addition, they also attempted to identify outliers in an outlier distribution. Note that in their formulation, both the variance of the Gaussian components and the outlier distribution were chosen as pre-defined constants. Given this specification of f_i, they solved the M step in closed form. Recall this involved finding a solution of Equation (4.6). In the E step, ownership probabilities τ_{ij} were computed according to the current values of the mixture probabilities π_i, and the current estimates of motions $\mathbf{a}_i$. In the M step, $\mathbf{a}_i$ and π_i were updated. This entire EM-iteration was repeated until the change in the parameters is sufficiently small.

Similarly, Ayer and Sawhney [4] modeled the likelihood function as an additive mixtures of Gaussian densities, where the motion of each layer was assumed to be affine. They also estimated the variances in the E step. The motion parameters for each layer were estimated using a robust framework in the M step. However, they did not model the outliers as a layer in the mixture models. Instead, outliers were detected and removed according to a simple test. Moreover, they also addressed the issue of how to choose the appropriate number of layers by using a Minimum Description Length encoding principle.

Weiss and Adelson [105] used mixture models and the EM algorithm to estimate the layered parametric motions as well. They started with a sufficiently large number of layers, and chose different variances to control the actual models that would be recovered. In addition, a spatial coherence constraint was added to the weights that assigned pixels to layers. This encourages layers to have spatially coherent support.

Yuille *et al.* [110] also exploited robust statistics, formulated the problem in a statistical physics framework, and use an EM algorithm with deterministic annealing to solve for the motion of each layer.

More recently, mixture models have been used to estimated image motion in more complex scenes. Weiss [103] proposed a layered mixture model in which the motion in each layer is modeled with a smooth flow field. The algorithm was based on nonparametric mixture estimation. It was able to segment higher order flow fields while avoiding over-fitting. Black *et al.* [17] proposed a framework to recover the "appearance changes" in a sequence as a mixture of different causes. Appearance changes, such as form change, illumination change, iconic change, and specular change often occur together in the scene, thus they employed the probabilistic mixture model formulation to estimate the various types of appearance change and to perform a soft assignment of pixels to causes. They used the EM algorithm to iteratively compute maximum likelihood estimates for the unknown parameters and the posterior probabilities that pixels at time t were explained by each of the causes.

4.2 Mixtures of Robust Bones

We deal with several motions within a single region using a straightforward extension of the mixture model approach described in [55]. That is, for a given image region we model the flow using, $\mathcal{L}$, affine layers (below we start with a fixed number of layers and then show how this number can be estimated in Chapter 6). In addition, to accommodate data which cannot be accounted for by any of these layers, we include an outlier process.

To estimate motion in layers, we must do two things: (1) assign pixels (i.e. the optical flow constraint at a pixel) to one of the layers and, (2) estimate the motion of each layer. Say that the flow constraint at any given pixel $\mathbf{x}$ is assigned to the i^{th} layer with an *ownership weight*, $m_i(\mathbf{x}; \sigma)$. Let the estimated motion parameters for layer i be $\mathbf{a}_i$. If we knew the assignment, $m_i(\mathbf{x}, \sigma)$, of constraints to layers, then the motion of a layer could be computed using weighted least squares estimation:

$$E(\mathbf{a}_i) = \sum_{\mathbf{x} \in \mathcal{R}} m_i(\mathbf{x}, \sigma)(\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t)^2 \tag{4.7}$$

where we minimize E with respect the the affine parameters $\mathbf{a}_i$, for layer i while holding the weights $m_i(\mathbf{x}, \sigma)$ fixed.

Similarly, if we knew the motion, $\mathbf{a}_i$ of each layer, we could decide how likely each constraint is to have come from each of the layers and use this likelihood to determine the weights $m_i(\mathbf{x}, \sigma)$. Our goal here is to solve both these problems: computing the affine parameters for each layer, $\mathbf{a}_i$ for $1 \leq i \leq \mathcal{L}$, and the appropriate layer assignment weights, $m_i(\mathbf{x}, \sigma)$ for $1 \leq i \leq (\mathcal{L} + 1)$. Here we denote the outliers as layer $\mathcal{L} + 1$.

The estimation process we use is a variant of the Expectation-Maximization (EM) algorithm which iterates between these two problems. The E-step involves the estimation of the ownership weights given the current estimate of the layer motions. The M-step uses the ownership weights to solve for the affine parameters of each layer using Equation (4.7). We provide the details of each of these steps below.

4.2.1 Ownership Weights

The current motion estimates, $\mathbf{a}_i$, for each layer can be used to compute a residual error, $r(\mathbf{x}; \mathbf{a}_i) = (\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t)$, using the optical flow constraint at each pixel. If the constraint belongs to layer i then the magnitude of the residual should be small. We compute a soft assignment of constraints to layers by defining a likelihood function, $l(r, \sigma)$, over the residual errors, r. $l(r, \sigma)$ represents the likelihood that pixel $\mathbf{x}$ at time t

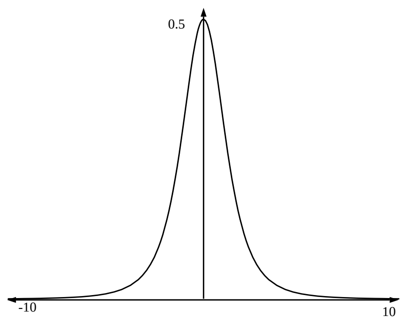


Figure 4.1: Weight function $l(r, \sigma)$.

is explained by the parameters $\mathbf{a}_i$ (that is, that $I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x}; \mathbf{a}_i), t-1)$), here it is defined to be

$$\frac{1}{K} l(r(\mathbf{x}; \mathbf{a}_i), \sigma) = p(I(\mathbf{x}, t) \mid \mathbf{a}_i), \tag{4.8}$$

where σ is a scale parameter, and K is a constant that is independent of σ and makes the left hand side integrate to one. Hence, K is defined to be

$$\int_{-\infty}^{\infty} l(r, \sigma)\ dr = K.$$

We could take $l(\cdot)$ to be a normal distribution as in ([55, 72]) or a robust distribution as in ([20, 58]). A robust $l(\cdot)$ falls off more rapidly than a normal distribution and provides a sharper assignment of pixels to layers. We use a robust $l(\cdot)$ and take

$$l(r(\mathbf{x}; \mathbf{a}_i), \sigma) = \frac{\sigma^3}{(\sigma^2 + r(\mathbf{x}; \mathbf{a}_i)^2)^2}$$

and $K = \pi/2$. The function is plotted in Figure 4.1.

For a given pixel, we consider the likelihood that the constraint at pixel $\mathbf{x}$ belongs to layer i to be

$$l_i(\mathbf{x}, \sigma) = l(\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t, \sigma) = \frac{\sigma^3}{(\sigma^2 + (\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t)^2)^2}. \tag{4.9}$$

Note that in Figure 4.1, constraints having small residuals are considered to have a higher likelihood of belonging to layer i, and this likelihood decays quickly to zero as the error increases.

We will also need the likelihood, say $l_{\mathcal{L}+1}(\mathbf{x}, \sigma)$, that the constraint at a given pixel arises from the outlier process. Following [43, 55] we take any constraint to be equally likely to be produced from this outlier process. We define an outlier to be a residual with magnitude greater than 2.5σ (cf. [86, 87]) and hence the likelihood of an outlier is

$$l_{\mathcal{L}+1}(\sigma) = \frac{\sigma^3}{(\sigma^2 + (2.5\sigma)^2)^2}. \tag{4.10}$$

Finally, let $M(\mathbf{x})$ be the sum of the likelihoods for each layer, including the outlier layer; that is, $M(\mathbf{x}) = \sum_{i=1}^{\mathcal{L}+1} l_i(\mathbf{x}, \sigma)$.

Given these likelihoods $l_i(\mathbf{x}, \sigma)$, $1 \leq i \leq (\mathcal{L} + 1)$, the ownership weights $m_i(\mathbf{x}, \sigma)$ are determined by rescaling the likelihoods so that the weights sum to one. That is,

$$m_i(\mathbf{x}, \sigma) = l_i(\mathbf{x}, \sigma)/M(\mathbf{x}), \tag{4.11}$$

for $1 \leq i \leq (\mathcal{L} + 1)$. This rescaling is particularly useful in situations where the layers are close enough so that a constraint has a significant likelihood of coming from two or more layers. In such a situation the reweighting can reduce or eliminate a bias towards the mean of nearby layers (see [72]). Note that the computation of ownership weights (Equation (4.11)) is independent of the uniform scaling constant K in Equation (4.8).

4.2.2 Layer Parameters

The ownership weights in Equation (4.11) provide a soft assignment of the data into the different layers. Given this assignment, we consider updating the layer parameters according to the reweighted least squares problem

$$E(\mathbf{a}) = \sum_{\mathbf{x} \in \mathcal{R}} \sum_{i=1}^{\mathcal{L}} m_i(\mathbf{x}, \sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}_i) + I_t)^2 \tag{4.12}$$

where we minimize E with respect to the affine parameters of every layer as described below. Note that the ownerships are used to weight the squared error from each layer. This formulation can be expected to be robust to outliers since the ownership for large residuals will be small.

This approach differs from the probabilistic mixture model approach presented in [55] in two ways. The first is that in place of the likelihood function in Equation (4.9) the previous mixture model approach used Gaussian component densities. We chose the current likelihood function for its robustness to outliers and computational convenience.

The second difference is that the mixture model approach attempts to estimate the mixture probability π_i (Equation (4.5)), averaged over the image region, that a constraint will belong to each of the layers. Here we simply take it to be equally likely to get a constraint from any of the layers.

4.2.3 Estimating and Annealing the Scale Parameter

The scale parameter, σ_i, controls the shape of the likelihood function and hence the assignment of constraints to layers. A small value of σ_i will force the weights towards zero or one (a hard assignment of constraints to layers) while a large value will cause constraints to be shared by multiple layers. Finding the global minimum of Equation (4.12) when σ_i is small is complicated by the existence of local minima. Jepson and Black [56] used an EM-algorithm coupled with deterministic annealing to estimate a layered representation of a gray-level image. Similarly, Ju *et al.* [60] used a simple gradient descent scheme with annealing to estimate motion. The idea is to start with a sufficiently large σ_i so that no data are treated as outliers and constraints are shared by layers. Then as σ_i decreases the influence of outliers is gradually reduced and constraints are assigned to layers.

We would like to estimate this parameter automatically from the data (cf. [87]). We use the similar annealing approach as described in Section 3.1.3. At each iteration we compute the current value of σ_i by taking into account an estimated scale parameter $\tilde{\sigma}_i$ and an annealing parameter $\hat{\sigma}_i$, such that

$$\sigma_i = \hat{\sigma}_i * \tilde{\sigma}_i,$$

where $\hat{\sigma}_i$ is set to a large value in the first iteration. It decreases by a fixed rate 0.95 till it reaches the unit value 1.0 in the last iteration. These parameters are the same as those that used in the single-layer "Skin and Bones" model. They are also fixed for all the experiments in the rest of this thesis.

We can estimate the scale parameter $\tilde{\sigma}_i$ independently for each layer using the estimated affine parameters, $\mathbf{a}_i$. Given a current value of σ_i for each layer i, if the residual errors were Gaussian then we could define $\tilde{\sigma}_i$ to be

$$\tilde{\sigma}_i^2 = \frac{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x};\mathbf{a}_i) + I_t)^2}{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)}. \tag{4.13}$$

In our case, however, we assume that the residuals, r come from the distribution

$$l(r,\sigma) = \frac{2\sigma^3}{\pi(\sigma^2 + r^2)^2} \tag{4.14}$$

and hence the estimate $\tilde{\sigma}_i$ using Equation (4.13) will be biased. To estimate the correct $\tilde{\sigma}_i$ for our distribution, we integrate an approximate version of Equation (4.13) over the range of inliers (values between $[-2.5\sigma, 2.5\sigma]$), which gives

$$\int_{-2.5\sigma}^{2.5\sigma} l(r,\sigma) r^2 \, dr \Big/ \int_{-2.5\sigma}^{2.5\sigma} l(r,\sigma)\, dr \approx 0.54\sigma^2. \tag{4.15}$$

Therefore, the estimated scale parameter $\tilde{\sigma}_i$ for layer i can be bias corrected by

$$\tilde{\sigma}_i = \sqrt{\frac{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x},\mathbf{a}_i) + I_t)^2}{0.54\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)}}. \tag{4.16}$$

4.2.4 Implementation

To estimate the affine transformation of each layer we implement the general motion estimation framework shown in Figure 3.2. In the coarse-to-fine strategy, if the patch size is not larger than 4×4 pixels, affine motion cannot reliably be estimated and a translational flow model is used instead.

Within each level of the pyramid, the affine transformation is estimated using a variant of the EM-algorithm, which is summarized in Figure 4.2. The process iterates between the E-step and M-step while the σ_i are estimated and annealed as described above. In the M-step the weights are computed as described above. Given these weights, the E-step updates the estimates of the affine transformations using an iterative gradient descent method. This gradient descent scheme in the E-step is a straightforward variation of that used in Section 3.1.4. An alternative to gradient descent would be to solve the Equation (4.7) using weighted least squares estimation.

4.2.5 Examples

Examples should help clarify how the layered method behaves. In this section, we apply the algorithm in the entire image region on synthetic and real image sequences.

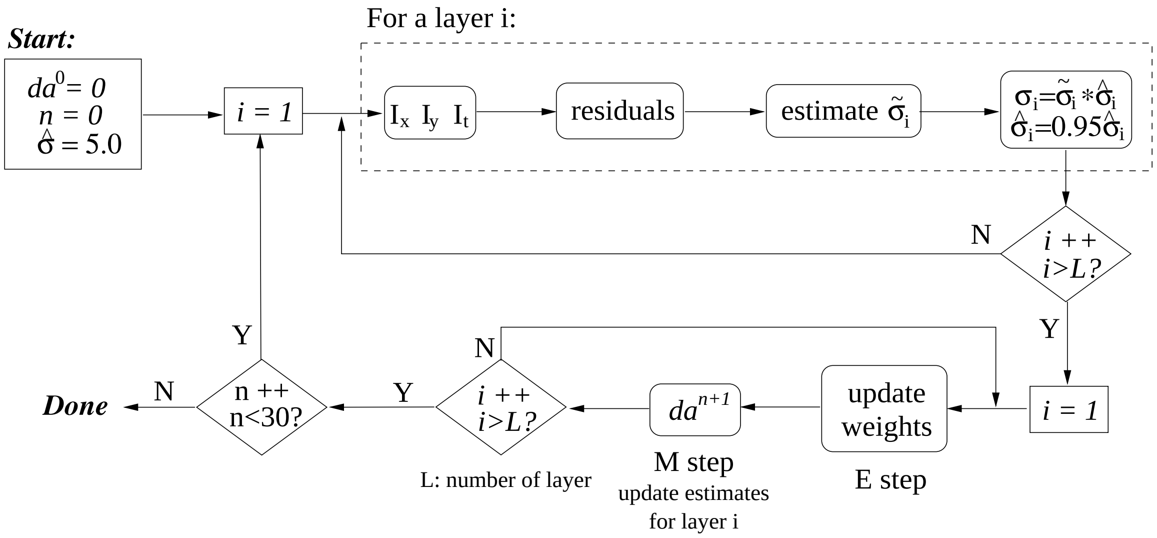
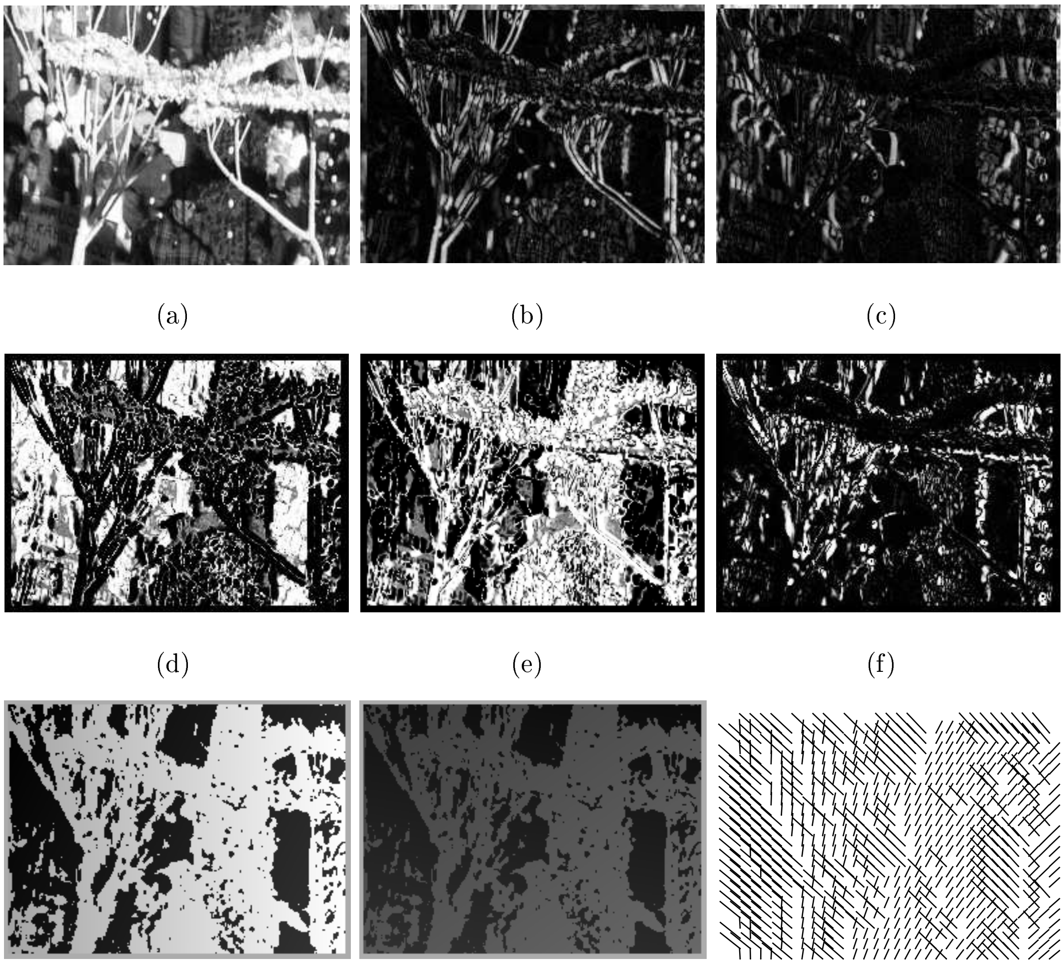


Figure 4.2: The iterative minimization process: multi-layer motion estimation

Skaters Sequence

Consider the synthetic "Skaters" image sequence[2] shown in Figure 4.3 (a) which consists of two motions corresponding to the trees in the foreground and the crowd of people in the background respectively. We took the region to be the entire image and assumed that the motion could be described by two layers plus an outlier layer. The estimated motion for each layer was used to warp the second image in the sequence backwards towards the first, in effect stabilizing the sequence with respect to one of the motions. The pixel-wise absolute intensity differences between these stabilized pairs are shown in Figure 4.3 (b) and (c). One recovered motion nulls the background while the other nulls the foreground. The weights, $m_i(\mathbf{x}, \sigma)$, for each layer are shown in Figure 4.3 (d-f). White indicates a weight near 1.0 while black indicates a weight near 0.0. The gray areas are weights near 0.5 and correspond to regions of uniform brightness where the optical flow constraints can be equally well assigned to any layer. The high weights in the outlier layer (see Figure 4.3(f)) predominantly correspond to areas near depth discontinuities or shadow boundaries. Figure 4.3 (g) and (h) show the horizontal and vertical motion at every pixel and Figure 4.3 (i) shows the flow as a needle diagram. The flow at each pixel is chosen from the layer most likely to account for that pixel's motion.

[2]We thank Jim Bergen for this sequence.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.3: **Skaters sequence: Mixtures of Robust Bones;** (a) First image in the sequence; (b) Absolute difference between the original image and the second image stabilized with respect to layer 1; (c) Absolute difference image with respect to layer2; (d) Weights for layer 1; (e) Weights for layer 2; (f) Weights for outlier layer; (g) Horizontal component of flow; (h) Vertical component of flow; (i) Vector field.

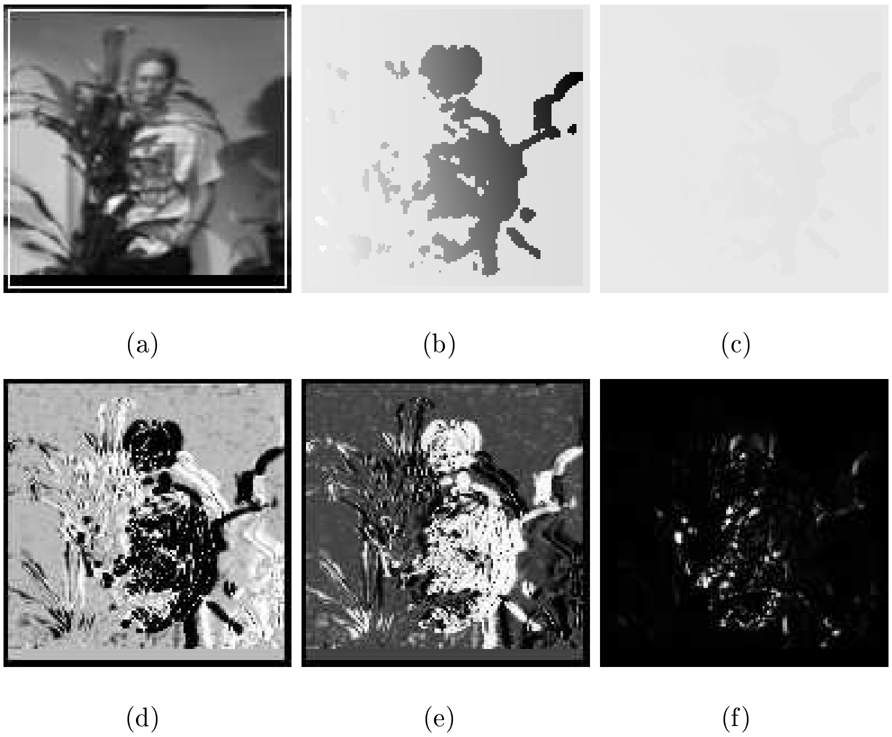


Figure 4.4: **Plant Sequence: Mixtures of Robust Bones;** (a) first image in the sequence; (b) horizontal component of flow; (c) vertical component of flow; (d) weights for layer one; (e) weights for layer two; (f) weights for outlier layer.

Plant Sequence

Consider the natural "plant" sequence[3] shown in Figure 4.4 (a) where a person moves to the right behind the two plants in the foreground. Again, we assume that the motion in the scene can be described by two layers plus an outlier layer. Figure 4.4 (b) and (c) show the horizontal and vertical motion at every pixel. The flow at each pixel is chosen from the layer which has larger ownership weight. The ownership weights for each layer are shown in Figure 4.4 (d-f). The high weights (white) in the outlier layer (Figure 4.4 (f)) predominantly correspond to areas near depth discontinuities or shadow boundaries. Most of the person has been correctly included in the second layer despite the occlusion caused by the plant's leaves. Note that there is a cast shadow moving in synchrony with the person in the scene and it thus appears in the second layer too.

The examples indicate that the layered method can provide robust estimates of the image motion in the presence of outliers and multiple motions.

[3]The sequence is available online: ftp://whitechapel.media.mit.edu/pub/images/plantseq.tar.Z.

4.2.6 Limitations

It is important to know that the method of robust mixture of affine motions is not a "dense" method. Therefore, the flow may not exhibit smoothness at the pixel level. The following examples will illustrate this property more clearly.

First, consider the variant of the Expectation-Maximization (EM) algorithm, which is used to estimate the mixture of locally affine motions. In the E steps, the ownership weights are estimated given the current motion models for the layers, with the assumption that the layer assignments at one location are independent of the layer assignments at all other locations. In other words, knowing the ownership of a particular location yields no information about the weights of all other locations in the image. In image formation, this is rarely the case. On the contrary, neighboring points with similar intensity or motion are likely to be from the same object [104].

To illustrate the problem that is caused by this assumption, consider the example shown in Figure 4.5[4]. This is a synthetic sequence where two textured circles move towards each other on top of a stationary textured background. We estimate three affine layers and an outlier layer over the entire image region. In the images of the weights (Figure 4.5 (e) to (h)), the gray areas correspond to regions of uniform brightness where the optical flow constraints can be equally well assigned to any layer. Figure 4.5(d) shows the ownership map, which is generated given the number of the layer that is most likely to account for that pixel's motion. Note that the homogeneous regions in the moving circles are assigned to the background layer. Due to the same problem, the estimated flow was noticed to be "speckled", rather than smooth, in regions that have little texture [58]. Recall that we assign a pixel to the layer which has the largest ownership weight. Given any motion model, the residual at the pixel that has little texture is always very small, say 0.0. Then, the ownership weight given by Equation 4.11 will be,

$$m_i(\mathbf{x}, \sigma_i) = \frac{l_i(\mathbf{x}, \sigma_i)}{M(\mathbf{x})} = \frac{1}{\sigma_i * M(\mathbf{x})}$$

where $M(\mathbf{x})$ is the sum of the likelihoods of each layer. Clearly, pixel $\mathbf{x}$ will be assigned to the layer that has the smallest σ. For the Textured Circle sequence, σ of layer one

[4]We thank H. Shum for this sequence.

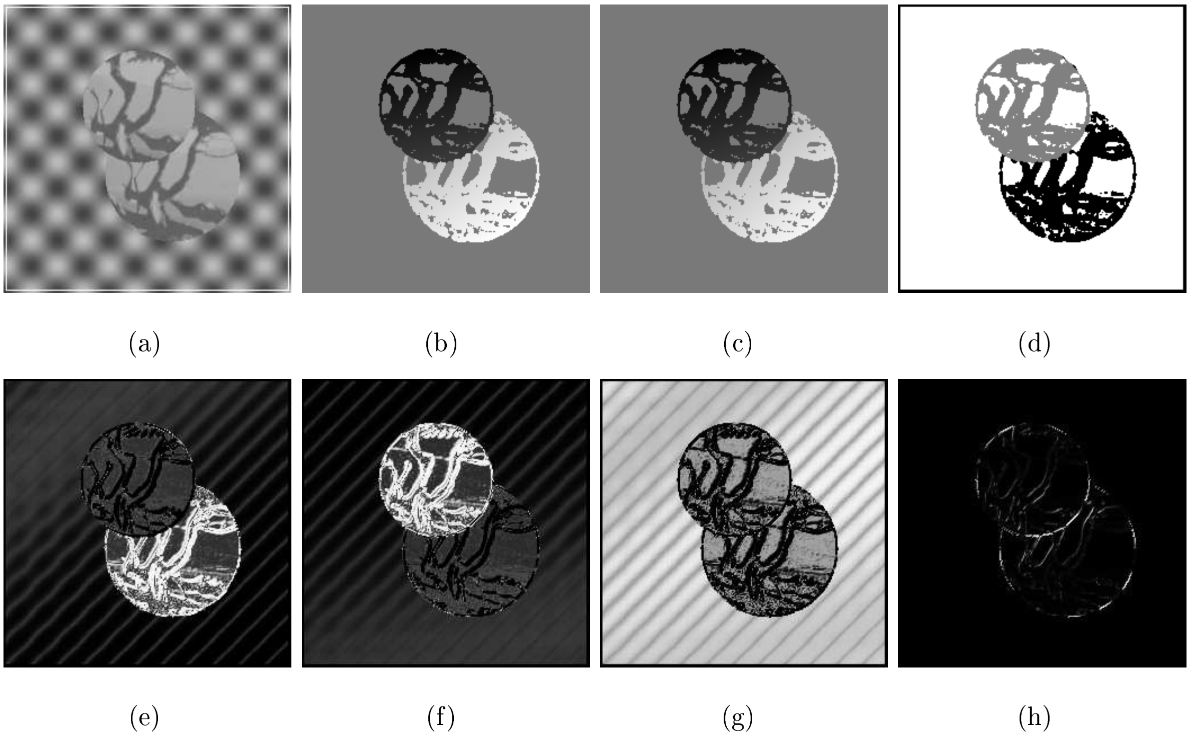


Figure 4.5: **Textured Circle Sequence: Mixtures of Robust Bones;** (a) first image in the sequence; (b) horizontal component of flow; (c) vertical component of flow; (d) ownership map; (e) weights for layer one; (f) weights for layer two; (g) weights for layer three; (h) weights for outlier layer.

is the smallest, hence pixels that have little texture are assigned to layer one (Figure 4.5(d)). Therefore, we need a spatial coherence constraint that favors the solution with coherent spatial labeling.

Second, consider an example of 1D noisy data shown in Figure 4.6. How would parametric models be fitted to these data? As pointed out by Weiss in [103], different outcomes will be obtained depending on the starting models. Two solutions are shown in Figure 4.6, where solution 1 contains three distinct models, and solution 2 contains only two. Which solution should be favored depends on the scene. There are analogous problems in motion estimation.

Consider the Synthetic Bars sequence[5] shown in Figure 4.7 (a), where the true velocity of the background is $(1.0, 0)$, the true velocity of the upper square is $(-1.0, 1.0)$, the lower square expands with respect to its center [6], and the true velocity of the long bar is

[5] The sequence is available online: ftp://whitechapel.media.mit.edu/pub/images/synthseq.tar.Z.
[6] no ground truth is available for this motion.

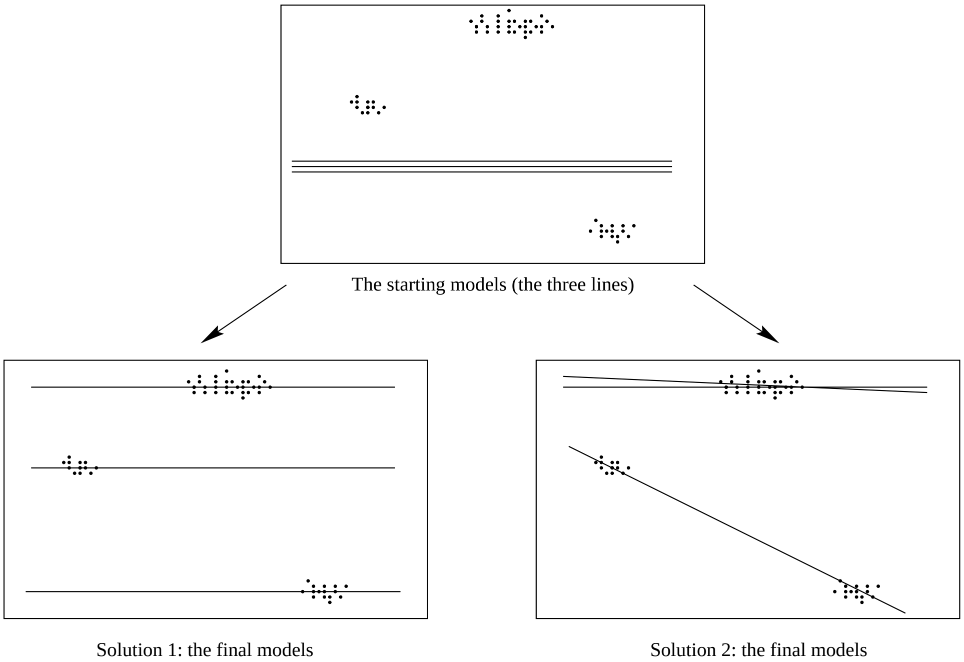


Figure 4.6: Outputs of a multiple model fitting algorithm: two possible solutions.

$(-1.0, -1.0)$. This sequence contains a significant amount of aliasing, particularly in the lower square. We simultaneously estimated four affine models, given zero as the initial value for the affine parameters. The coarse-to-fine strategy is not used, since the image velocities are not greater than one pixel. From the weight images shown in Figure 4.7 (e)-(i), we see that the recovered affine motions of layer one (Figure 4.7 (e)) and layer two (Figure 4.7 (f)) are similar and correspond to the background motion; the recovered affine motion of layer three (Figure 4.7 (g)) fits the mainly the right side of the lower square; the estimated layer three (Figure 4.7(h)) models the upper square, the long bar and part of lower square.

	a_0	a_1	a_2	a_3	a_4	a_5
Layer 1:	0.997	0.0	0.0	0.0	0.0	0.0
Layer 2:	0.970	0.0	−0.001	0.003	0.0	0.0
Layer 3:	0.712	0.0	−0.005	0.059	−0.002	0.0
Layer 4:	−0.832	−0.006	0.002	−0.388	−0.020	−0.004

Table 4.1: Recovered affine motion coefficients for the Synthetic Bars sequence.

Table 4.1 shows the recovered affine motion coefficients of each layer. We can see that the coefficients of layer one and two are about equivalent, while the coefficients of layer three reflect a translational motion (a_0) in the horizontal direction and a stretching

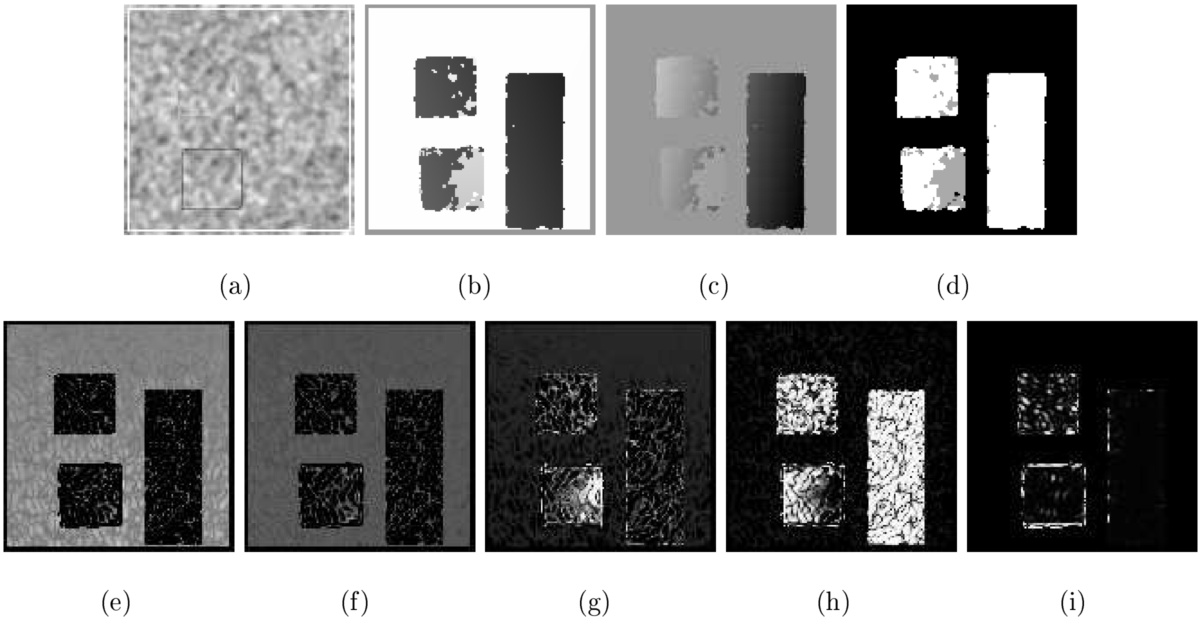


Figure 4.7: **Synthetic Bars Sequence: Mixtures of Robust Bones;** (a) first image in the sequence; (b) horizontal component of flow; (c) vertical component of flow; (d) ownership map; (e) weights for layer one; (f) weights for layer two; (g) weights for layer three; (h) weights for layer four; (i) weights for outlier layer.

motion (a_4) in the vertical direction that fits both the upper square and the long bar. The example illustrates the instability problem in motion estimation, especially when fitting large number of layers to data. Additional constraints may be needed to ensure the stability of the estimation. Since in general data is more likely to be grouped into spatially coherent segments, we can introduce an spatial coherence constraint to make the estimation problem more stable.

Third, consider the example shown in Figure 4.8 (a), in which three layers are estimated for the Flower Garden sequence. Since the motion of the flower bed is more complex than affine, both layer two (Figure 4.8 (e)) and three (Figure 4.8 (f)) fit portions of the ground, the houses, and the sky. The motion of the tree, however, is not recovered correctly. Layer one is affected by the leverage points from the lower part of the flower bed, and the recovered affine motion coefficient, a_0, shown in Table 4.2, is smaller than the true translation velocity of the tree (which is approximately 5.5 pixels). Table 4.2 also demon-states that a_0 of layer three is larger than a_0 of layer two. However, layer three mainly contains farther objects, such as the houses, therefore its translation

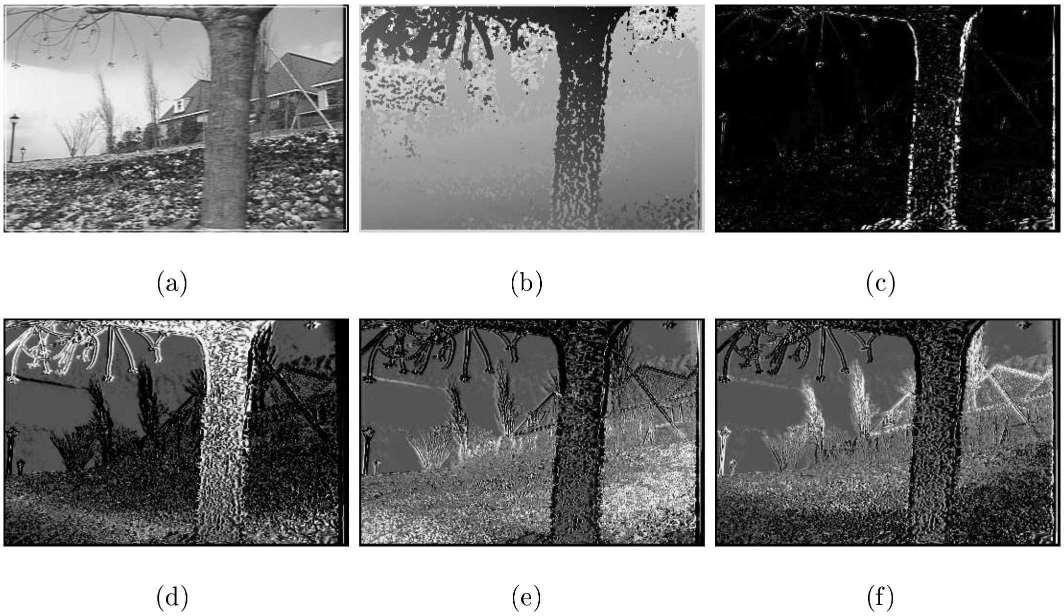


Figure 4.8: **Flower Garden Sequence: Mixtures of Robust Bones;** (a) first image in the sequence; (b) horizontal component of flow; (c) weights for outlier layer; (d) weights for layer one; (e) weights for layer two; (f) weights for layer three.

velocity should be smaller than those of the layer two, which contains closer surfaces, such as the flower bed.

The following section illustrates how a spatial coherence constraint on the ownership weights can improve the multi-layer motion estimation problem.

	a_0	a_1	a_2	a_3	a_4	a_5
Layer 1:	-3.852	-0.007	0.009	0.319	0.001	-0.004
Layer 2:	-1.138	-0.003	-0.016	-0.019	0.0	-0.001
Layer 3:	-1.425	-0.001	-0.006	-0.034	0.0	-0.001

Table 4.2: Recovered affine motion coefficients for the Flower Garden sequence.

4.3 A Spatial Constraint on Ownership Weights

In the formulation described in Section 4.2, pixels are assigned to models based on their residuals only. In effect, this assumes a type of independence in the ownership weights, However, this is rarely the case. In this section, we describe a modification of the EM algorithm which can take advantage of new information that neighboring pixels are likely to belong to the same layer.

The previous work by Black and Jepson [18], and by Etoh and Shirai [34] discussed the use of static intensity constraints for motion computations. In Black and Jepson's work, the image was first segmented into multiple fragments of similar intensity by an anisotropic diffusion algorithm. Affine flow was then estimated separately for each fragment. Likewise in Etoh and Shirai's work, the image was segmented into region fragments by a clustering procedure. Each fragment was associated with a spatial position, a 2D translation and an intensity.

Weiss and Adelson [105] also addressed the smoothness of layered estimates at the pixel level. The main difference between their approach and the previous two approaches is that Weiss and Adelson used intensity segmentation to constrain the possible motion models, thus static cues can be grouped together based on their consistency with a common global motion. To be specific, they developed an alternate E step which assumed spatial dependence of the ownership labels. Instead of computing the ownership weights at each image position independently, a Markov Random Field (MRF) distribution was used to constrain the ownership weights. That is, the estimated ownership weights are the extrema of an energy function of which one term is defined to be,

$$\sum_{\mathbf{x},\mathbf{y},i} w_{\mathbf{x},\mathbf{y}} m_i(\mathbf{x}, \sigma_i) m_i(\mathbf{y}, \sigma_i) \tag{4.17}$$

The term measured the joint likelihood of two locations $\mathbf{x}$ and $\mathbf{y}$, with an expected strength $w_{\mathbf{x},\mathbf{y}}$ which was determined by the distance between the two locations and the difference between the intensity values at these two locations. Their formulation has two problems. First, the minimization process with respect to the ownership weights is computationally intensive. Second, the solution depends on the definition of $w_{\mathbf{x},\mathbf{y}}$, which is difficult to estimate in a principled way.

Similarly, we can add a term to encourage spatial coherence of the ownership weights to the "Skin and Bones" model, under the assumption that nearby pixels are likely to belong to the same model.

4.3.1 A Posterior Probability Function of Ownership Weights

The extension involves the addition of a spatial prior on the ownership weights. More exactly, we need to define the conditional probability of assigning a pixel to a model given the observed motion constraint and the ownership weights of its neighbors. We develop the following formulation based on a posterior probability $p(H_i|D_i, C_i)$ that considers a spatial coherence constraint of the ownership weights. Bayes' theorem gives the rule for updating belief in a hypothesis H_i (i.e. the probability that pixel $\mathbf{x}$ belongs to layer i) given the data D_i (which includes the observed motion constraint at $\mathbf{x}$, the estimated motion model $\mathbf{a}_i$ of layer i, and the scale parameter σ_i), and background information (context) C_i (which contains the ownership weights $w_i(\mathbf{y}, \sigma_i)$, $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, where $\mathcal{N}(\mathbf{x})$ is the set of neighboring pixels of $\mathbf{x}$):

$$p(H_i|D_i, C_i) \propto p(H_i|C_i) \; * \; p(D_i|H_i, C_i) \tag{4.18}$$

The left-hand term, $p(H_i|D_i, C_i)$, is called the posterior probability, and it gives the probability of the hypothesis H_i after considering the effect of data D_i in context C_i. The $p(H_i|C_i)$ term is just the prior probability of H_i given C_i alone; that is, the belief in H_i before the data D_i is considered. The term $p(D_i|H_i, C_i)$ is called the likelihood, and it gives the probability of the data assuming the hypothesis H_i and background information C_i is true.

We still assume that the likelihood function is independent of the background information C_i. Specially, we use $l_i(\mathbf{x}, \sigma_i)$ defined in Equation (4.11) to be the likelihood $p(D_i|H_i, C_i)$. Given $w_i(\mathbf{y}, \sigma_i)$, $\mathbf{y} \in \mathcal{N}(\mathbf{x})$ alone, the optimal estimate of the probability that pixel $\mathbf{x}$ belongs to layer i is the mean of $w_i(\mathbf{y}, \sigma_i)$, which is represented by $\mu_i(\mathbf{y}, \sigma_i)$ and used as the prior probability, $p(H_i|C_i)$. The ownership weight $w_i(\mathbf{x}, \sigma_i)$ is determined by rescaling the posterior probabilities so that the weights sum to one. That is:

$$w_i(\mathbf{x}, \sigma_i) = \frac{\mu_i(\mathbf{y}, \sigma_i) * l_i(\mathbf{x}, \sigma_i)}{\mathcal{M}(\mathbf{x})}, \qquad \mathbf{y} \in \mathcal{N}(\mathbf{x}), \tag{4.19}$$

$$\mathcal{M}(\mathbf{x}) = \left[\sum_{i=1}^{\mathcal{L}} \mu_i(\mathbf{y}, \sigma_i) * l_i(\mathbf{x}, \sigma_i)\right] + l_{\mathcal{L}+1}(\sigma). \tag{4.20}$$

Notice that the spatial prior is only applied to the motion layers, not to the outlier layer, since outliers are often not coherent in space. We use an alternative E step (Figure 4.9),

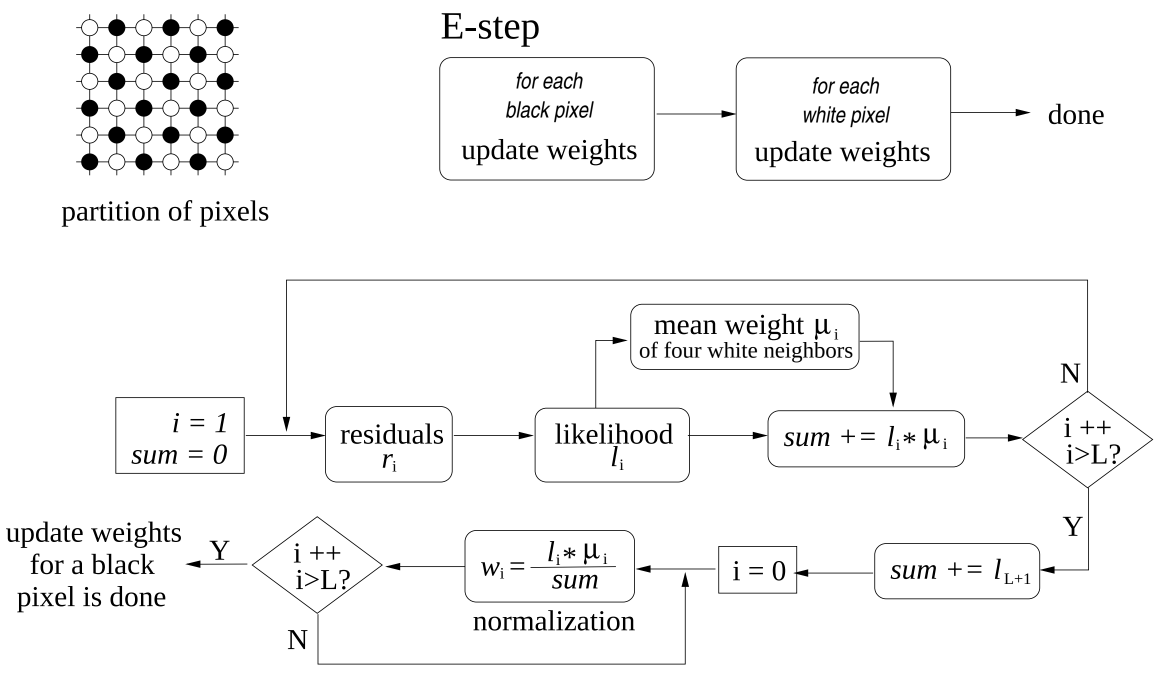


Figure 4.9: The E-step: multi-layer motion estimation

where the prior probabilities $\mu_i(\mathbf{y}, \sigma_i)$ are obtained by applying a single cycle of Besag's ICM procedure [13]: each pixel is visited in a raster scan order and, given the weights of its neighborhood, the mean value is computed.

There is a resemblance between the spatial prior probability used in our formulation and the mixture prior probability π_i used in [55]. Recall that without the prior, a motion constraint is simply taken to be equally likely to be from any of the layers. The prior probabilities used here and in [55] are based on the observations at other locations. In our formulation, only the local observations are considered, while in Jepson and Black's formulation, observations from the entire region are used to estimate the mixture prior probability.

The ownership weight with the spatial prior defined in Equation (4.19) provides a soft assignment of the data into different layers. Given this assignment, layer parameters can be updated by minimizing

$$E(\mathbf{a}) = \sum_{\mathbf{x} \in \mathcal{R}} \sum_{i=1}^{\mathcal{L}} w_i(\mathbf{x}, \sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}_i) + I_t)^2 \tag{4.21}$$

The M step is unchanged, as are all the parameters that are used in estimating the layered affine motions. Our formulation has two advantages over Weiss and Adelson's

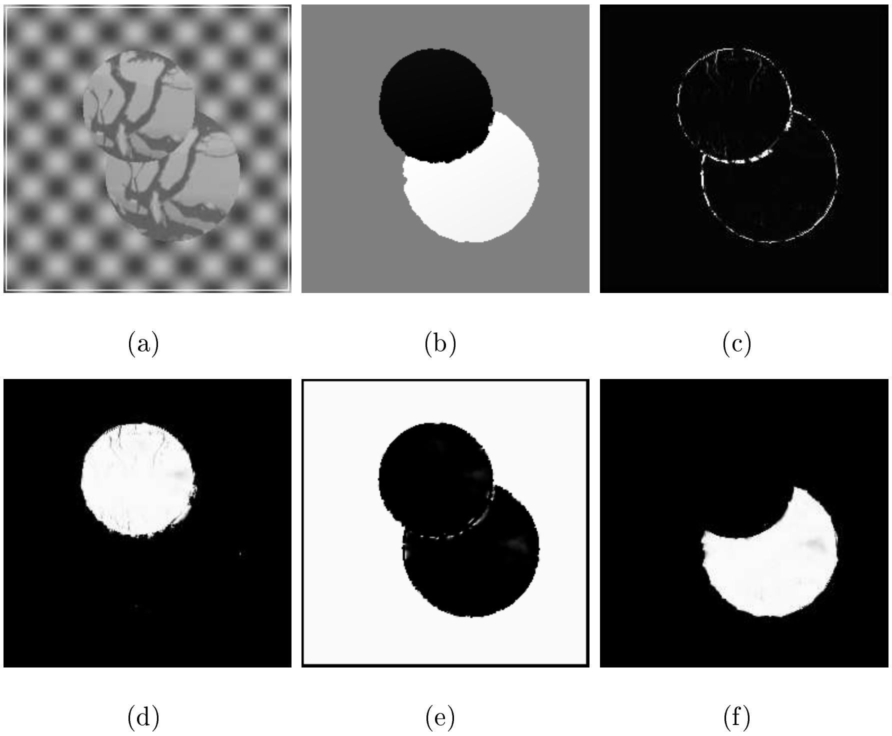


Figure 4.10: **Textured Circle Sequence: (with spatial prior)** (a) first image in the sequence; (b) horizontal component of flow; (c) weights for outlier layer; (d) weights for layer one; (e) weights for layer two; (f) weights for layer three.

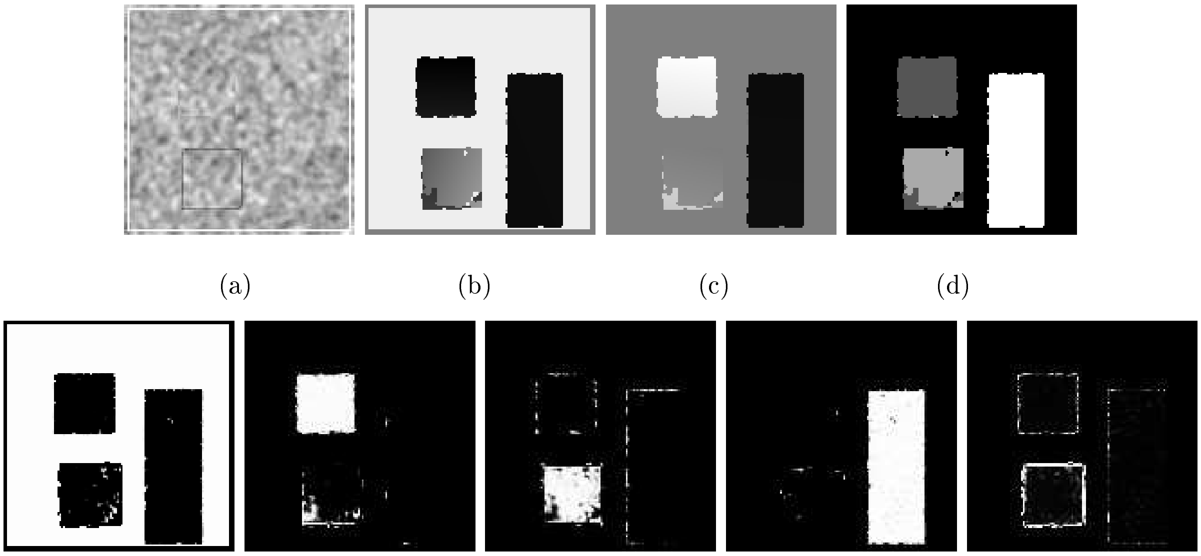
work [105]. First, it involves only a marginal increase of computation. The formulation fits naturally into the EM framework. Second, it focuses on the motion information. The spatial prior depends on the static intensity information indirectly through the likelihood function l_i. Hence, we avoid using any ad hoc function.

4.3.2 Examples

To illustrate the behavior of the addition of the spatial prior, we revisit the three examples shown in Section 4.2.6.

Figure 4.10 shows the example of Textured Circles sequence. Three coherent regions are recovered each corresponding to the background or one of the circles.

Figure 4.11 shows the example of the Synthetic Bars sequence. Four estimated layers are illustrated in the weight images, and Table 4.3 shows the estimated affine motion coefficients for each layer. The motion of the background (layer one) is recovered correctly. The recovered affine motions of the upper square (layer two) and the long bar (layer four)



(a) (b) (c) (d)

(e) (f) (g) (h) (i)

Figure 4.11: **Synthetic Bars Sequence: (with spatial prior)** (a) first image in the sequence; (b) horizontal component of flow; (c) vertical component of flow; (d) ownership map; (e) weights for layer one; (f) weights for layer two; (g) weights for layer three; (h) weights for layer four; (i) weights for outlier layer.

are close to the ground truth (layer two: $a_0 = -1.0, a_3 = 1.0$; layer four: $a_0 = -1.0, a_3 = -1.0$). Due to the aliasing at the corner of lower square, the corner is assigned to the wrong layer.

	a_0	a_1	a_2	a_3	a_4	a_5
Layer 1:	1.0	0.0	0.0	0.0	0.0	0.0
Layer 2:	−0.935	0.0	0.007	0.965	0.001	−0.006
Layer 3:	0.004	0.013	0.005	−0.081	−0.001	0.004
Layer 4:	−1.055	0.0	0.0	−1.050	0.0	0.0

Table 4.3: Recovered affine motion coefficients for the Synthetic Bars sequence: with the spatial prior

Figure 4.12 shows the estimated three-layer representation of the Flower Garden sequence. The ground plane is grouped to be one layer, and the sky area coherently defines another layer. Table 4.3 shows the estimated affine motions for each layer. Compare with Table 4.2, the first layer corresponding to the motion of the tree is recovered correctly. The estimated affine motions of the other two layers are improved as well. Layer two, which corresponds to the motion of closer flower bed and houses, is recovered with the faster horizontal translation velocity. Layer three corresponds to the motion of far away objects and sky, and its a_0 (absolute value) is the smallest among all the three layers.

	a_0	a_1	a_2	a_3	a_4	a_5
Layer 1:	-5.334	-0.012	-0.004	0.350	0.003	-0.005
Layer 2:	-1.230	-0.003	-0.014	-0.020	0.0	-0.001
Layer 3:	-0.883	0.005	0.0	-0.135	0.0	-0.001

Table 4.4: Recovered affine motion coefficients for the Flower Garden sequence: with the spatial prior

Clearly, the motion estimation algorithm with the spatial smoothness constraint on the ownership weights is more stable. The success of the method is because of the spatially coherent estimates of the layer support maps. We reduce the effect of leverage points that are far away from the coherent segments of a layer. Note that the gray areas in weight images have been eliminated. The ownership weights provide the "soft" assignments in the beginning of the EM framework. These "soft" assignments turn to "hard" assignments upon the convergence of the iterative procedure.

For the experiments shown in the rest part of this thesis, we minimize the energy function using Equation (4.21) as the data term. That is, the spatial smoothness prior on the ownership is applied by default. Only for the Yosemite sequence, we also show experimental results that does not apply the spatial prior, which will be pointed out in the text that explains the experiments.

4.4 How Many Layers?

We use the finite mixture models in this chapter to estimate a layered representation of the scene. The number of layers should be known in advance. The problem arises with the question of how many layers there are. In this section, we illustrate the performance of the algorithm given different numbers of layers. In Chapter 6, we will address the problem of how to choose the appropriate number of layers that are necessary to represent the motion in the scene.

Figure 4.14 demonstrates the example of the Flower Garden sequence given two to six layers. The first row shows the horizontal component of flow. The following rows show the weights of each layer. For example, the image in the N^{th} row and M^{th} column shows the weights for layer $N-1$ when M layers are estimated. In the two-layer case,

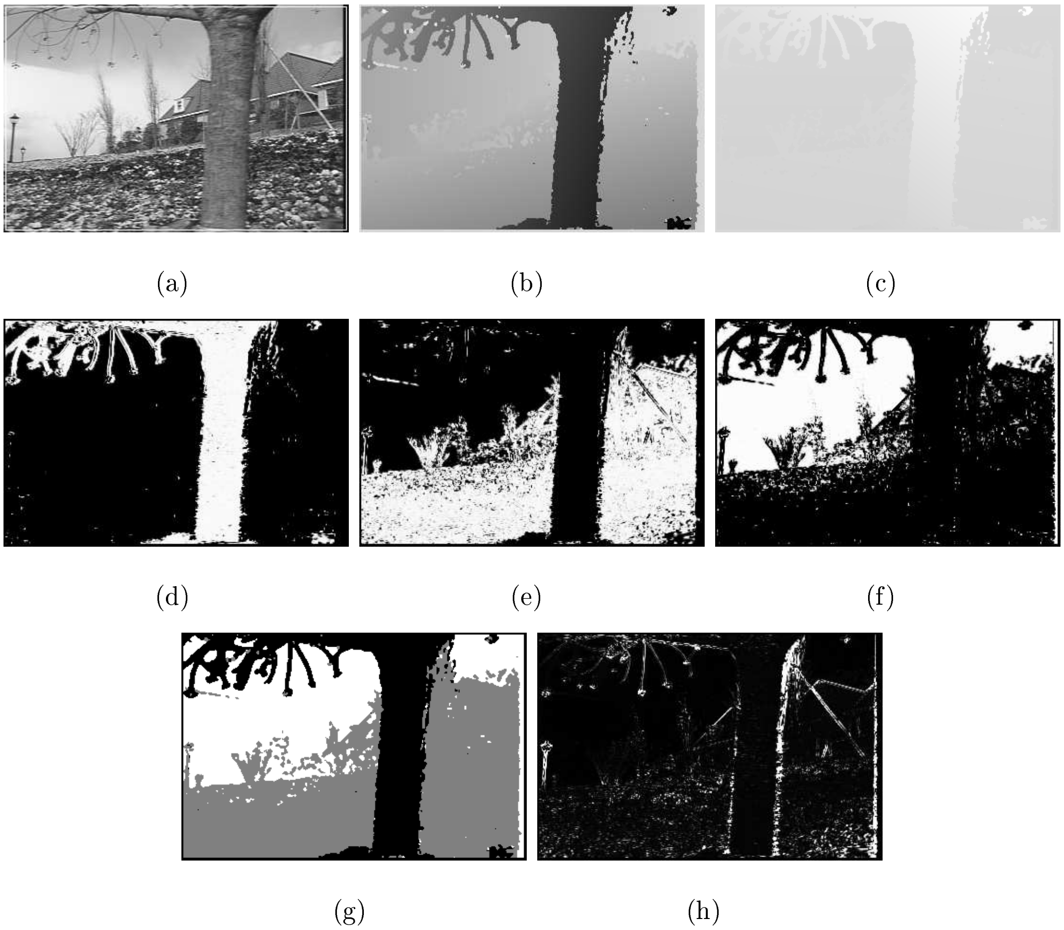
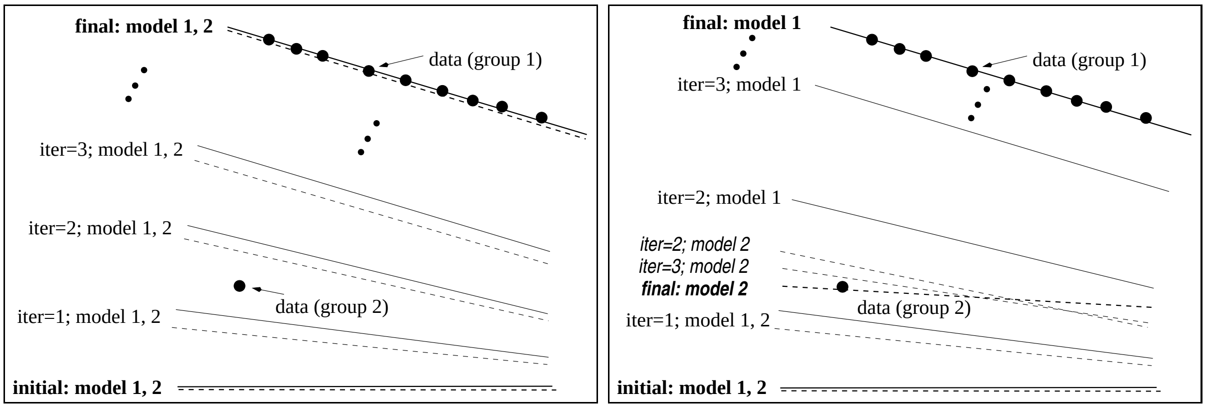


Figure 4.12: **Flower Garden Sequence: (with spatial prior)** (a) first image in the sequence; (b) horizontal component of flow; (c) vertical component of flow; (d) weights for layer one; (e) weights for layer two; (f) weights for layer three; (g) ownership map; (h) weights for outlier layer.

the foreground tree layer and the background flower garden layer are recovered. In the three-layer case, the far away sky areas are separated from the closer areas of flower bed and houses. In the four-layer case, we get a new layer of branches, however, it is also grouped with part of the flower bed. In the five-layer case, a new layer of distant objects (trees, poles) is recovered. In the six-layer case, the new layer includes the front lump in the right corner.

Note that our formulation is different from those that use a Markov Random Field (MRF) prior for motion segmentation [7, 97, 105]. Ayer [7], and Vasconcelos and Lippman [97] computed the number of motion models and their associated mixture parameters first, then the image was segmented into regions of homogeneous motion by maximum a posterior probability (MAP) estimation, where a MRF prior is assumed for the underlying motion regions. Weiss and Adelson incorporated the MRF prior into the EM framework, therefore motion segmentation and estimation are accomplished simultaneously.

The spatial prior used in our formulation is not like the standard MRF-based segmentation method. Although highly "speckled" segmentations are deemed unlikely, the segmentation may contain a coherent chunk of the flower bed moving with the branches (see Figure 4.14, the four-layer case). Therefore, our method allows spatially disjoint regions to move together, and generally these regions will be locally smooth. However, it is possible that the segmentation is fragmented (see Figure 4.14, the five-layer case). In addition, unlike the previous methods based on mixture models [55, 58], our formulation with the spatial smoothness prior on the ownership weights is unlikely to contain multiple layers that converge to a single motion. Since, if a pixel and its neighbors belong to the same layer i, the ownership weights w_i of these pixels tend to reinforce with each other, and converge to 1.0 much more quickly with the spatial smoothness prior than without the prior. To understand this process, consider the example shown in Figure 4.13, where 9 observations out of 10 come from the first group. We use this simplified example to illustrate cases where essentially one motion is present in a patch. Two models (solid line represents the first model and dashed line denotes the second model) will be fitted to data, and each starts with the same initial setting. Without the spatial prior (Figure 4.13(a)), data is assumed to come from each group equally for all iterations. Both mod-



(a) (b)

Figure 4.13: Converging in the optimization process: (a) without the spatial smoothness prior on the ownership weights; (b) with the spatial prior.

els converge to group one. With the spatial prior (Figure 4.13(b)), model one is pulled toward group one much more quickly. However, model two no longer moves to group one constantly, since the influence of data in group one decreases rapidly. Eventually, group one data have little influence on model two, which causes model two to converge to the group that may have little support.

To precisely compare the performance of the layered estimates, we apply the method to the Yosemite sequence when 2, 4, and 6 layers are estimated globally. Table 4.5 shows the error statistics, where the data of the one-layer case is from Section 3.1. There is a significant improvement with respect to the estimates of two layers and of four layers. However, errors of the four-layer model and the six-layer model are comparable. We can get the same conclusion by visual inspection of the estimated flows shown in Figure 4.15. The Yosemite sequence has a much more complex layer structure than Flower Garden sequence. There are local structures which are hard to recover precisely by globally defined layers. Moreover, approaches based on mixture models can cope with a small number of motions within a region. Estimating a large number of layers often does not make notable progress, yet it is computationally expensive. Therefore, the multi-layer robust motion estimation method still needs to be applied locally to smaller image regions.

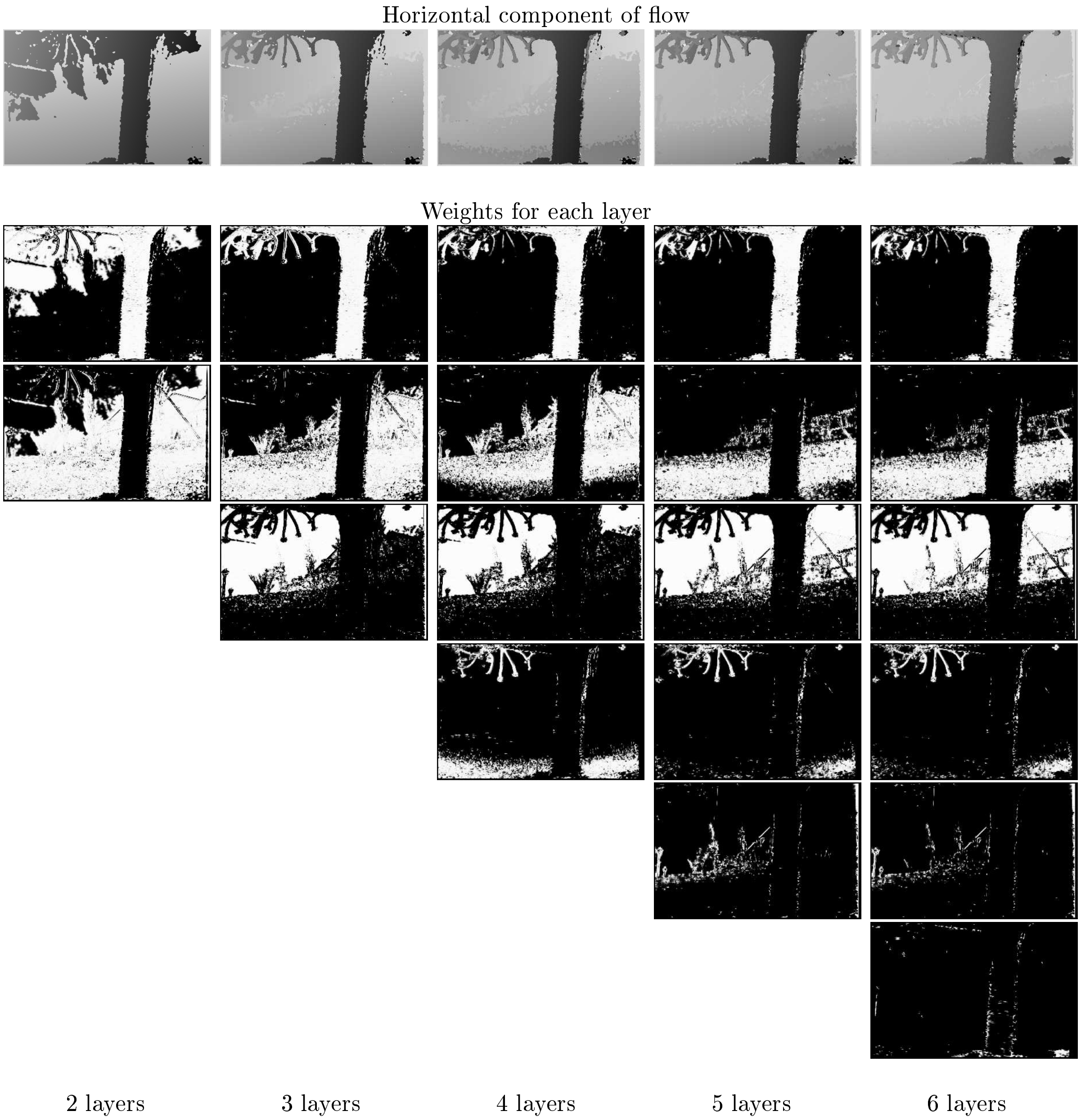


Figure 4.14: **How many layers: Flower Garden Sequence.**

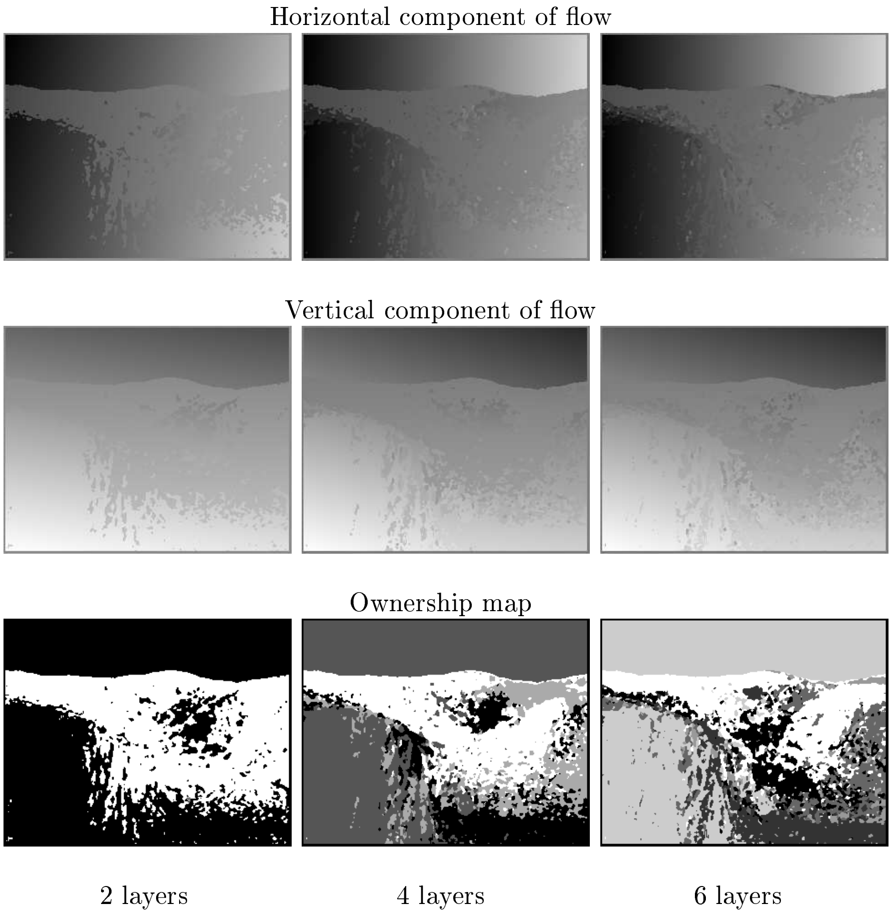


Figure 4.15: **How many layers: yosemite sequence.**

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
1 layer:	0.565	8.65°	3.55°	0.4%	1.7%	4.4%	13.0%	73.2%
2 layers:	0.338	6.46°	4.24°	1.9%	9.0%	19.6%	45.8%	81.2%
4 layers:	0.204	4.43°	3.96°	10.1%	31.6%	46.2%	68.6%	91.2%
6 layers:	0.200	4.45°	4.30°	9.0%	35.2%	49.1%	68.1%	91.6%

Table 4.5: Error results for the Yosemite sequence: layered affine motions.

4.5 Tiling the Image

As with the "Skin and Bones" method proposed in Chapter 3, we apply the multi-layer mixture model method described in Section 4.2 and 4.3 in small patches over the image; for example, we choose non-overlapping 32×32 pixel patches and estimate two layers in each patch for the experiments in this section.

For the first experiment in this section, we apply the same tiling of the image to Yosemite sequence as before, the first image of which is shown in Figure 4.16(a) with the grid of patches superimposed. However, we do not apply the spatial coherence prior on ownership weights for this experiment, but use the basic method described in Section 4.2 to recover two affine motions *independently* in each patch. The experiment is to illustrate the accuracy of a locally multi-layer mixture model method.

Technique	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Single-layer model	0.153	2.94°	2.58°	15.8%	44.7%	65.2%	85.8%	97.6%
Two-layer model	0.129	2.57°	2.51°	22.6%	51.7%	71.8%	89.5%	98.4%

Table 4.6: Error results for Yosemite sequence: multi-layer mixture model method.

In order to display the results, at each pixel we show the motion for the layer which has the maximum ownership. The horizontal and vertical components of this estimated flow are shown in Figure 4.16 (b) and (c). There are notable block structures in the flow fields[7]. Figure 4.16 (d) and (e) shows the weights for the two motion layers. Gray areas correspond to a weight of 0.5 and these regions indicate places there only one motion is present (for example, in the valley floor). Figure 4.16 (f) shows the points that were not account for by either layer and were treated as outliers. The result of

[7]Note that we have not added a regularization term ("skin") in the mixture model method. In the following chapter, we will discuss this term and demonstrate how it can improve the motion estimation.

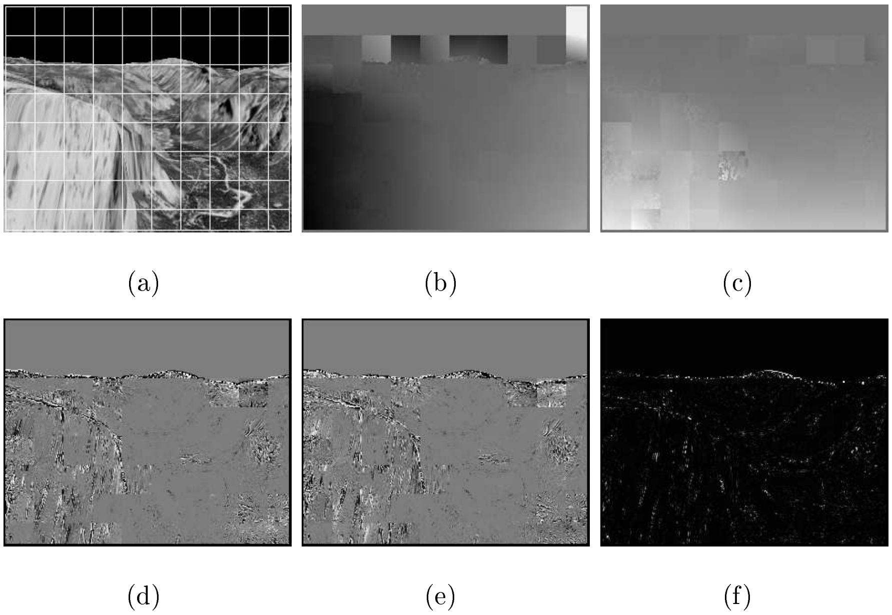


Figure 4.16: **Yosemite Sequence:** multi-layer bones with non-overlapping patches; (a) one image with the segmented image shown; (b) horizontal component of flow; (c) vertical component of flow; (d) weights for layer one; (e) weights for layer two; (f) weights for outlier layer.

the two-layer mixture model method is compared with the previous result of the single-layer model (without regularization term) in Table 4.6. The mean error is improved by approximately 14%, and the standard deviation is more or less unchanged.

4.5.1 Problems caused by tiling the image

Consider the experimental results depicted Figure 4.17, in which a textured circle in the center of the image translates to the left. In this example, we were "unlucky" with the placement of the regular grid of patches. In cases where a patch receives little support from one of the layers, two layers may converge to a single motion. This is most likely to occur when the small region contains very little texture. Estimated affine motions (Figure 4.17 (b) and (c)) are also affected by leverage points in the regions that span a motion boundary. With the spatial smoothness prior on the ownership weights, the estimated motions in some regions that contain motion boundaries are more stable. However, the boundary is still slightly jagged.

A layer with very little support may not be recovered, and the problem often happens

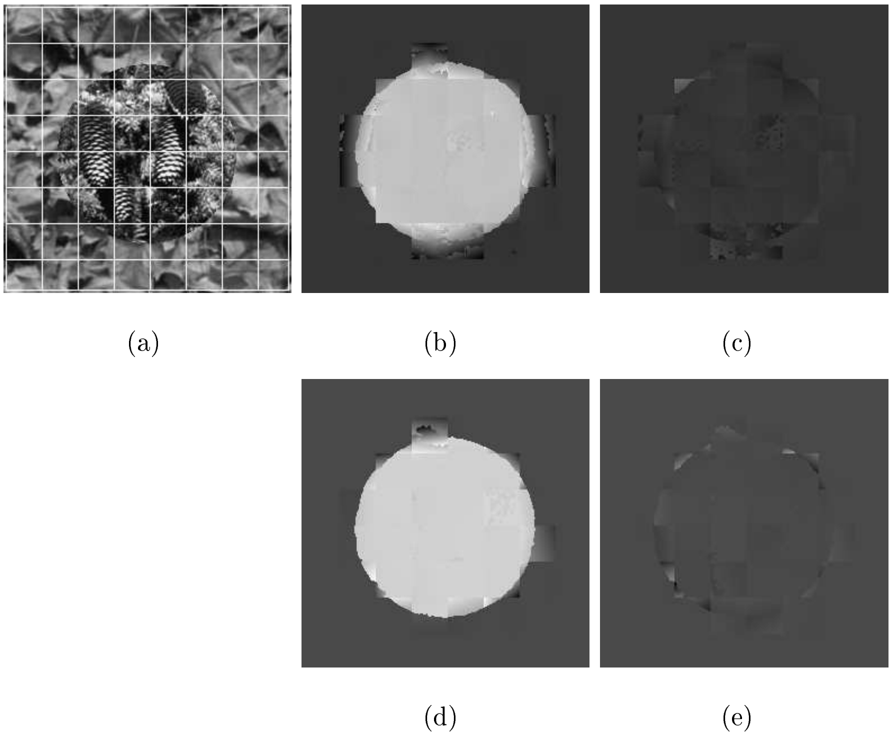


Figure 4.17: **Circle sequence:** multi-layer bones with non-overlapping patches; (a) one image with the segmented image shown; (b) horizontal component of flow (without the spatial prior); (c) vertical component of flow (without the spatial prior); (d) horizontal component of flow (with the spatial prior); (e) vertical component of flow (with the spatial prior).

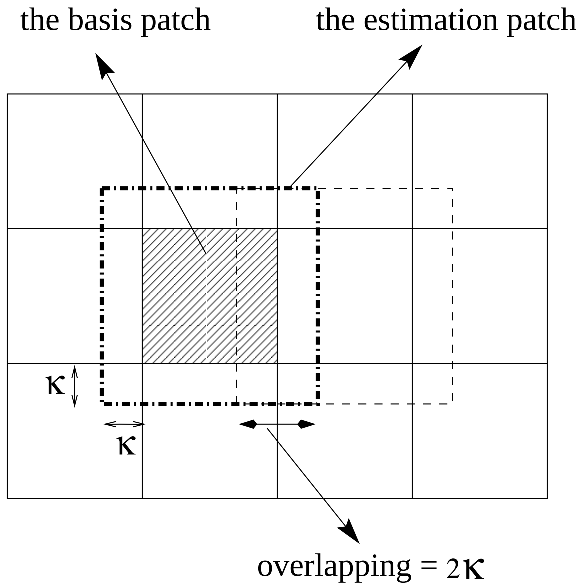


Figure 4.18: Tiling the image with overlapped patches.

when one of layers only occupies a small boundary portion of the region. Tiling the image with overlapped patches is a simple extension that might reduce the problem. Since if a layer is coherent and has supports close to the center part of the region, the layer will generally occupy a portion of the region that is large enough to estimate image motion correctly.

4.5.2 Tiling the image with overlapped patches

The image is still divided into equally sized 32×32 basic patches. However, we estimate layered affine motions in the corresponding estimation patches, which are dilated $\mathcal{K}$ pixels on each side of the basic patches (see Figure 4.18). Therefore, the size of overlapping between two neighboring patches at each side is $2 * \mathcal{K}$ pixels. Note that overlapping does not affect the optimization. The layered affine motions are still estimated independently as before in each estimation patch, whose size is now $(32+\mathcal{K}) \times (32+\mathcal{K})$. Bab-Hadiashar and Suter [8] also used overlapped patches to compute the motion estimates with a patch centered upon each pixel. They used an extreme case with $(1+\mathcal{K}) \times (1+\mathcal{K})$ patches. Obviously, the algorithm is more expensive with such a tiling scheme.

To display the results, at each pixel in the basic patches[8] we show the motion for the

[8]Basic patches are non-overlapping, and are defined in the same way as in Chapter 3.4.

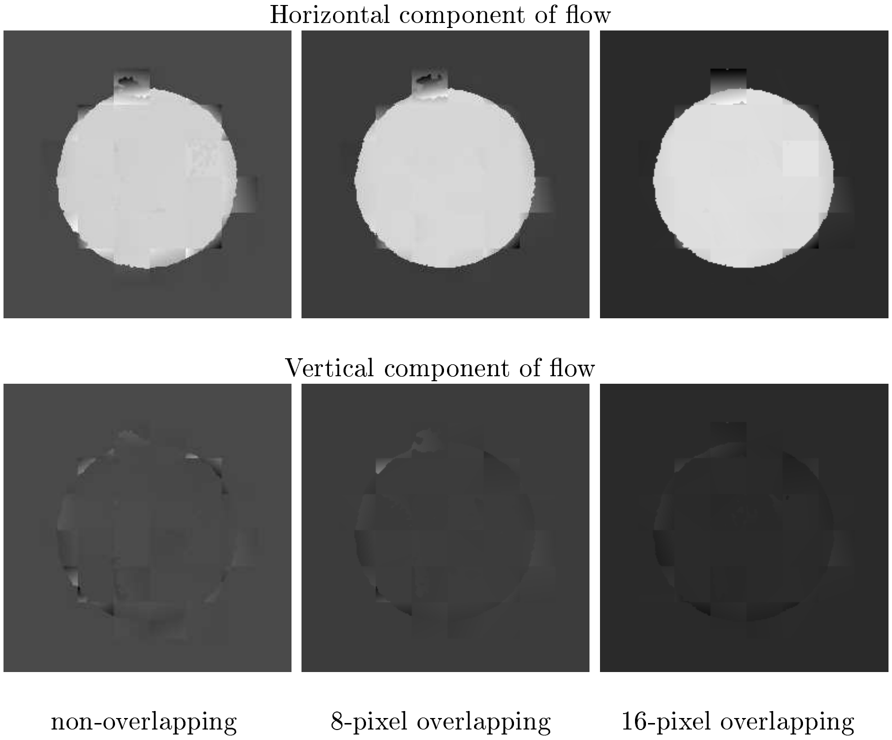


Figure 4.19: **Circle Sequence:** multi-layer bones with overlapped patches.

layer which has the maximum ownership. Figure 4.19 shows the estimated flows for the circle sequence in the cases of non-overlapping, 8-pixel overlapping, and 16-pixel overlapping respectively. The spatial smoothness prior on the ownership weights is used for this experiment. By visual inspection, almost all patches boarding the motion boundary have two clearly distinct motions when the patches are overlapped. Obviously, using overlapped patches increases the computational cost of the method. In general, we would like to keep the size of overlapping small. In the experiment, the improvement from 8-pixel overlapping to 16-pixel overlapping is marginal, thus 8-pixel overlapping is preferred.

To illustrate the improvement of accuracy, we applied the multi-layer bones method with overlapped patches to Yosemite sequence. Table 4.7 (row one to row three) shows the error statistics for non-overlapping, 8-pixel overlapping, and 16-overlapping patches respectively. Similarly, with 8-pixel overlapping patches the improvement is significant. While the performance of 16-pixel overlapping patches is comparable with that of 8-pixel ones.

If we apply the spatial smoothness prior on the ownership weights to Yosemite se-

Overlapping (pixels)	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Two-layer bones without the spatial prior:								
0:	0.129	2.57°	2.51°	22.6%	51.7%	71.8%	89.5%	98.4%
8:	0.114	2.34°	2.04°	25.3%	57.0%	75.2%	91.1%	98.8%
16:	0.114	2.36°	1.95°	26.2%	57.3%	72.0%	90.0%	99.5%
Two-layer bones with the spatial prior:								
0:	0.150	3.22°	3.97°	21.1%	46.3%	63.6%	82.8%	96.2%
8:	0.132	2.78°	2.77°	22.0%	51.5%	70.0%	86.5%	97.0%
16:	0.125	2.53°	2.56°	26.3%	56.0%	74.5%	88.6%	97.6%

Table 4.7: Error results for Yosemite sequence: tiling the image differently.

quence, the accuracy of the method will decline slightly. From Table 4.7 (row four to row six), we see that the percent of flow vectors with error larger than 10 degrees are about 2 percent less than the corresponding cases without the spatial smoothness prior. The best performance is obtained with 16-pixel overlapping patches.

Figure 4.20 shows the estimated flows with the spatial smoothness prior in patches of non-overlapping, 8-pixel overlapping, and 16-pixel overlapping respectively. In some patches of the front mountain, motion constraints are mainly from one orientation, thus the motion estimation problem is under-constrained. Using overlapped patches may bring in constraints from other orientations, but it will not solve the problem completely. We will show how "skin" can improve the estimation in the under-constrained patches in the next chapter. In addition, as we pointed out in Section 4.4, motions of multiple layers are not likely to converge to a single affine motion when the spatial smoothness prior is used. Since only one motion is present in most patches of Yosemite sequence, the errors in the estimated flow are larger when the spatial prior is used. Therefore, we need to select the proper number of layers in a patch that interprets the motion best. In Chapter 6, we will present a framework to choose the appropriate number of layers.

4.6 Examples: Multi-layer Bones

In this section, we demonstrate the behavior of the multi-layer bones method for some real image sequences. For all the experiments hereafter, we apply the spatial smoothness prior on the ownership weights to estimate layered affine motions. The patches are overlapped with their neighbors by 8 pixels on each side. By default, two affine layers

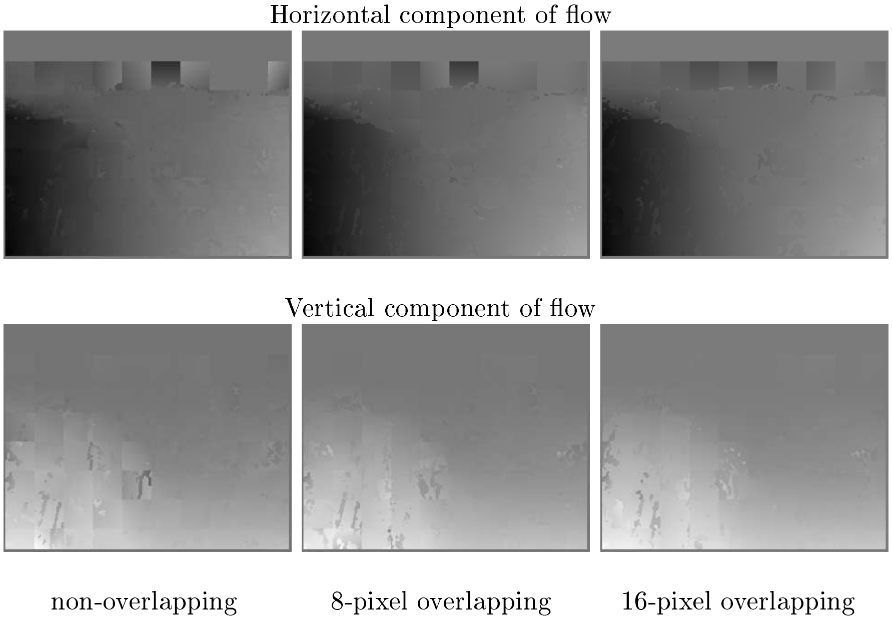


Figure 4.20: **Yosemite Sequence:** multi-layer bones with overlapped patches.

are assumed within each patch.

4.6.1 Flower Garden Sequence

The example of the Flower Garden sequence is given in Figure 4.21. As shown in the figure, the image is segmented into patches which may span surfaces at a number of depths. The affine motions of each patch are estimated independently. Compared to the results of single-layer "Skin and Bones" method (see Figure 3.25), the motion at the boundaries of the tree is estimated more accurately (see Figure 4.21 (b)). Figure 4.21 (c) shows the weights for the outlier layers. Note that outliers occur predominantly around the occlusion/disocclusion boundaries of the tree. Also in some regions which span tree branches and the sky, the multi-layer estimation process is under-constrained, and the affine motion of one layer is corrupted.

4.6.2 SRI Tree Sequence

The SRI Tree sequence is a more complex example with many discontinuities. The first image with segmented regions is shown in Figure 4.22 (a). Figure 4.22 (b) shows the horizontal component of flow, and Figure 4.22 (c) shows the weights of the outlier

(a) (b) (c)

Figure 4.21: **Flower sequence: multi-layer bones;** (a) one image with segmented patches shown; (b) horizontal component of flow; (c) weights for outlier layer.

(a) (b) (c)

Figure 4.22: **SRI Tree sequence:** multi-layer bones; (a) image with segmented regions shown; (b) horizontal component of flow; (c) weights for outlier layer.

layer. The estimated motion field exhibits sharp motion boundaries, yet still recovers the generally smoothed flow of the ground plane. We notice that the "speckling" still happens. However, speckling appears in the (small) region level, not in the pixel level as those in [58].

By visual inspection, it is clear that the motion fields shown in the examples of this and previous sections are not as smooth as the true flow and sometimes shows a block structure. In some regions, most notably at the regions that contain single oriented motion constraints or little texture, the estimated motion is incorrect. In the following chapter, we illustrate how a regularization term (skin) improves these locally affine estimates (bones).

Chapter 5

Regularization with Transparency

The need to regularize noisy data arises in many computer vision and image processing problems. Here we will consider what happens when there are multiple measurements at a given point. To illustrate what this means we will consider a 1D example of the standard regularization in Section 5.1, and extend it to the transparent case in Section 5.2. Section 5.3 applies the *regularization with transparency* to the optical flow estimation problem. We formulate the "Skin and Bones" model which allows multiple motion layers within image patches. Section 5.4 demonstrates the experimental results.

5.1 Standard Regularization

We first consider the standard regularization problem without transparency. Figure 5.1a shows an example of noisy and discontinuous data (cf. Blake and Zisserman [23]). Given noisy data measurements, d_k, $1 \leq k \leq K$, our goal is to estimate a piecewise-smooth approximation, u_k, of the true function. The standard regularization problem can be formalized as finding the u_k that minimizes

$$E(\mathbf{u}, \mathbf{d}) = \sum_{k=1}^{K} \left[(u_k - d_k)^2 + (u_k - u_{k-1})^2 \right], \tag{5.1}$$

where the first term constrains the solution to be close to the data and the second term enforces spatial smoothness between neighboring values of u_k. Minimizing this least-squares formulation results in the smoothed surface shown in Figure 5.1b which does not preserve the spatial discontinuity in the data. To account for discontinuities and outlying

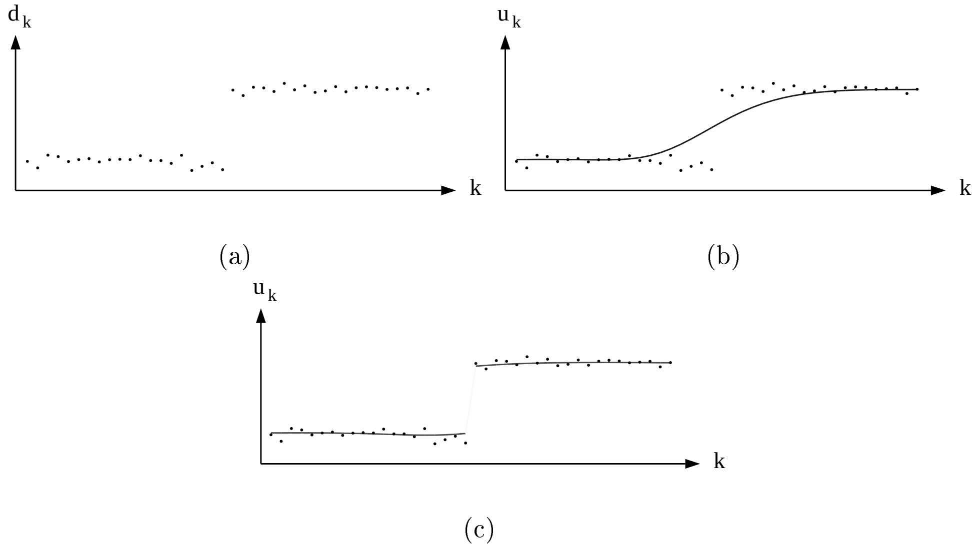


Figure 5.1: **Regularizing discontinuous data:** (a) Noisy data; (b) Least-squares fit to the data; (c) Piecewise smooth fit (robust regularization).

data measurements, we can convert this to the robust estimation problem [19, 39]

$$E(\mathbf{u}, \mathbf{d}) = \sum_{k=1}^{K} \left[\rho(u_k - d_k, \sigma_D) + \rho(u_k - u_{k-1}, \sigma_S)\right], \tag{5.2}$$

where ρ is a robust error function and the σ_i are scale parameters. We take ρ to be

$$\rho(x, \sigma) = \frac{x^2}{\sigma^2 + x^2} \tag{5.3}$$

which is used in [16, 20, 41] and is shown in Figure 3.1.[1] Minimizing the robust formulation in Equation (5.2) results in the piecewise smooth fit shown in Figure 5.1c.

5.2 Regularization with Transparency

The robust formulation can be extended naturally to cope with transparency. Consider the noisy data in Figure 5.2a. At each spatial position, k, there are multiple values, $d_{k,1}$ and $d_{k,2}$ which might, for example, be derived from depth measurements of two transparent surfaces. Fitting a single surface to this data using a least-squares formulation does not provide a useful solution as shown in 5.2b.

[1] The likelihood function in the previous chapter is related to this ρ function by $l(r,\sigma) = \frac{\sigma}{2r}\frac{\partial}{\partial r}\rho(r,\sigma) = \frac{\sigma\ \psi(r,\sigma)}{2r} = \frac{\sigma^3}{(\sigma^2+r^2)^2}$.

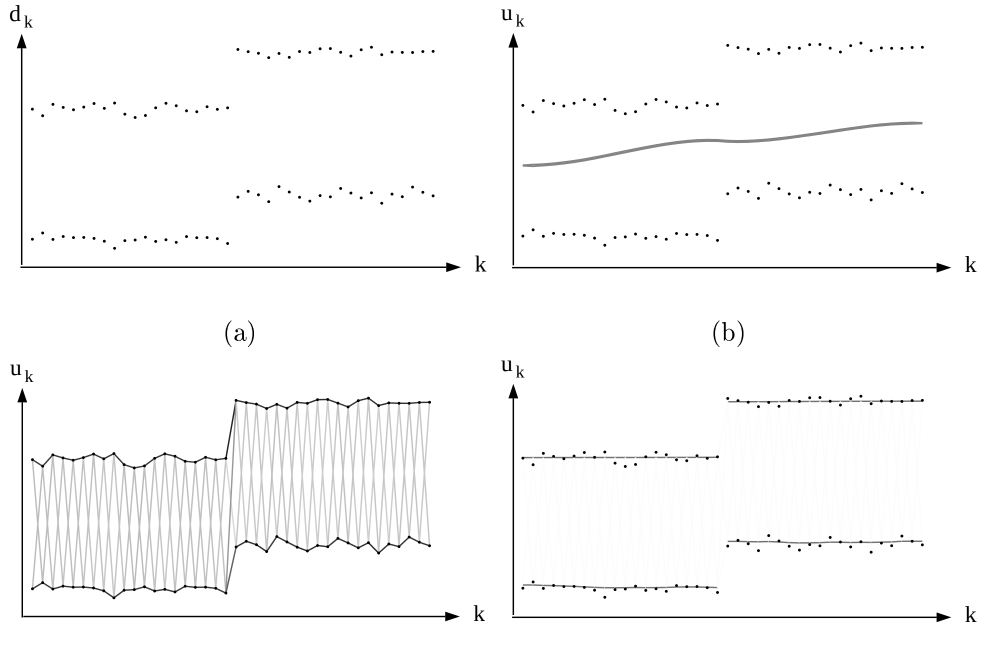


Figure 5.2: **Transparent regularization:** (a) Transparent data; (b) Single-layer regularization; (c) Weight of the connection between neighboring points in all layers (large σ); (d) Transparent regularization, piecewise smooth result (small σ).

Our goal is to regularize the measurements to derive two piecewise-smooth approximations $u_{k,1}$ and $u_{k,2}$ without knowing *a priori* which measurements are grouped with which other measurements. A given point $u_{k,1}$ has two neighbors to its left: $u_{k-1,1}$ and $u_{k-1,2}$. It is important to note that we do not know which, if either, of these measurements belongs to the same "surface" as $u_{k,1}$. If we knew the segmentation of the data points into surfaces, one could regularize the surfaces independently.

When the segmentation is not known *a priori*, we can still regularize by minimizing

$$E(\mathbf{u}, \mathbf{d}) = \sum_{k=1}^{K} \sum_{i=1}^{\mathcal{L}} \left[\rho(u_{k,i} - d_{k,i}, \sigma_D) + \sum_{j=1}^{\mathcal{L}} \rho(u_{k,i} - u_{k-1,j}, \sigma_S) \right], \tag{5.4}$$

with respect to each surface point $u_{k,i}$, where $\mathcal{L}$ represents the number of layers. This means that we smooth a point with respect to *all* its neighbors in all surfaces. If any of these points are similar, they will be treated as inliers by ρ and will have a strong influence on the solution. If they differ, they will be treated as outliers and will be *automatically* ignored. Minimizing Equation (5.4) smoothes the data without explicitly assigning data to particular layers.[2]

To illustrate this, Figure 5.2(c) shows the "weight" that the ρ-function gives to each neighbor. The dark lines indicate a strong connection between the surface points while the light lines indicate a weak connection. Note that we could threshold these values to derive a segmentation of the data into surfaces, but that there is no need to do this explicitly. As Equation (5.4) is minimized the values of σ_i are gradually lowered. At high values, more of the neighboring points receive a high weight, but as it decreases, outlying points receive lower and lower weight. Figure 5.2(d) shows the result of minimizing Equation (5.4) using gradient descent with a continuation method. The solution converges to the desired piecewise-smooth, and transparent, surface interpretation.

5.3 Optical Flow

In previous formulations in Section 3.3, this constraint is formulated to minimize the difference between neighboring single layer optical flow vectors $\mathbf{u}_i$ and $\mathbf{u}_j$. When the local flow estimation is performed with a mixture of affine bones, this traditional constraint is no longer applicable. The transparent regularization theory introduced above can be incorporated into the optical flow problem in a straightforward way to allow the regularization of multi-layer bones.

We modify the Skin & Bones model by adding a spatial coherence term to the multi-layer data term in Equation (4.21). The smoothness term is defined to examine all neighboring layers as described above. The new objective function for layer i of an image patch s is

$$\begin{aligned}
&E_i(\mathbf{a}_i(s)) = E_{D_i}(\mathbf{a}_i(s)) + E_{S_i}(\mathbf{a}_i(s)),\\
&E_{D_i} = \tfrac{1}{|\mathcal{R}(s)|}\textstyle\sum_{\mathbf{x}\in\mathcal{R}(\mathrm{s})} w_i(\mathbf{x},\sigma_i(s))(\nabla I\cdot\mathbf{u}(\mathbf{x},\mathbf{a}_i(s)) + I_t)^2,\\
&E_{S_i} = \tfrac{\lambda}{\sum_{\mathbf{y}\in\mathcal{G}(\mathrm{s})}|\mathcal{N}(\mathrm{s},\mathbf{y})|}\textstyle\sum_{\mathbf{y}\in\mathcal{G}(\mathrm{s})}\sum_{\mathrm{t}\in\mathcal{N}(\mathrm{s},\mathbf{y})}\sum_{l\in\mathcal{L}(\mathrm{t})}\rho(\mathbf{u}(\mathbf{y},\mathbf{a}_i(s)) - \mathbf{u}(\mathbf{y},\mathbf{a}_l(t)),\sigma_{skin}(s)),
\end{aligned} \tag{5.5}$$

where s is an image region, λ controls the relative importance of the two terms, $\mathcal{R}(s)$ are the pixels of region s, $\mathbf{a}_i(s)$ are the affine parameters of layer i in patch s, $\mathcal{G}(s)$ is the set

[2] Equation (5.4) can be reformulated as a weighted least squares problem much like Equation (4.12) and solved using an EM algorithm.

that contains the pixels at the boundaries of patch s, $\mathcal{N}(s, \mathbf{y})$ are the neighboring patches connected to patch s at pixel $\mathbf{y}$. The two terms of E (data and spatial) are normalized with respect to the size of $\mathcal{R}(s), \mathcal{N}(s, \mathbf{y})$ and $\mathcal{G}(s)$ respectively and each has its own scale parameter. Note that the use of a robust error norm, ρ, allows spatial discontinuities at the boundary of the region. We take ρ to be the function given in Equation (5.3).

The first term of Equation (5.5) is simply the multi-layer bone with spatial smoothness prior on the ownership weights $w_i(\mathbf{x}, \sigma_i(s))$ from Section 4.3. The smoothness term of "skin" is formulated to minimize the difference between optical flow vectors at the boundary of the region for *all* neighboring patches and for *all* layers presented in that patch. Motions that are similar will tend to reinforce each other while dissimilar motions will be ignored as outliers. Although patches all have the same number of layers for the experiments described in this chapter, different numbers of layers in neighboring patches are allowed. The smoothness term is computed with respect to any and all neighboring layers.

Note that the smoothness term of "skin" is different from the spatial smoothness prior defined in Section 4.3. The latter is applied within each multi-layer patch, while the former is applied between neighboring patches. Hereafter, we use *inter-patch smoothness* to refer to the regularization term "skin", and *intra-patch smoothness* to refer to the spatial smoothness prior on the ownership weights.

While Equation (5.5) may appear complicated, it can be minimized in exactly the same way as all the previous objective functions considered so far. We minimize this function using the same gradient descent scheme and σ estimating and annealing described in Section 3.3. This process alternates between solving for the $\mathbf{a}_i(s)$ in each layer taking into account the inter-patch smoothness term and solving for the weights $w_i(\mathbf{x}, \sigma_i(s))$.

The scale parameter of the inter-patch smoothness term, $\sigma_{skin}(s)$, is estimated and annealed in the same way as described in Section 3.1.3. Given all the observations in the smoothness term with respect to patch s, the estimated $\tilde{\sigma}_{skin}(s)$ is defined to be 1.4826 multiplied by the median absolute deviation of the observations. The annealing parameter $\hat{\sigma}_{skin}(s)$ starts with a large value (which is 5.0) in the first iteration and decreases by a fixed rate 0.95 at each iteration, while its final value in the last (30th in our experiments) iteration is the unit 1.0. The final scale parameter $\sigma_{skin}(s)$ is the multiplication of the estimated part and the annealing part. Notice that we use the same annealing rate for both the data term and the regularization term. λ is taken to be $\frac{\sigma(s)_S^2}{\sigma(s)_D^2}$. These parameters remain fixed for the experiments in this Thesis. Note that all the parameters of the multi-layer data term are the same as those used in the previous chapter.

Unlike traditional parametric motion estimation schemes, the addition of the spatial coherence constraint at patch boundaries means that each step in the non-linear optimization takes into account both the optical flow constraints within the region and the affine parameters of the neighboring regions. This results in more accurate motion estimates and a more stable optimization problem.

5.4 Experimental Results: Skin & Bones

The "Skin and Bones" method is applied to the following image sequences. Among the experiments shown in this section, the synthetic sequences are used to rigorously compare the estimated motion with the true motion, while real image sequences are used to show the reliability of the algorithm. All the experiments use the multi-layer "Skin and Bones" model with the intra-patch smoothness prior within each patch. The size of overlapping between patches is 8 pixels, and the size of non-overlapping basic patches is 32×32. We apply a similar minimization process as the one used in the single-layer case (Figure 3.17), but consider the differences between flow vectors at all boundary pixels for *all* neighboring patches and for *all* layers presented in that patch. Like in the previous chapter, the horizontal and vertical component of flow are generated by the layer which has the maximum ownership at each pixel position in each basic patch. Note that only motion layers are consider when we determine which layer has the maximum ownership, thus outliers are assigned to one of the motion layers also.

5.4.1 Synthetic Sequences

We revisit the following three synthetic sequences to compare the results of multi-layer "Skin and Bones" method with the results of multi-layer "Bones" which were shown in

(a) (b) (c)

Figure 5.3: **Circle Sequence: Skin & Bones;** (a) horizontal component of flow; (b) vertical component of flow; (c) weights for outlier layer.

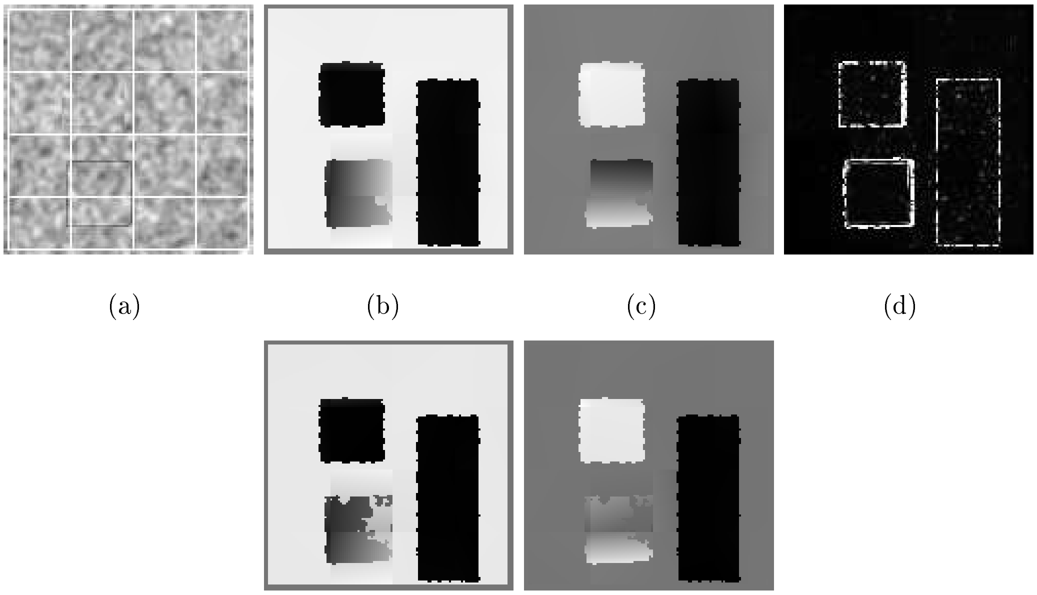
Chapter 4.

Circle Sequence

The Circle sequence was used to demonstrate the effect of overlapping in Figure 4.17. The result of the "Skin and Bones" method is shown in Figure 5.3. The horizontal and vertical component of flow are shown in Figure 5.3 (a) and (b), and the outliers, which occur only at the motion boundaries, are displayed in Figure 5.3 (c). Compare with the estimated flow without the skin term (the second column of Figure 4.17), an unstable patch at the top boundary disappears, and the vertical flow is much smoother.

Synthetic Bars Sequence

The Synthetic Bars sequence was used to illustrate the effect of the intra-patch smoothness prior in Figure 4.11. The result of the "Skin and Bones" method is shown in Figure 5.4 (a)-(d). The estimated flow without the skin term (see Figure 5.4 (e) and (f)) shows clearly blocked structure in the vertical and horizontal flow of the lower square. With the transparent regularization, the estimated horizontal and vertical velocities (see Figure 5.4 (b) and (c)) appear much smoother inside the bars, at the same time, the motion discontinuities are still clearly defined and smoothly connected between patches.



(a) (b) (c) (d)

(e) (f)

Figure 5.4: **Synthetic Bars Sequence: Skin & Bones;** (a) one image with segmented region shown; (b) horizontal component of flow; (c) vertical component of flow; (d) weights for outlier layer; (e) horizontal component of flow (bones only); (f) vertical component of flow (bones only).

Yosemite Sequence

The Yosemite sequence is used to show the statistics of angular measurements between the estimated flow vectors and the true flow vectors. The recovered optical flow using Equation (5.5) is shown in Figure 5.5 (a) and (d). Comparing the results to those in the second column of Figure 4.20, one can see that unstable results in these lower left patches are gone and that the flow appears smooth in the entire image. The results with and without skin term, for the single-layer case and two-layer case with and without the spatial prior on ownership weights, are compared quantitatively in Table 5.1. For the multi-layer methods, the addition of "skin" reduces the average angular error by 9.8% and 25% for the cases without and with the spatial prior on ownership weights respectively. Note that with the skin term, the errors for the two multi-layer cases are nearly identical, and the performance of the multi-layer cases is only slightly better than that of the single-layer method if we consider the percent of flow vectors with error less than 1°. These results basically agree with the fact that only one affine motion is present

Technique	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Black & Anandan [16]	0.232	3.44°	3.34°	10.5%	35.7%	58.0%	82.0%	96.0%
Bab-Hadiashar [8]	0.12	2.51°	2.58°	21.2%	54.6%	75.8%	91.2%	97.7%
Single-layer Bones	0.153	2.94°	2.58°	15.8%	44.7%	65.2%	85.8%	97.6%
Two-layer Bones (no prior)	0.114	2.34°	2.04°	25.3%	57.0%	75.2%	91.1%	98.8%
Two-layer Bones (prior)	0.133	2.78°	2.77°	22.0%	51.5%	70.0%	86.5%	97.0%
Single-layer S&B	0.098	2.11°	1.84°	30.5%	61.7%	78.7%	92.8%	99.4%
Two-layer S&B (no prior)	0.104	2.11°	1.85°	34.4%	62.0%	76.7%	92.2%	99.5%
Two-layer S&B (prior)	0.103	2.09°	1.83°	33.2%	62.3%	78.3%	92.7%	99.5%

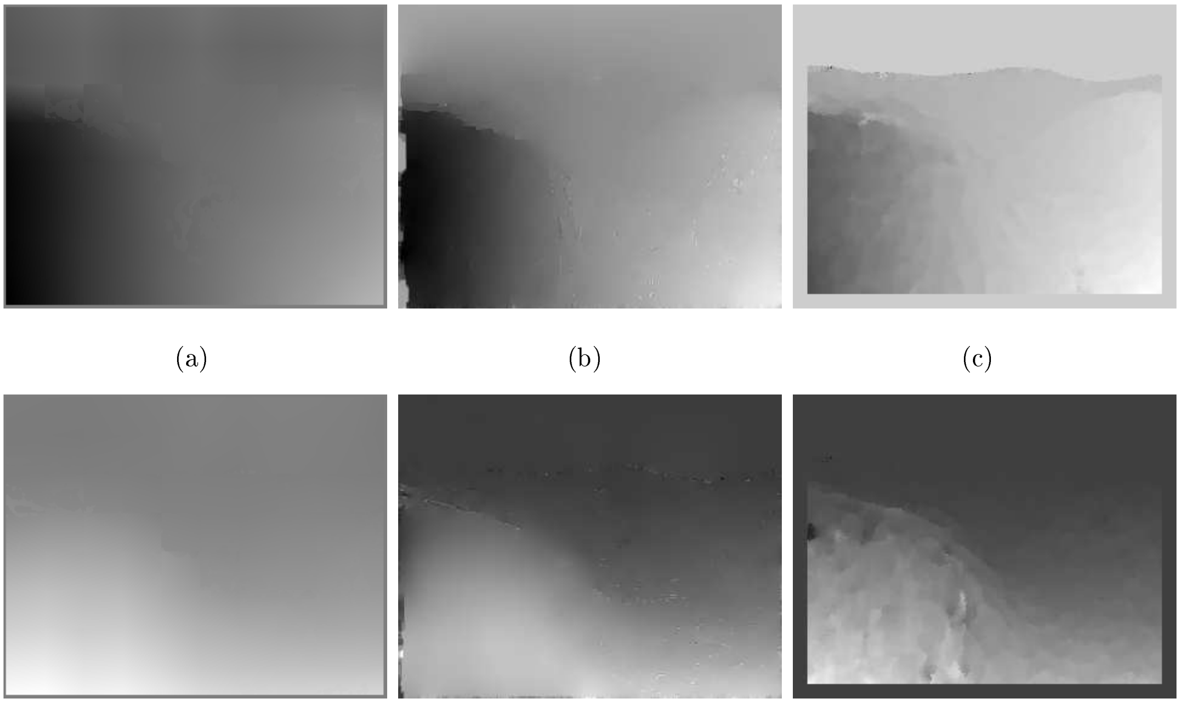
Table 5.1: Error results for the Yosemite fly-through sequence: multi-layer Skin and Bones.

in most patches.

The dense optical flow method proposed by Black and Anandan [16] has been widely acknowledged as one of the most reliable methods for computing general and piecewise smooth flows. Recently, Bab-hadiashar and Suter [8] proposed a robust dense optical flow method, which produced promising results for both the Yosemite sequence and the Marbled Block sequence. These two methods are among the best existing dense optical flow methods. To precisely compare the "Skin and Bones" method with these methods, we ran their optical flow algorithms[3], and used the same program that we used for the "Skin and Bones" method to compute errors. The error statistics of these two methods are shown in the first two rows of Table 5.1. The recovered horizontal and vertical velocities are shown in Figure 5.5 (b) (e), and (c) (f). Note that we used frame 11 for all the experiments of Yosemite sequence, only for the experiment of the method proposed by Bab-hadiashar and Suter, we used frame 7. Since their method requires multiple frames to compute the spatial and temporal derivatives, and the results are affected by the number of frames that are used. Frame 7 is the center frame of the Yosemite sequence, thus its performance is the best[4]. Also we clipped 5 pixels from all boundaries (except the top) to compute the errors statistics, except for the method of Bab-hadiashar and Suter, which did not estimate the flow in the 11-pixel wide boundary areas. By quantitative comparison, the performance of the "Skin and Bones" method is the best among the three dense methods. Note that both methods which use regularization and coarse-to-

[3]The code is available on line.

[4]Using frame 11, the mean angular error increases to 3.2°.



(a) (b) (c)

(d) (e) (f)

Figure 5.5: **Yosemite Sequence: Skin & Bones, and other methods;** (a) horizontal component of flow (Skin & Bones); (b) horizontal component of flow (Black & Anandan); (c) horizontal component of flow (Bab-hadiashar & Suter); (d) vertical component of flow (Skin & Bones); (e) vertical component of flow (Black & Anandan); (f) vertical component of flow (Bab-hadiashar & Suter).

fine techniques smoothed the flow at the boundary of sky (see first two columns of Figure 5.5). The method of Bab-hadiashar and Suter, however, provided a sharp discontinuity near this boundary.

5.4.2 Real Image Sequences

We evaluate the "Skin and Bones" method using various real image sequences in this section.

Marbled Block Sequence

The Marbled Block sequence contains many sharp discontinuities in both depth and motion. The recovered optical flow and the horizontal/vertical velocity using the "Skin and Bones" method is shown in Figure 5.5 (d), (b), and (c). Comparing it with the estimated flow without the transparent regularization (see Figure 5.5 (e) and (f)), the flow is much smoother by visual inspection. However, the skin term may not solve all the under-constrained cases, such as when a patch contains multiple objects where one of them has very little brightness variation. For example, in the patches at the top of the light block, only the background motion is recovered due to the lack of texture of the light block. Furthermore, since the regularization term is only applied at the patch boundaries, the flow inside a patch may appear uneven (see the patches that contain the left boundary of the front block[5]).

The angular error statistics are compared quantitatively in Table 5.2. Similarly to the Yosemite sequence, the addition of "skin" reduces the average angular error by 25% for the Marbled Blocks sequence. In Table 5.2, we also demonstrate the results of Black and Anandan [16], Bab-Hadiashar and Suter [8]. The mean angular error of the "Skin and Bones" method is comparable to that of the robust method of Bab-Hadiashar and Suter, which is the smallest one. Also, the standard deviation of our method is the lowest among all the experiments.

[5]In these patches, there is very little texture, thus it is not reliable to estimate two motion layers. Note that we use the intra-patch smoothness prior on ownership weights, but one motion layer is still estimated incorrectly. Therefore, selecting the appropriate number of layers is an important issue, which will be addressed in the following chapter.

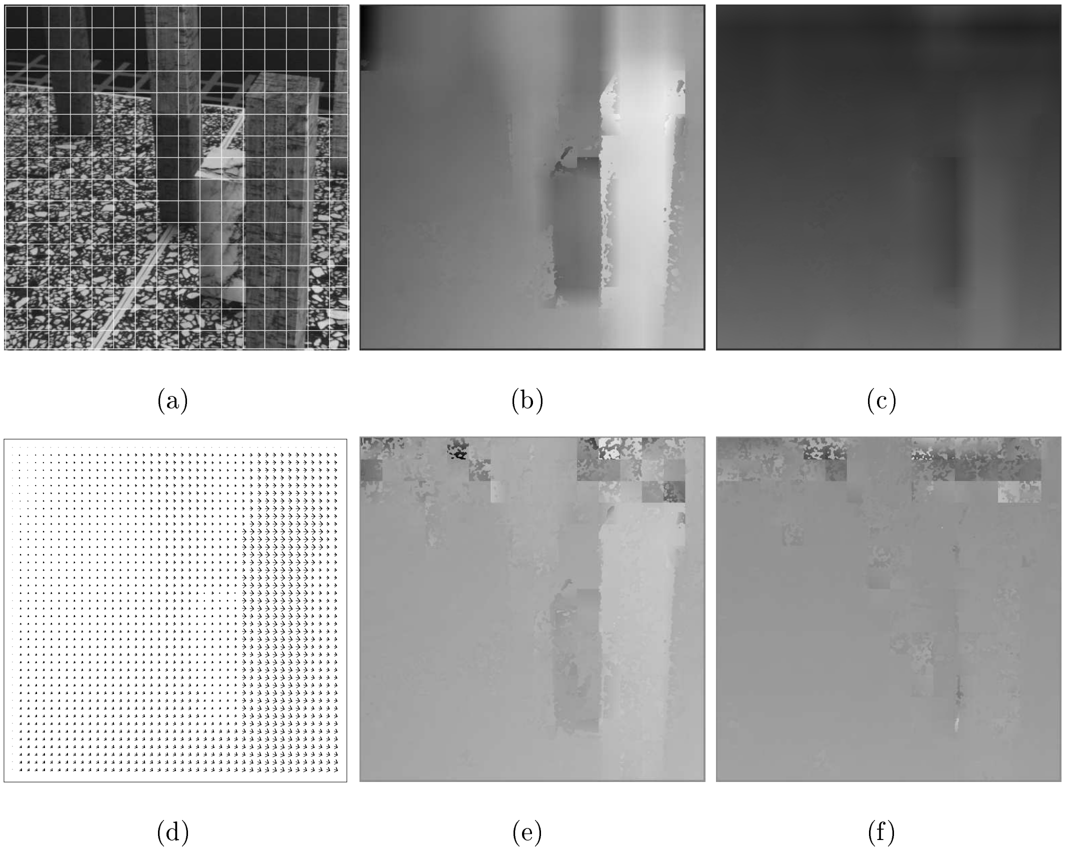
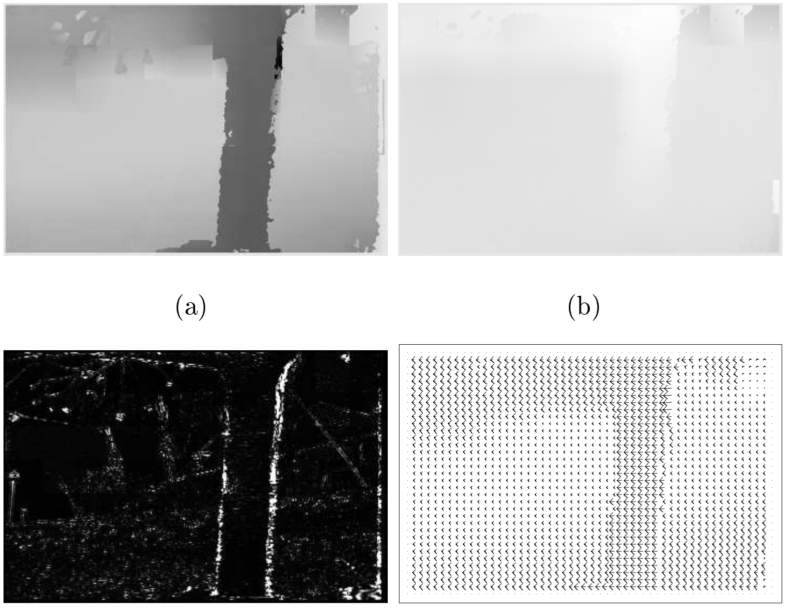


Figure 5.6: **Marbled Block Sequence: Skin & Bones;** (a) one image with segmented region shown; (b) horizontal component of flow; (c) vertical component of flow; (d) vector field; (e) horizontal component of flow (without skin); (f) vertical component of flow (without skin).

	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
			$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Black & Anandan [16]	4.04°	4.38°	9.4%	29.0%	56.1%	85.0%	90.4%
Bab-Hadiashar [8]	3.36°	4.28°	2.9%	30.9%	78.6%	90.6%	94.5%
Single-layer Bones	4.08°	4.96°	10.5%	33.8%	60.6%	82.4%	90.9%
Multi-layer Bones	4.59°	6.10°	9.7%	32.1%	57.0%	78.7%	89.3%
Single-layer Skin&Bones	3.44°	4.00°	11.8%	37.4%	67.7%	88.2%	92.6%
Multi-layer Skin&Bones	3.44°	3.89°	7.9%	39.0%	68.9%	87.8%	92.6%

Table 5.2: **Marbled Block Sequence: error results**; Skin&Bones.



(a) (b) (c) (d)

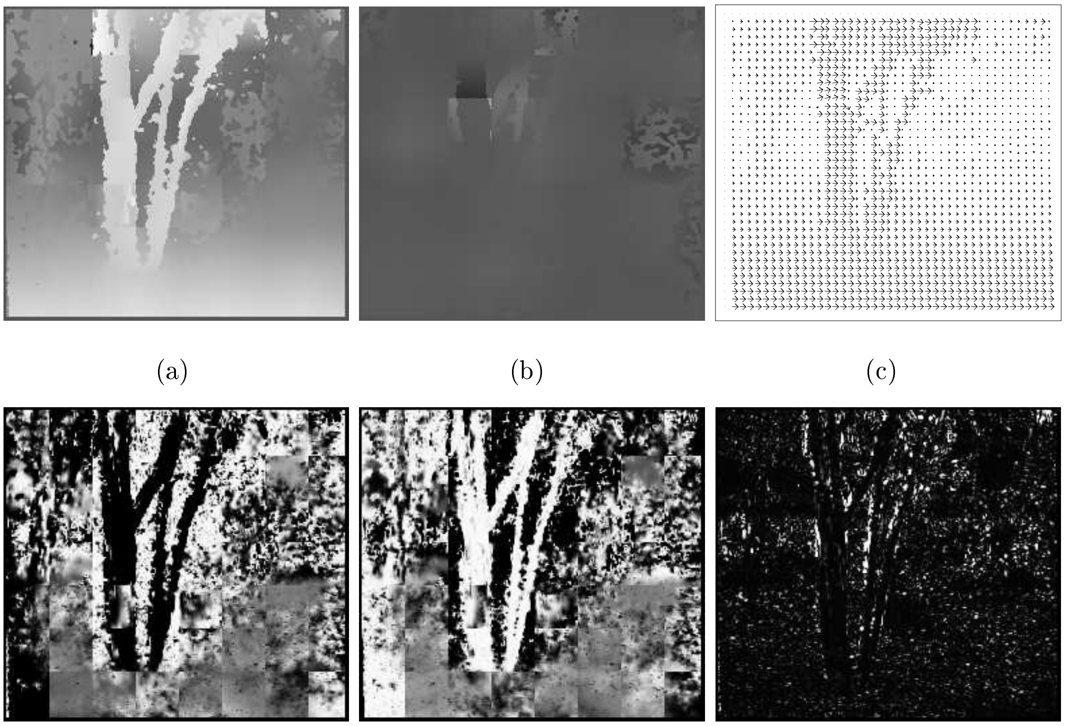
Figure 5.7: **Flower Garden Sequence: Skin & Bones;** (a) horizontal component of flow; (b) vertical component of flow; (c) weights for outlier layer; (d) vector field.

Flower Garden Sequence

Figure 5.7 shows the results of the Flower Garden sequence. We have visited the sequence several times in the previous two chapters. Recall how the single-layer model method recovered the dominant motion within a patch (Figure 3.25). With the multi-layer extension, the method of mixture of affine layers can recover multiple motions simultaneously. Yet in some regions at the tree brunches, the motion estimation problem is under-constrained, thus the method can not recover two layers reliably (Figure 4.21). With the transparent regularization, in the regions bordering the tree two distinct motions are recovered, which are also smoothly connected to their neighbors.

SRI Tree Sequence

The SRI Tree Sequence is another sequence that illustrates the effect of regularization. The estimated flow field is shown in Figure 5.8. Comparing it with the result of multi-layer bones (see Figure 4.22), we can see that the flow of the ground plane is much smoother. Figure 5.8 (d) and (e) show the weights for the two motion layers within each region. Gray areas correspond to weights near 0.5, and these regions indicate places where

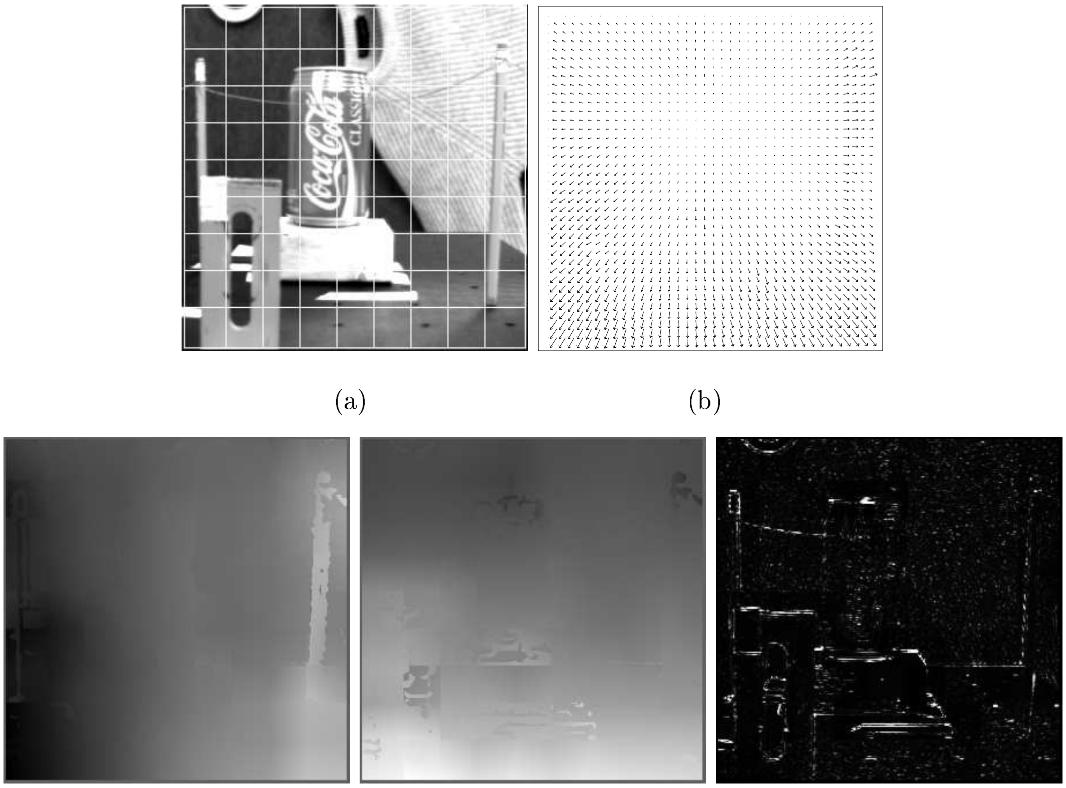


(a) (b) (c)

(d) (e) (f)

Figure 5.8: **SRItree Sequence: Skin & Bones;** (a) horizontal component of flow; (b) vertical component of flow; (c) vector field; (d) weights for layer one; (e) weights for layer two; (f) weights for outlier layer.

essentially one motion was present. Examining the weights indicates that the ground plane is essentially treated as a single layer. Regions that span a motion boundary have two distinct sets of weights. One portion of the region has high weights (white areas in the figure) while the other has low weights within a particular layer. This pattern is reversed in the other layer. The branches of the trees and the background are assigned to different layers when they both appear in the same region. Outliers occur at the boundary between the tree branches and the background. Recall that the multi-layer bones method with the spatial prior on ownership weights is unlikely to converge to a single motion. With the regularization term, the motion estimation method is more stable in a patch that contains a single layer.



(a) (b)

(c) (d) (e)

Figure 5.9: **Nasa Sequence: Skin & Bones;** (a) one image with segmented region shown; (b) vector field; (c) horizontal component of flow; (d) vertical component of flow; (e) weights for outlier layer.

NASA Coke Sequence

The NASA Coke sequence is similar to the Marbled Block sequence where the motion is primarily dilational. Figure 5.9 shows the result obtained with our algorithm. Note that the method also recovers the motion of right pole with sharp discontinuities at the motion boundaries. In some patches, due to the single oriented motion constraints, the vertical flow is not estimated properly (see Figure 5.9 (d)).

Hamburg Taxi Sequence

The next image sequence captures a different situation, where the camera is static but three vehicles are moving independently. In Figure 5.10 we give the results of the "Skin and Bones" method for the Hamburg Taxi sequence. Since the road contains very little texture, our method did not recover the motion boundaries of the white car correctly

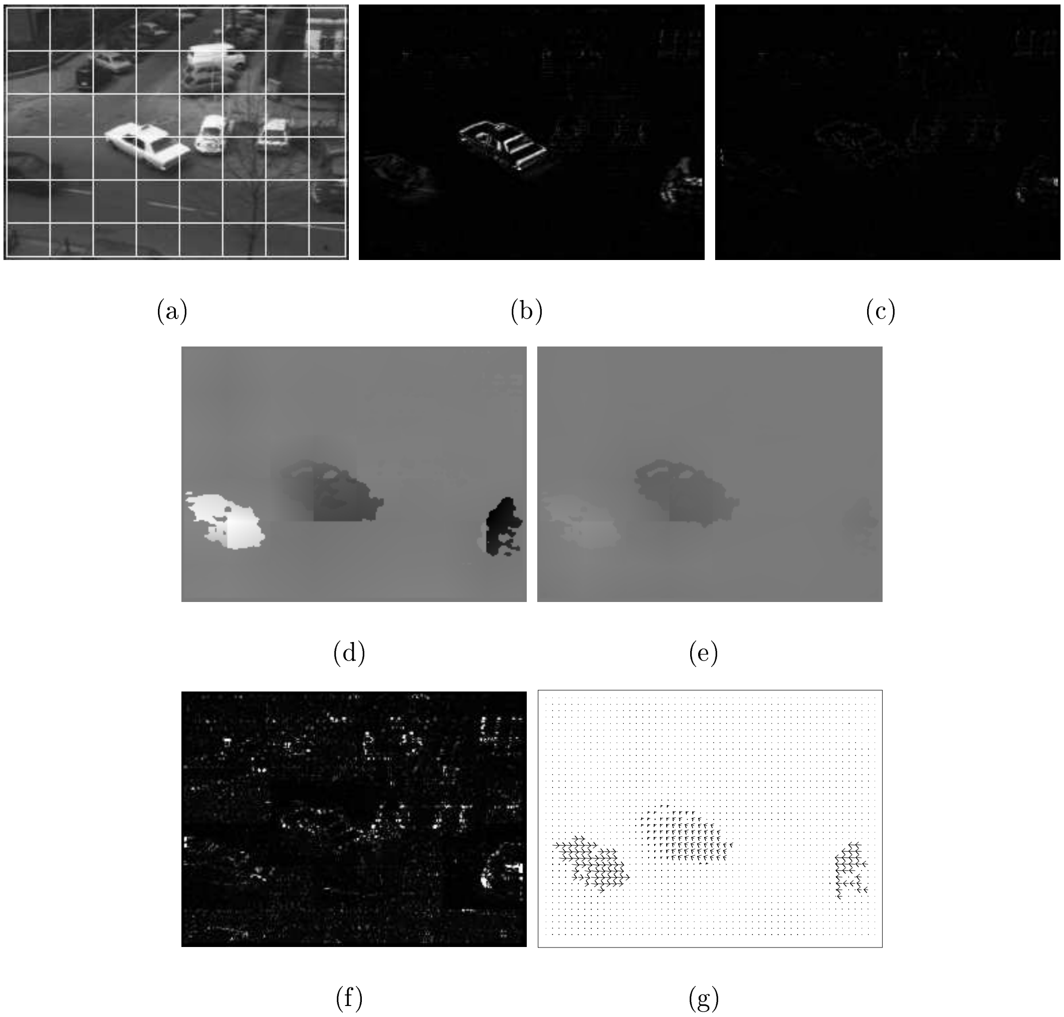


Figure 5.10: **Taxi Sequence: Skin & Bones;** (a) one image with segmented region shown; (b) absolute difference between the two original images; (c) absolute difference between the stabilized frame and the second image. (d) horizontal component of flow; (e) vertical component of flow; (f) weights for outlier layer; (g) vector field.

(see the horizontal flow Figure 5.10 (d)). The quality of the recovered flow is often measured in terms of how well it is able to stabilize the sequence through warping. We use the absolute difference of two images to illustrate the effect of stabilization. For better contrast, the value is scaled into the range from 0 to 255. The difference images between the original frames and the stabilized frames are shown in Figure 5.10 (b) and (c) respectively. We can see that all three vehicles and the background are well stabilized.

In summary, we list the three main advantages of the "Skin and Bones" model:

- Parameterized motion estimation within local patches can recover accurate optical flow estimates;

- Multi-layer estimation with a mixture of parametric motion models can induce the proper recovery of optical flow in patches that contain multiple objects, particularly when motion boundaries are present.

- The spatial smoothness prior on the ownership weights reduces the influence of the "leverage" points and gives rise to a more stable motion estimation process.

- The transparent regularization term can result in a stable optimization problem and more accurate motion estimates, particularly when the patches do not contain sufficient brightness variation.

Chapter 6

Estimating the Number of Layers

Generally, we can hypothesize a large number of "Skin and Bones" models that can produce the same optical flow field. Consider the example show in Figure 6.1, where four thin bars move toward the center in the horizontal dimension. Two possible descriptions of the motion are listed, where the second one is a single linear model and the first one contains four constant models corresponding to each bar. The basic problem addressed in this chapter, therefore, is to define criteria by which we can select a unique model to represent the motion of given images, and to specify a computationally efficient algorithm for finding this model. In this chapter, we show how to use the Minimum Description Length (MDL) Principle to search for the best representation. In Section 6.1, we present some of the information theory background behind this principle. Then in Section 6.2, we apply the MDL principle to select the number of layers within image patches. Section 6.3 describes the revision process which is used to find the most appropriate number of layers presented in a patch. Finally, in Section 6.4, experimental results will be demonstrated.

6.1 Minimum Description Length Principle

In previous formulations of the "Skin and Bones" method, we allowed multiple parametric motion models to describe the data, but we did not address the fundamental question of how many models to use. Instead, a mixture of two affine models was estimated in each patch, and where only a single motion was present both layers converged to the same affine motion. A critical issue is deciding the appropriate level of model complexity to use in the representation. For estimating multiple motions using parameterized methods

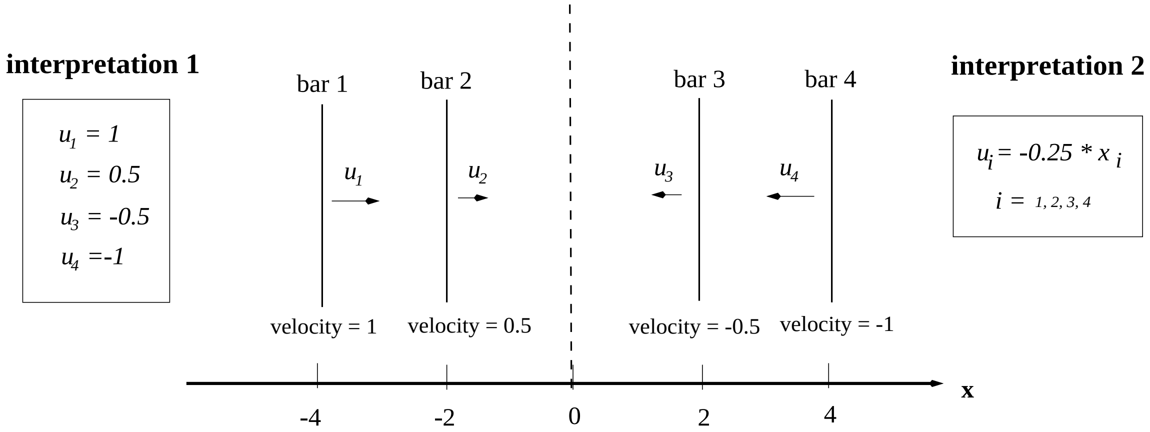


Figure 6.1: Multiple description of motion data.

such as the one presented in Chapter 4, model complexity will depend on: 1) the number of layers that are used to describe the motion; and 2) the order of the model that is used to represent each motion layer, where common choices of motion models include constant flow, affine flow, and planar flow.

Maximum likelihood estimation described in Chapter 4 provided a means for finding the optimal parameters when the model complexity is fixed, but will not help in deciding how many models to use, or how to compare the performance of models of various orders. We need a criterion to balance model complexity with model accuracy. Unfortunately, finding the number of components in a mixture model is a very difficult problem which has not been completely resolved [72]. In this section, we explore the problem of finding an optimal representation in a Minimum Description Length (MDL) paradigm [83].

The *minimum principle* implies an intuitive idea that simpler models are better than more complex ones. Therefore, the best representation is the one that explains the data with the simplest model. This principle is commonly applied in science, and is often used to eliminate overly complicated hypotheses. According to the information theory, the number of bits, which are required to reproduce the observations from the model, can be used to describe the simplicity of a hypothesis. Based on *minimum principle*, Rissanen [83, 84] proposed the Minimum Description Length principle as an estimation criterion. It is an approach of searching the model with the shortest *total code length*, i.e., the number of binary digits required to encode both the data and the model itself. Such a model defines a distribution which assigns the maximum probability to the observed data, and therefore it may interpreted as the "most likely explanation" of the observations. MDL unifies the estimation of integer-valued structure parameters which specify model order and type, and real-valued parameters which specify a model for the data source. In contrast, Bayesian estimation, in general, uses prior probability densities that are not related to minimizing the description length of the data.

From an information theory viewpoint, for any information source with positive probability function P, the code length for all strings will be close to the ideal $-\log P(x)$[1], which is referred as *ideal code length.* According to Shannon's probabilistic notion of information, the average of $-\log P(x)$ over all data realizations is the well-known entropy:

$$-\sum_{t=1}^{n} P(x = x_t) \log P(x = x_t) \tag{6.1}$$

Shannon defined "entropy" to be the lack predictability between elements in a representation. If there is some predictability from one element to another, then entropy is not at its maximum, and a shorter encoding can be constructed. When the encoding can not be compressed further, the resulting signal consists of "pure information". Thus if we find the representation with the shortest possible encoding, in some sense, we have found the information in the image.

The original MDL criterion minimizes the code length:

$$\mathcal{CL}(x, \theta) = \mathcal{CL}(x|\theta) + \mathcal{CL}(\theta) = -\log P(x|\theta) + \mathcal{CL}(\theta) \tag{6.2}$$

where $\mathcal{CL}(x, \theta)$ is the description length function, or *ideal total coding length.* $\mathcal{CL}(x|\theta)$ represents the bits used to encode the probabilistic model $P(x|\theta)$ which describes the data. The term $-\log P(x|\theta)$ represents the well-known lower bound on achievable prefix or arithmetic code lengths [28]. $\theta = \theta_1, \cdots, \theta_k$ denotes a k-component parameter vector ranging over a subset Ω^k of the k-dimensional Euclidean space. The term $\mathcal{CL}(\theta)$ is the description length for the parameter vector. The data items x_t in $x = x_1, \cdots, x_n$ range over a finite or countable set.

The code length depends on the precision selected for the parameters. The bits that are used to encode parameters decrease if we use coarser precision, however, the first term

[1] The base two logarithm is used in this chapter.

in Equation (6.2) generally increases since the truncated parameter vector can deviate more from the non-truncated optimal value θ. Consider the parameters to be truncated numbers to some precision, say θ_j to the precision $\delta_j = 2^{-q_j}$, where q_j is the number of fractional binary digits taken in the truncation. Assume the parameters vary uniformly over some interval $[a_j, b_j)$, the code length defined in Equation (6.2) becomes:

$$\mathcal{CL}(x, \theta) = -\log P(x|\hat{\theta}) - \sum_{j=1}^{k} \log((b_j - a_j)\delta_j) \tag{6.3}$$

where $\hat{\theta}$ is the truncated parameter vector. Additional bits are needed to specify both the required precision of the parameters, as well as the dimension, or model order, of the parameter vector.

The above formulation is defined without any assumption about the nature of the parameters. These parameters are referred to as the nuisance parameters. That is, the prior distribution $\pi(\theta)$ of parameters is unknown. The MDL criterion determines the nuisance parameters so that the shortest code length for the data together with the nuisance parameters themselves results. When the "true priors" are not obvious, the minimal encoding framework provides us with a method of approximating them: we pick up the best practical representation we have available. As pointed out by Leclerc [65], this method is useful in vision problems because it gives us a way to produce estimates using models that are too complex for calculation of direct priors.

It is worth mentioning that MDL has been recently applied to computer vision problems such as image segmentation [31, 65], motion segmentation [4, 111], and 3D rigid motion segmentation [44].

6.2 Encoding of the Multi-layer Bones

In the case of multi-layer bones, the data within a patch (i.e., the residual errors) is described by a probability distribution $P(\mathbf{x}|\hat{\theta}, \xi)$, where $\hat{\theta}$ represents the truncated affine parameters of each layer, and ξ is the *model structure*. In the case of multi-layer bones, the *model structure*, ξ, consists of a specification of the layer assignment at each pixel. We let $\mathcal{O}_n, n = 1, \cdots, \mathcal{L} + 1$ denote the partition of the data, where $\mathcal{O}_n$ consists of all pixels that are assigned to layer n and layer $\mathcal{L} + 1$ stands for the outlier layer. Implicit in $(\hat{\theta}, \xi)$ is thus the number of layers $\mathcal{L}$. Here we define the total code length for one multi-layer bone, s, to be:

$$\mathcal{CL}(s) = \sum_{n=1}^{\mathcal{L}(s)+1} -\log P(\mathbf{x}|\theta(s), \mathbf{x} \in \mathcal{O}_n(s)) + \mathcal{CL}(\theta(s)) + \mathcal{CL}(\xi(s)) + \mathcal{CL}(\mathcal{L}(s)), \tag{6.4}$$

where $\mathcal{CL}(\mathcal{L}(s))$ refers to series:

$$\log \mathcal{L}(s) + \log\log \mathcal{L}(s) + \log\log\log \mathcal{L}(s) + \cdots, \tag{6.5}$$

where the sum continues until the last positive term. $\mathcal{CL}(\mathcal{L}(s))$ represents the ideal coding length function of an integer $\mathcal{L}(s)$ [84]. In most of what follows, for notational simplicity we will omit the explicit dependence on region s. In the following part of this section, we describe the coding formula of each part of Equation (6.4).

There are two major advantages of the MDL approach. First, the MDL approach is able to combine purely stochastic models (such as noise) with deterministic models (such as polynomials). Second, the MDL approach can be applied to both integer-valued structure parameters and real-valued model parameters. Using MDL, we can estimate the least number of bits that are needed to encode the observed data with regard to a particular data model.

6.2.1 Encoding of Affine Models

As a rough approximation, we assume the motion parameters a_0 and a_3 (horizontal and vertical translation) come from a uniform distribution between -16 and 16 pixels per frame, with a resolution of 1/100th of a pixel. The affine parameters (a_1, a_2, a_4, a_5) are chosen between -0.5 and 0.5 with a resolution of 1/10000th. We also assume an independent uniform prior for each coefficient. Therefore, the distribution functions for the truncated affine vector $P(\hat{\mathbf{a}}) = \sum_{i=0}^{5} P(\hat{a}_i) = (\frac{1}{32*100})^2 + (\frac{1}{10000})^2$, where $\hat{a}_i$ denotes the truncated parameter a_i given the precision of encoding. According to coding theory, the optimal coding cost is defined to be $-\log P(\hat{\mathbf{a}})$. We use the same coding scheme for the affine parameters of each layer. This gives an encoding cost for the six affine motion parameters of $\mathcal{L}$ layers:

$$\mathcal{CL}(\theta) = \mathcal{L} * (-2\log\frac{1}{3200} - 4\log\frac{1}{10000}). \tag{6.6}$$

We should mention that this differs from the classic MDL formula [84], in that $\mathcal{CL}(\theta)$ is defined to be $\frac{6\mathcal{L}}{2}\log n$, where n is the number of observations. The classic formula is an asymptotic form derived for the general case, without any knowledge of the parameters. It is done by first truncating the vector and then converting the result to an integer, which in turn is encoded with the length defined in Equation (6.5). Rissanen [84] derived the classic MDL formula based on an analysis of the optimal precision in the sense of minimum code length. In the case of encoding an affine transformation (Equation (3.2)), we have additional knowledge in determining the precision for encoding. First, parameter a_i has a natural bound, since the motion between a consecutive pair of frames in a video is not arbitrarily large. Second, affine parameters (a_1, a_2, a_4, a_5) are the coefficients to the coordinates (x, y), and they are in general much smaller than the motion parameters a_0 and a_3. Therefore, we encode affine parameters with finer precision. With this prior knowledge, we use fixed precision to encode the parameters in Equation (6.6).

Furthermore, more complex motion models may be used, which may result in smaller residual errors at the cost of more parameters to encode. For computational considerations, we only consider the case of fixed-order motion models in this Chapter.

6.2.2 Encoding of Model Structure

The *model structure* contains the information of which layer each pixel is assigned to. Given the number of motion layers $\mathcal{L}$, each pixel is assigned to one of the $\mathcal{L}+1$ layers, where the last layer stands for the outlier layer. At one image location, let P be the probability distribution, where p_i is the probability of this pixel belonging to layer i, $i \in 1, \cdots, \mathcal{L}+1$. Then according to coding theory, the average coding cost of the model structure at this pixel is given by $-\sum_{i=1}^{\mathcal{L}+1} p_i \log(p_i)$. If we ignore spatial correlations, then the total optimal coding cost of model structure in one patch is:

$$\mathcal{CL}(\xi) = -|\mathcal{R}| \sum_{i=1}^{\mathcal{L}+1} p_i \log(p_i), \tag{6.7}$$

Where $|\mathcal{R}|$ is the number of pixels in region $\mathcal{R}$.

Recall that in Chapter 4, the flow constraint at any given pixel $\mathbf{x}$ is assigned to the i^{th} layer with an ownership weight $w_i(\mathbf{x})$. Therefore, $w_i(\mathbf{x})$ can be used to estimate p_i.

We take the p_i to be the average of the ownership weights considering every pixels in the region, i.e.,

$$\mathcal{CL}(\xi) = -\sum_{i=1}^{\mathcal{L}+1}\left[(\sum_{\mathbf{x}\in\mathcal{R}} w_i(\mathbf{x})) * \log(\frac{\sum_{\mathbf{x}\in\mathcal{R}} w_i(\mathbf{x})}{|\mathcal{R}|})\right], \tag{6.8}$$

Note that by definition $0 * \log(0) = 0$. Consider fitting one motion layer to a patch, let $p_1 = 0.7$, $p_2 = 0.3$ be the probabilities computed from the ownership weights, the bits needed to encode the model structure are $|\mathcal{R}| * 0.88$. If the number of layers increases to two, and new probabilities are $p_1 = 0.6$, $p_2 = 0.38$, and $p_3 = 0.02$, the encode length also increases to $|\mathcal{R}| * 1.08$. The two-layer model is considered to explain the image motion better only if more bits will be saved in encoding residual errors.

6.2.3 Encoding of the Residual Errors

The last part consists of the encoding of pixel intensity $I(\mathbf{x}, t)$ of the second image given the first image in a pair of consecutive images, the layer i, and the truncated affine motion model $\hat{\mathbf{a}}_i$. While the image at time t can be approximated as a warped version of the image at time $t-1$, we must also encode the residual errors, $r(\mathbf{x}; \hat{\mathbf{a}}_i)$. The residual r only needs to be encoded to within -0.5 and 0.5 grey levels [2], giving discrete values for r^k. We encode r^k between $-2.5\sigma_i$ and $2.5\sigma_i$ using the prior probability distribution for r,

$$p(r^k(\mathbf{x}; \hat{\mathbf{a}}_i) \mid \sigma_i) = \frac{2}{\pi}\, l(r^k(\mathbf{x}; \hat{\mathbf{a}}_i),\ \sigma_i)\ \delta r \quad \mathbf{x} \in \mathcal{O}_i \tag{6.9}$$

where $\delta r = r^{k+1} - r^k = 1$. We also assume an independent prior probability distribution for r at all image positions. This gives the cost to encode a residual at pixel $\mathbf{x}$ in layer i:

$$\begin{aligned} -\log P(\mathbf{x}|\hat{\mathbf{a}}_i, \mathbf{x} \in \mathcal{O}_i) &= \sum_{\mathbf{x}\in\mathcal{O}_i} -\log(p(r^k(\mathbf{x}; \hat{\mathbf{a}}_i) \mid \sigma_i)) && (6.10)\\ &= \sum_{\mathbf{x}\in\mathcal{O}_i}\left[2\log(\sigma_i^2 + r^k(\mathbf{x}; \hat{\mathbf{a}}_i)^2) - 3\log(\sigma_i) + \log(\frac{2}{\pi})\right]. && (6.11)\end{aligned}$$

where $\mathcal{O}_i$ consists of all pixels in region $\mathcal{R}$ that are assigned to layer i, and $i \in 1, \cdots, \mathcal{L}$.

Finally, for the outlier layer,

$$-\log P(\mathbf{x}|\mathbf{x} \in \mathcal{O}_{\mathcal{L}+1}) = |\mathcal{O}_{\mathcal{L}+1}| * 8, \tag{6.12}$$

[2] We assume that the intensity values of an image are saved as integers, therefore it is enough to use 0 to specify any residual that is between -0.5 and 0.5.

where $|\mathcal{O}_{\mathcal{L}+1}|$ the number of pixels in the outlier layer of region $\mathcal{R}$. The cost of encoding an outlier pixel is just the cost of encoding the gray-level (integer) value of that image pixel and hence we take the cost to be eight bits. Intuitively, the code length of an inlier should be smaller than that of an outlier. Therefore, bits that are needed to encode the residual errors should not be more than eight. The cost to encode the maximum residual, $2.5\sigma_i$, is:

$$-\log(\frac{2}{\pi}\frac{\sigma_i^3}{(\sigma_i^2+(2.5\sigma_i)^2)^2}) \approx 6.3674581 + \log\sigma_i$$

For the above number to be less than 8, σ_i must be less than 3.1. On the other hand, the cost to encode the minimum residual 0 is $\log(\frac{\pi*\sigma_i}{2})$. To let this number be greater than 0, σ_i must be greater than $\frac{2}{\pi}$. These upper and lower bounds are used to constrain the estimated $\tilde{\sigma}_i$ at each iteration to be,

$$\begin{aligned} \tilde{\sigma}_i &= \max(\min(\tilde{\sigma}_i, \sigma^{(k)}), \frac{2}{\pi}) \\ \sigma^{(k)} &\longleftarrow 3.1 + k * 0.1 \end{aligned}$$

where $\sigma^{(k)}$ denotes the upper bound of the estimated sigma $\tilde{\sigma}_i$ when k iterations are left. $\tilde{\sigma}_i$ is described in Section 4.2.3.

6.3 Incremental Revision Process

The minimization of the total coding cost defined in Equation (6.4) with respect to the parameter vector θ is inherently a combination of two problems: parameter estimation and hypothesis testing. Standard steepest descent-based optimization techniques are thus not applicable. In addition, exhaustive search of the parameter space is computationally infeasible even for small-sized images. In practice, we have found that it is sufficient to estimate the parameters given a fixed model complexity (i.e., the number of motion layers in our case), then apply MDL to find the optimal code length for different model complexities.

We consider an incremental approach that will add a new layer if the revision can improve the motion estimates significantly. In our implementation, the MDL principle is used to compare recovered "Skin and Bones" models with differing numbers of layers.

We will choose the number of layers within each patch that have the minimum encoding cost while explaining the observations best. For this purpose, the number of the bits required to encode the multi-layer bones defined in Equation (6.4) is computed.

The incremental revision process starts with one layer in each patch and the motion for that layer is estimated using the "Skin and Bones" method. A new layer is then added and revised motion estimates are computed. If the encoding length of the revised model is smaller than that of the previous one, then the old one will be discarded. This revision process is carried out in every patch in the image until none of the regions improve, or the maximum number of layers is reached. In our experiments, we set the maximum number of layers to be 10, which is more than sufficient given the size of the patches.

6.4 Examples

Experiments have been carried out to test the MDL framework for selecting the best description of the scene. Here, we present results on the problem of motion-based segmentation that uses a globally layered model (i.e., the entire image region is used as a single patch), and on the problem of optical flow estimation that uses the "Skin and Bones" model. For all experiments, no distortion is allowed during the coding process, which corresponds to a precise reconstruction of the second frame from the first frame, the affine models, residual errors, and the layer assignments at each pixel.

6.4.1 Globally Layered Model

Our focus in these experiments is on the problem of estimating the correct number of layers in the entire image region. Layered affine motions are estimated within a big and global patch using the method described in Chapter 4 with the spatial smoothness prior on the ownership weights.

Textured Circles Sequence

The first experiment is conducted on the 256×256 pixel Textured Circles sequence. Each row but the last one in Figure 6.2 shows the recovered weights of each layer for different number of layers starting from one. When only one affine layer is estimated, the motion of background is recovered, and all textured pixels inside the two circles are considered as outliers. In the two-layer case, the motions of background and the upper circle are recovered simultaneously, and the textured pixels that belong to the other circle are treated as outliers. The MDL process finds that the cost of encoding the two-layer model is lower than that of the one-layer model, therefore the two-layer description is the better one and the revision process continues with one more layer added. Three layers corresponding to the background and the two circles are recovered correctly for the three-layer case. Outliers occur primarily at the occlusion and disocclusion boundaries. The bits that are required to encode the three-layer model are less than that of the two-layer model, thus a revision with a four-layer model is demanded. The additional layer estimated in the four-layer case has very little support from pixels near the motion boundaries. According to the MDL criterion, the three-layer model describes the scene better than the four-layer model, hence the revision process stops. In the last row of Figure 6.2, final results are presented with the texture map of each layer and the horizontal component of the flow field. The final coding cost is about 1.98 bits per pixel (Bpp).

Synthetic Bars sequence

The second experiment is performed on the Synthetic Bars sequence. The results are shown in the same way as that of the Textured Circles sequence. Figure 6.3 shows the recovered weights of each layer from the one-layer case to the five-layer case. In the four-layer model, the background and the three bars each correspond to one recovered layer, and it is deemed to be the best description of the scene. Final results are shown in the last row of Figure 6.3, where the texture maps of the four layers and the horizontal component of flow are displayed. The coding cost of this sequence is about 2.88Bpp.

Plant Sequence

Next, the algorithm is tested on a real image sequence. The image at the lower right corner of Figure 6.4 shows a frame of the sequence containing two plants in the foreground with a person moving in the background behind them. The person is occluded by the plant's leaves in a complex manner. Figure 6.4 shows the revision process of the MDL framework. In the final results, two layers were selected and estimated, one for the person and one for the two plants and the wall background. Most of the person has been correctly included in the second layer despite the occlusion caused by the plant's leaves. Also, the coding cost is approximately 2.7Bpp.

Flower Garden Sequence

Figure 6.5 shows the results for the Flower Garden sequence. The MDL framework automatically selects two layers, which correspond to the tree and the flower garden with background houses respectively. When a three-layer model is used, the ground plane of flowers and the houses in the background are separated into two layers. Although this description captures more information in the scene, it does not reduce the coding cost. We can see that the outliers do not diminish significantly after adding a third layer. Therefore, the saved bits to encode residual errors are not enough to compensate for the extra bits that are required to encode the structure of the layer assignments. Since this is a real image sequence with complicated textures, the coding cost is much higher than those of the synthetic sequences. The final codec performance of the Flower Garden sequence is 4.94Bpp.

SRI Tree Sequence

The next experiment is run on the SRI Tree sequence, which contains many depth discontinuities, not only at the boundaries of the tree but also in the background. Figure 6.6 shows the results of the MDL framework, where two layers are chosen to represent the scene, one layer for the background, and the other layer for the tree. For the three-layer case, a new layer which corresponds to a distant tree at left is recovered. Due to the increase of the coding cost, this layer was discarded. The SRI Tree sequence contains rich textures almost everywhere, and it is coding cost is 5.93Bpp. Note that the recovered flow field is still speckled, though the smoothness prior on the ownership weights is employed. For all the experiments shown in this thesis, we use the non-filtered original images to estimate image motions. The SRI Tree sequence contains significant amount of noise and outliers, which contribute to the notable speckling effect.

Hamburg Taxi Sequence

The final global motion experiment is performed on the Hamburg Taxi sequence. Figure 6.7 illustrates the incremental estimation process from one layer to four layers. At the final stage, the fourth layer is discarded and a three-layer description is selected. The right-most vehicle has the lowest contrast with respect to the background, therefore part of the van is assigned to the background layer. Also, one affine motion model is recovered to fit the motion of both vehicles that move left. From the horizontal velocity image, we can see that the right vehicle moves faster than the middle one, while both of them belong to the same affine layer. In the four-layer case, this layer was separated into two different layers. However, since one affine model approximates the motion of both vehicles well enough, the new layer is not accepted by the MDL criterion. The coding cost for this sequence is 2.88Bpp.

6.4.2 The "Skin and Bones" Model

To illustrate the effect of automatic selecting the number of layers using MDL criterion in the "Skin and Bones" model, we revisit the Yosemite sequence, Flower Garden sequence and SRI Tree sequence in this section.

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Skin&Bones (2 layers):	0.103	2.09°	1.83°	33.2%	62.3%	78.3%	92.7%	99.5%
Skin&Bones, with MDL:	0.102	2.08°	1.81°	34.0%	61.7%	77.6%	93.1%	99.5%

Table 6.1: Error results for the Yosemite Sequence: Skin and Bones with MDL.

Yosemite Sequence

The recovered optical flow for the Yosemite sequence example with automatic estimation of the number of layers is shown in Figure 6.8 (d)-(f). Compare this with the results of "Skin and Bones" model shown in Figure 5.5, there are few notable differences by visual inspection. Table 6.1 compares results quantitatively with and without estimating of the number of layers within each patch. For the fixed-order "Skin and Bones"model, each patch is assumed to have two affine motion layers. For the optimized-order "Skin and

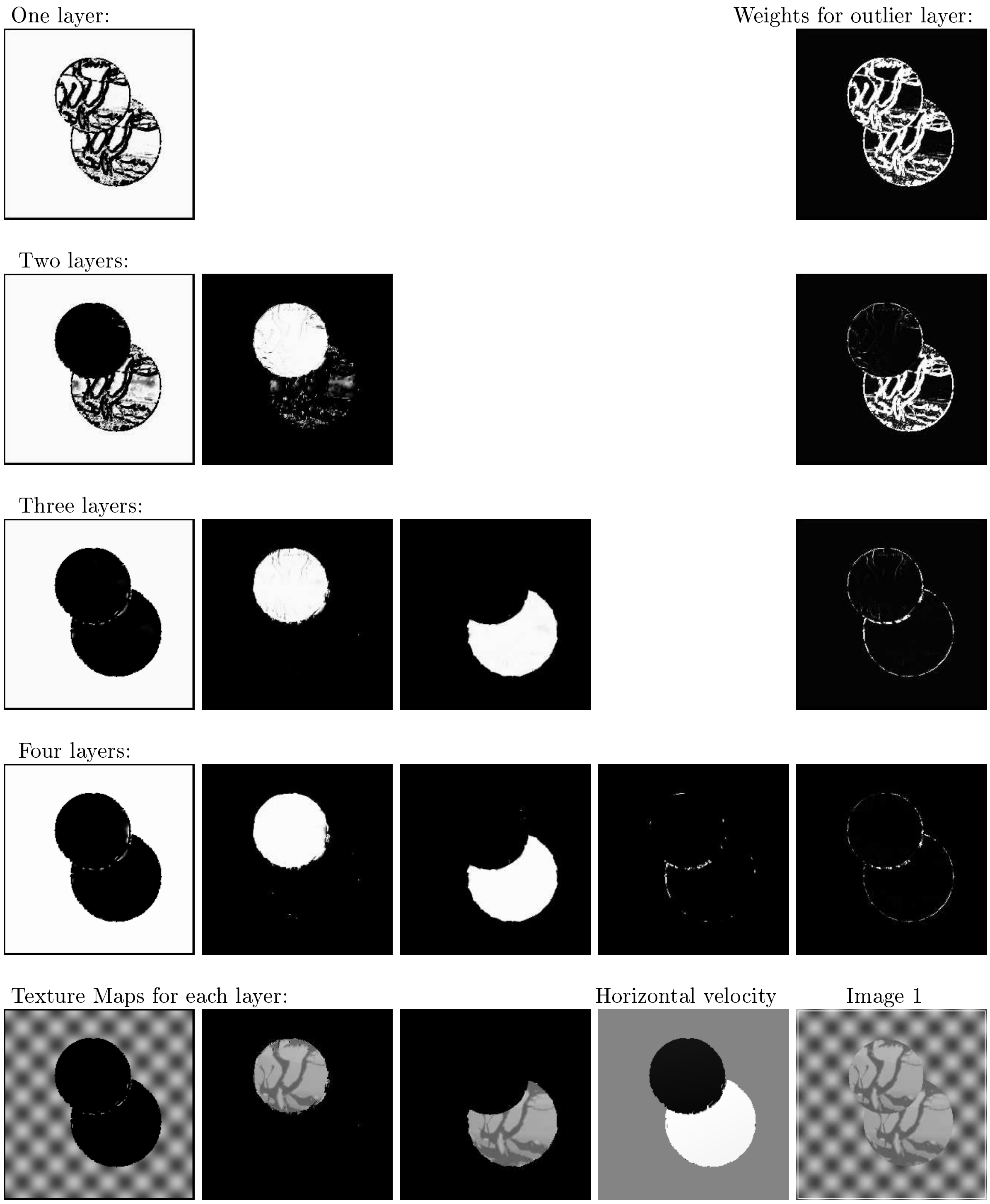


Figure 6.2: **Texture Circles Sequence:** estimate the number of layers.

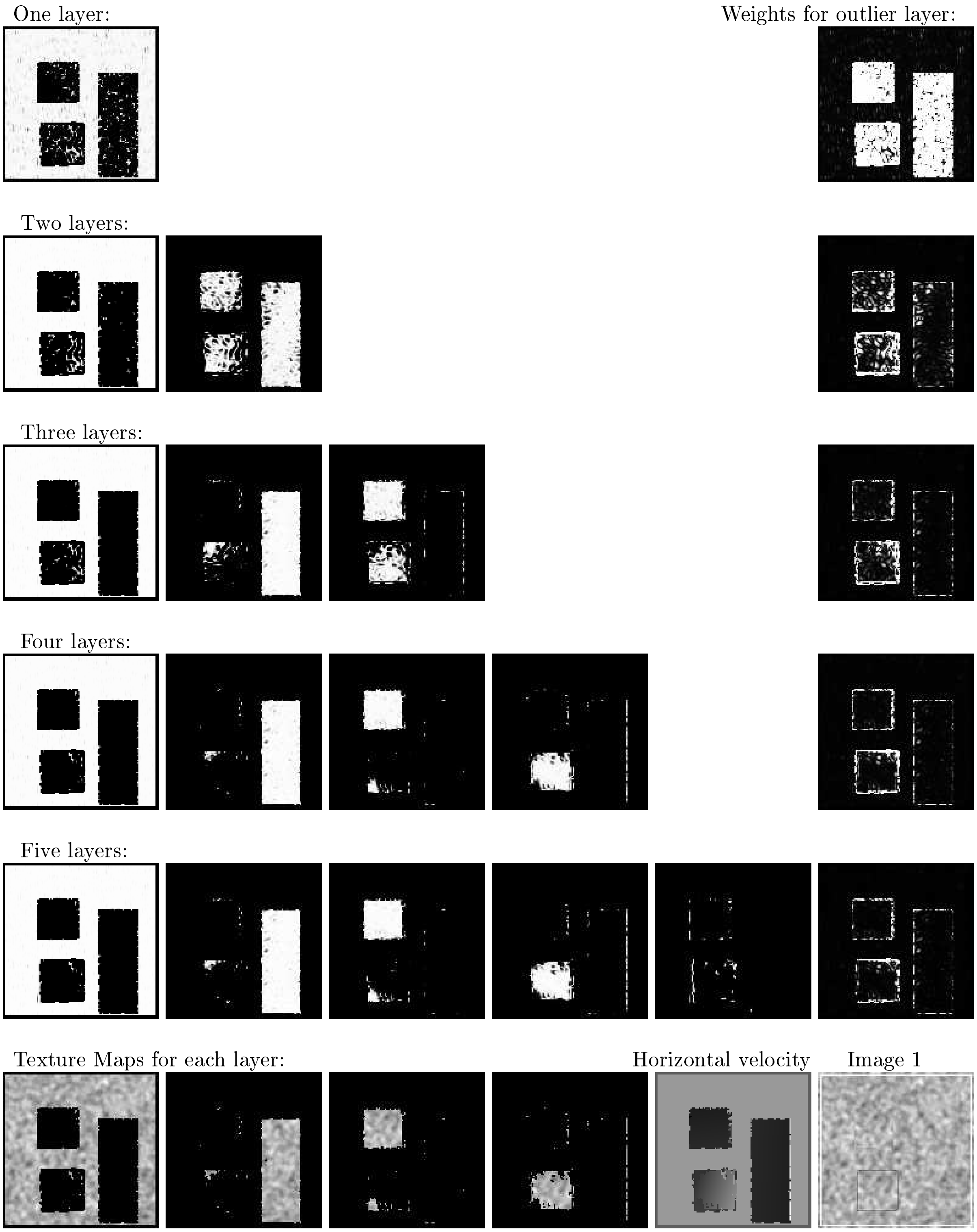


Figure 6.3: **Synthetic Bars Sequence:** estimate the number of layers.

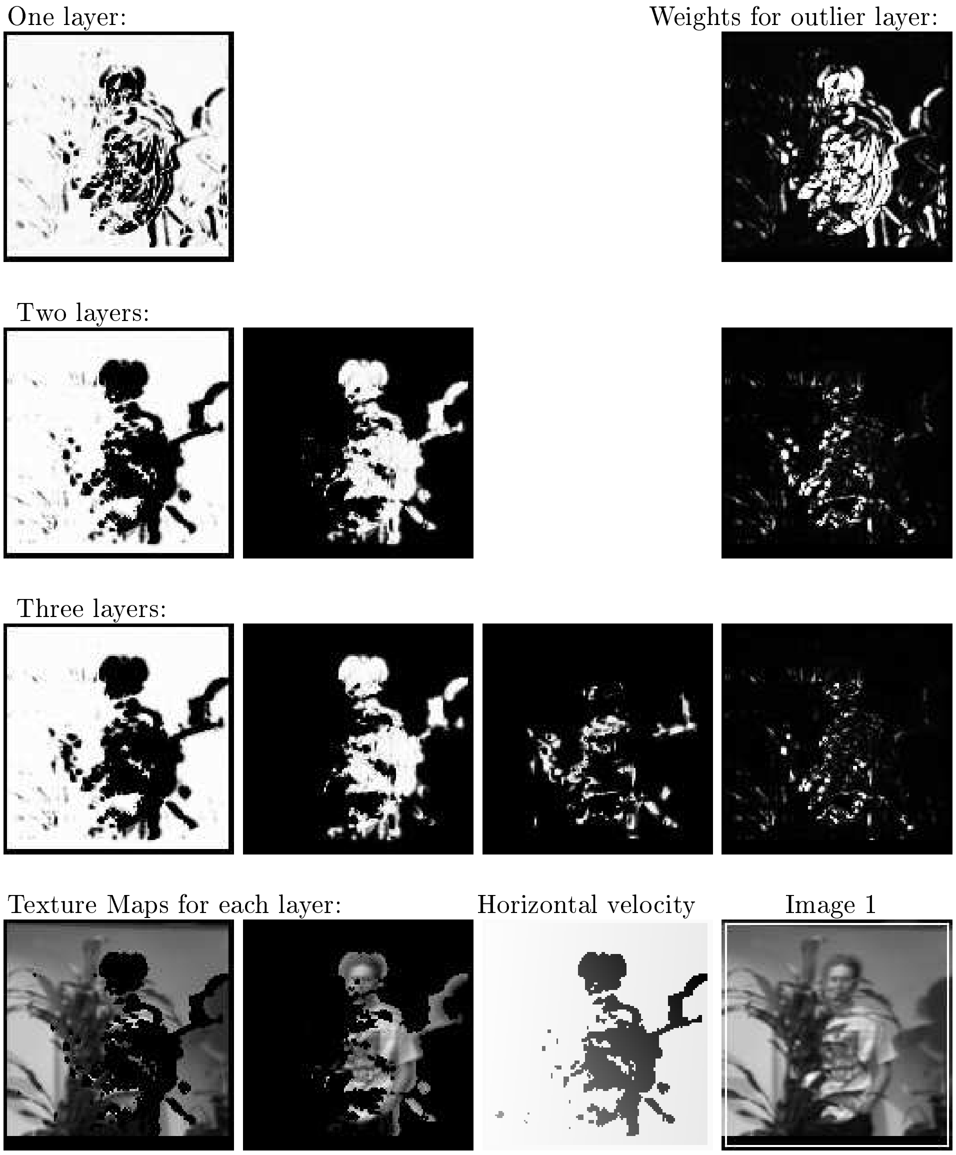


Figure 6.4: **Plant Sequence:** estimate the number of layers.

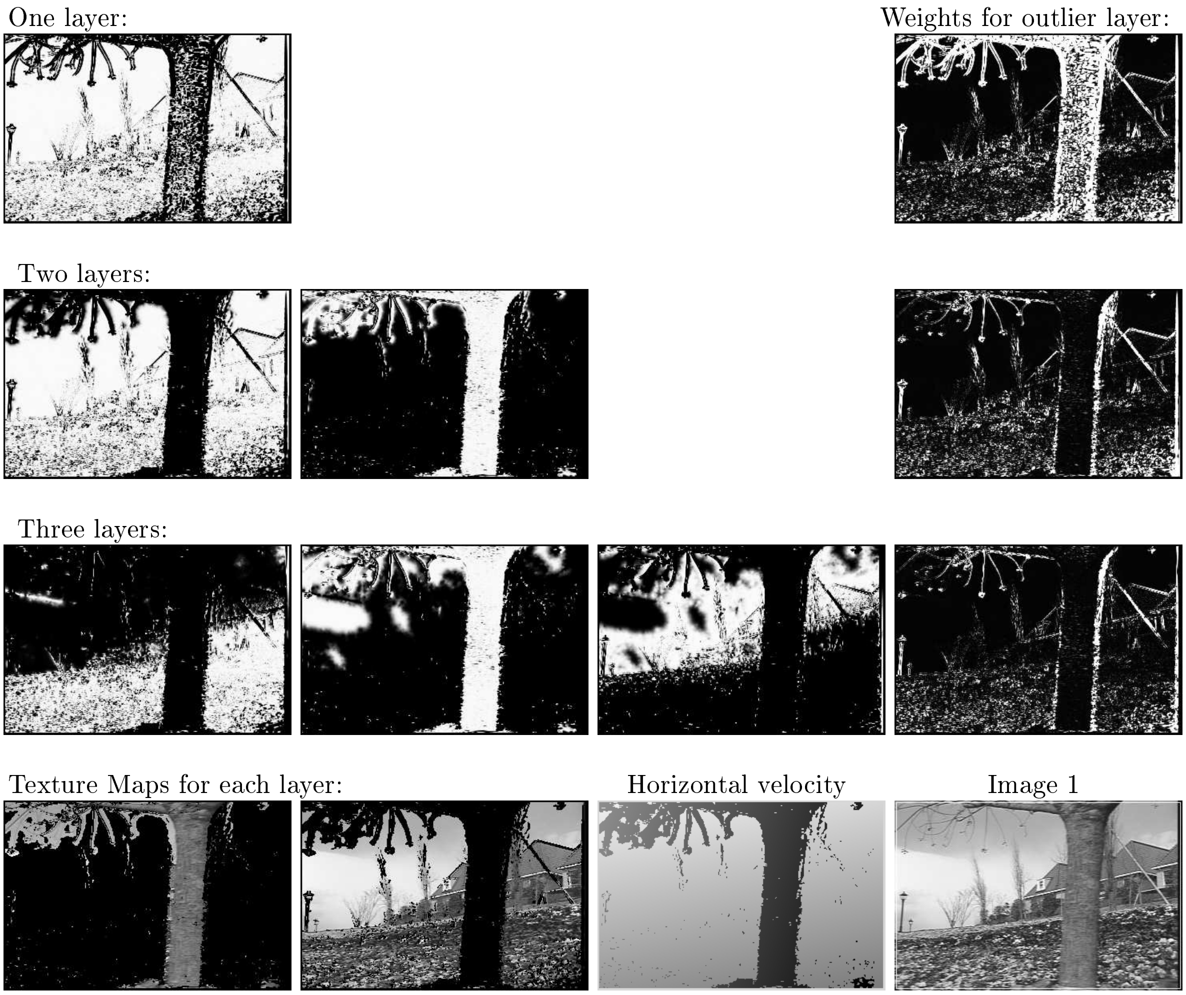


Figure 6.5: **Flower Garden Sequence:** estimate the number of layers.

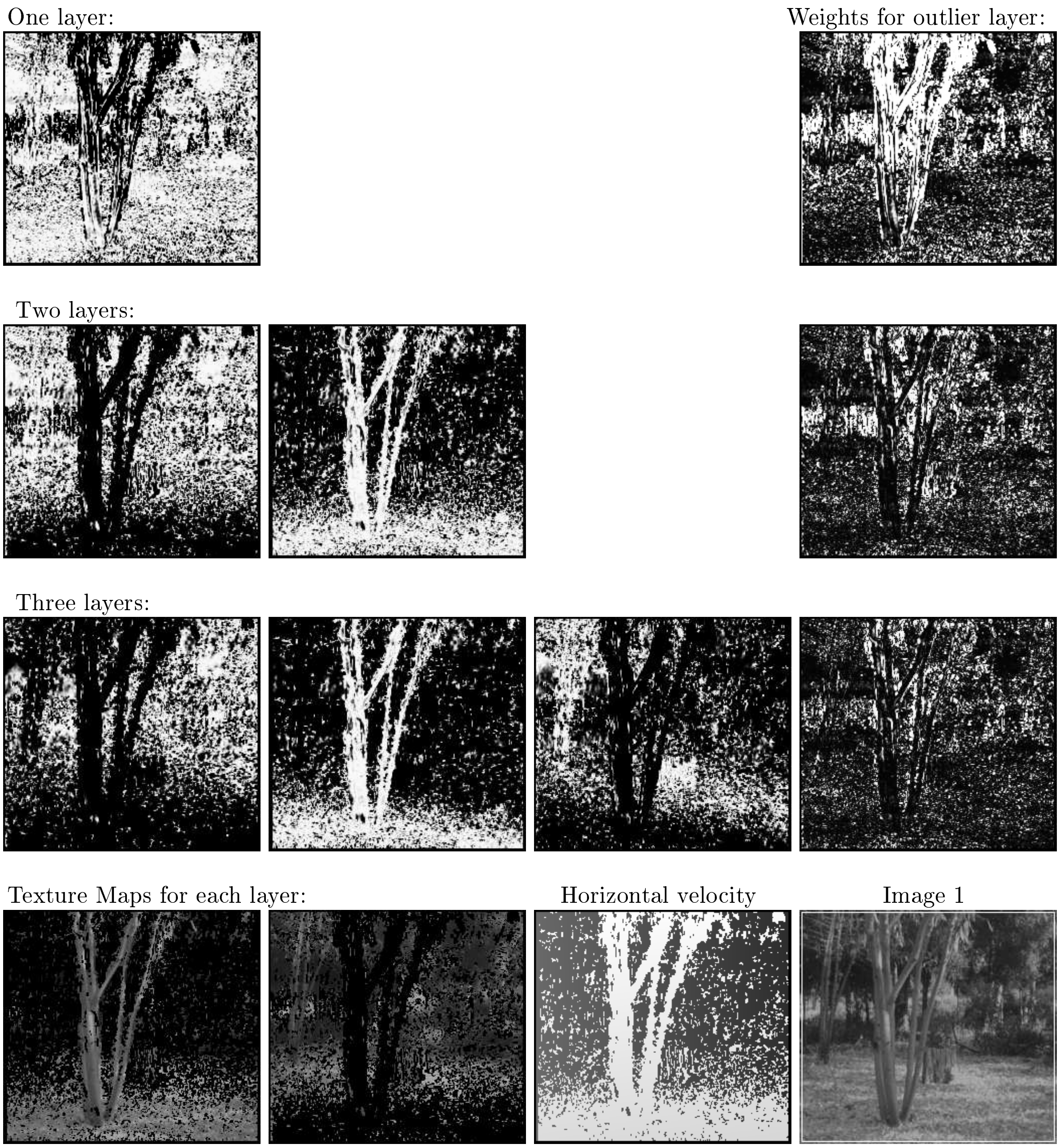


Figure 6.6: **SRItree Sequence:** estimate the number of layers.

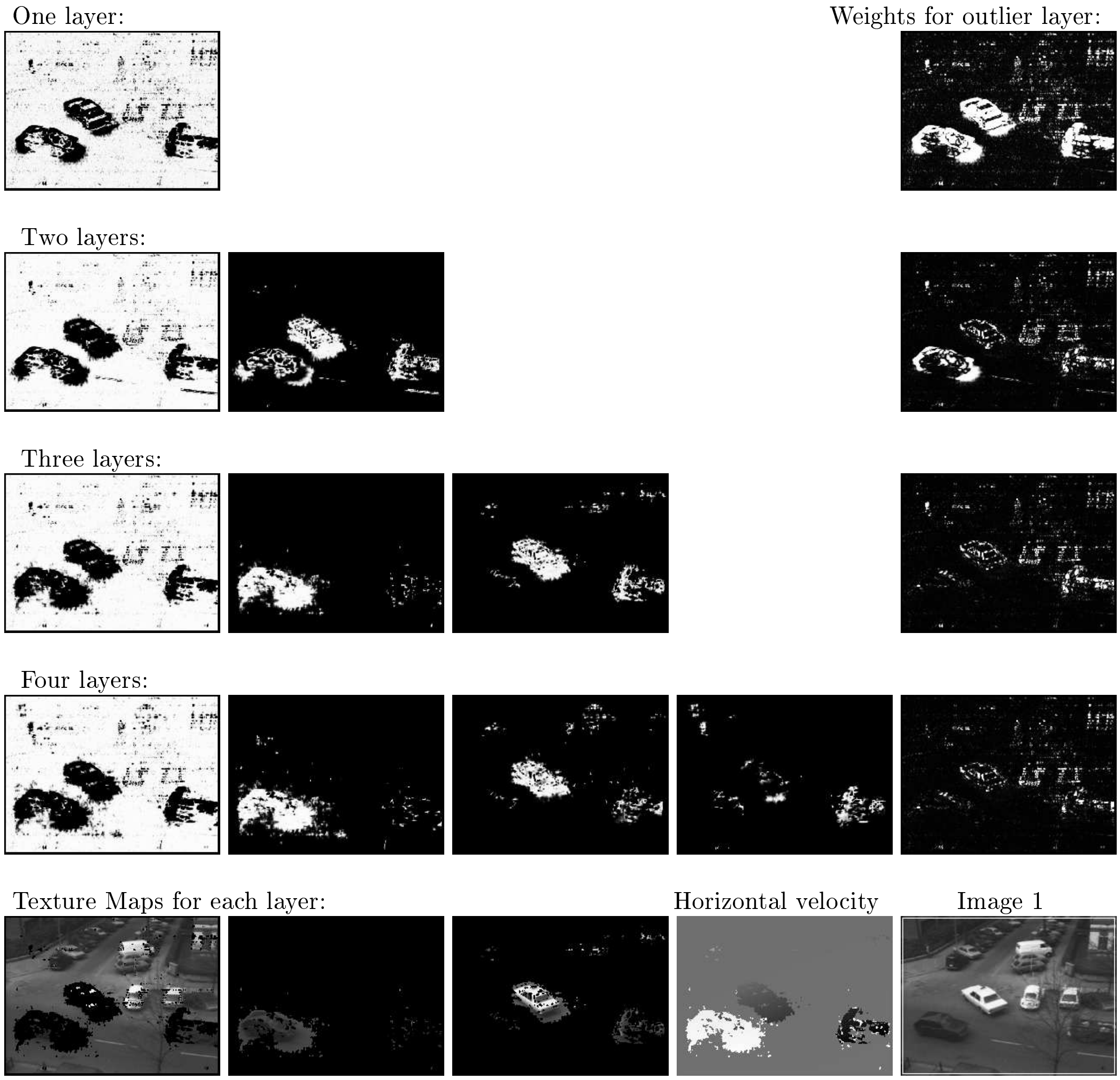


Figure 6.7: **Taxi Sequence:** estimate the number of layers.

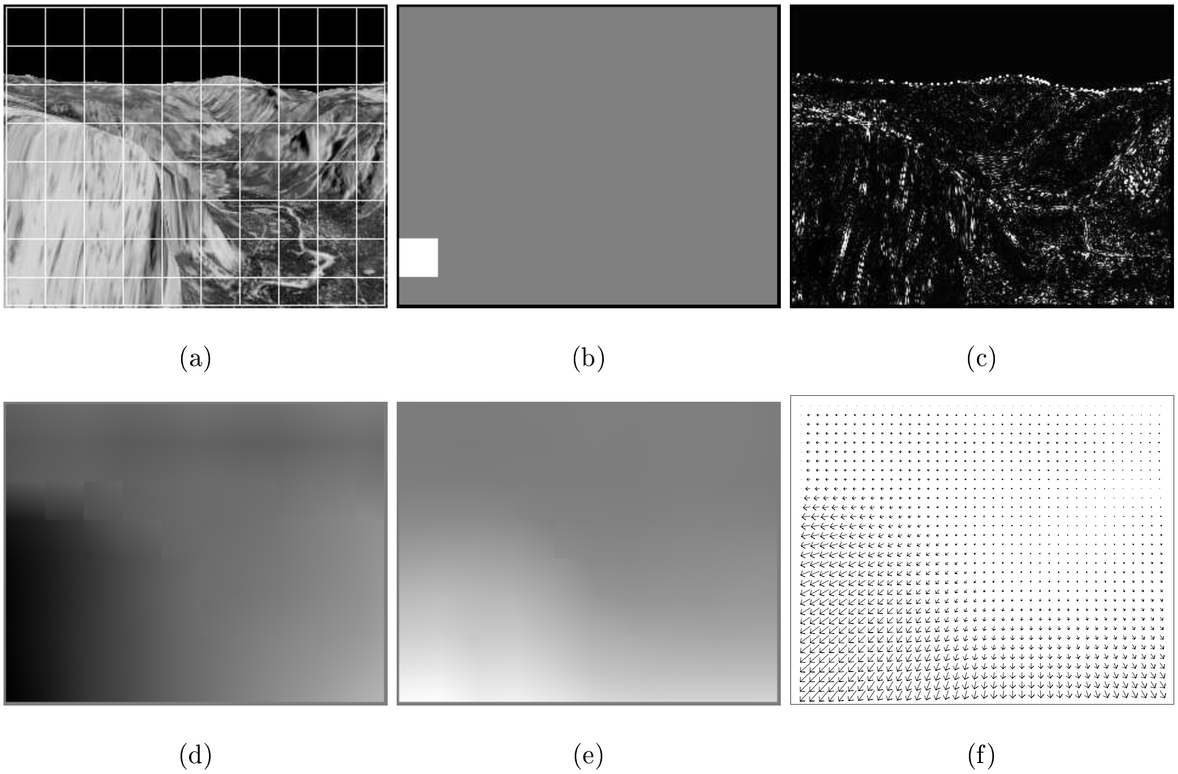


Figure 6.8: **Yosemite Garden Sequence: Skin & Bones;** (a) image one with segmented region shown; (b) number of layers estimated in each patch (gray: one layer; white: two layers); (c) weights for outlier layer; (d) horizontal component of estimated flow; (e) vertical component of estimated flow; (f) vector field of estimated flow.

Bones" model (with MDL criterion), a single layer is chosen in all patches except one (see Figure 6.8 (b)). The results of angular errors of both methods are comparable. The coding cost of the sequence with MDL criterion is 3.027Bpp, with fixed-order model (two layers in all patches) is 3.07Bpp. The main difference between the coding costs comes from the encoding of the affine parameters of the extra layer, which confirms with the fact that one affine motion is present in each image region.

Flower Garden Sequence

Figure 6.9 shows the result of the Flower Garden sequence example with the estimation of the number of layers. When only one layer is used in a patch in which multiple motions are present (e.g., at the tree boundaries), the dominant motion is recovered in some patches, while in other patches the motion is affected by leverage points (see Figure 6.9(d)). For the two layer cases, in the region that contains tree boundaries, two distinct affine motions are recovered simultaneously and smoothly connected with its neighbors (Figure 6.9(e)).

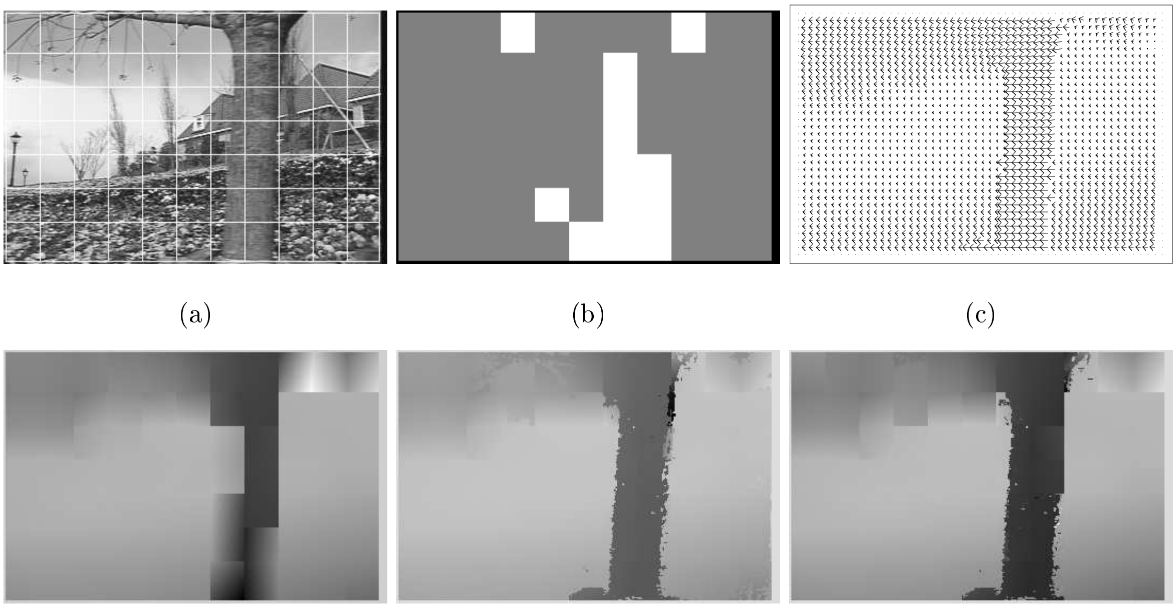
Figure 6.9(b) shows the number of layers selected in each patch using the MDL criterion. Two layers are selected in regions that border the tree, while one layer is selected in the regions of flower bed, houses and sky. Note that in regions that contain branches of the tree, one layer is chosen since the sky area has no brightness variation. Figure 6.9(f) shows the estimated optical flow with the number of layers estimated automatically. Note that the transparent regularization term is now applied as a constraint on the spatial smoothness between neighboring patches, which may contain different numbers of layers.

Consider the coding cost; the optimal code length per pixel according the MDL criterion is 3.821Bpp (while the code length for two-layer "Skin and Bones" model is 4.004Bpp). Comparing with experiment described in previous section where a global layered affine model is used, there is a significant saving in the total code length using the "Skin and Bones" model. It indicates that less bits are required to encode the residual errors and the layer assignments. It also verifies the fact that locally affine motion models are, in general, a better approximation to the image motion than a single global model.

SRI Tree Sequence

Figure 6.10 shows results of the SRI Tree sequence. Again for the one-layer model, dominant motion is recovered in most patches, except two of the patches which contain both tree branches and the background (Figure 6.10(d)). In the two-layer case, motion of the tree and the background are recovered correctly and connected smoothly between patches (see Figure 6.10(e)).

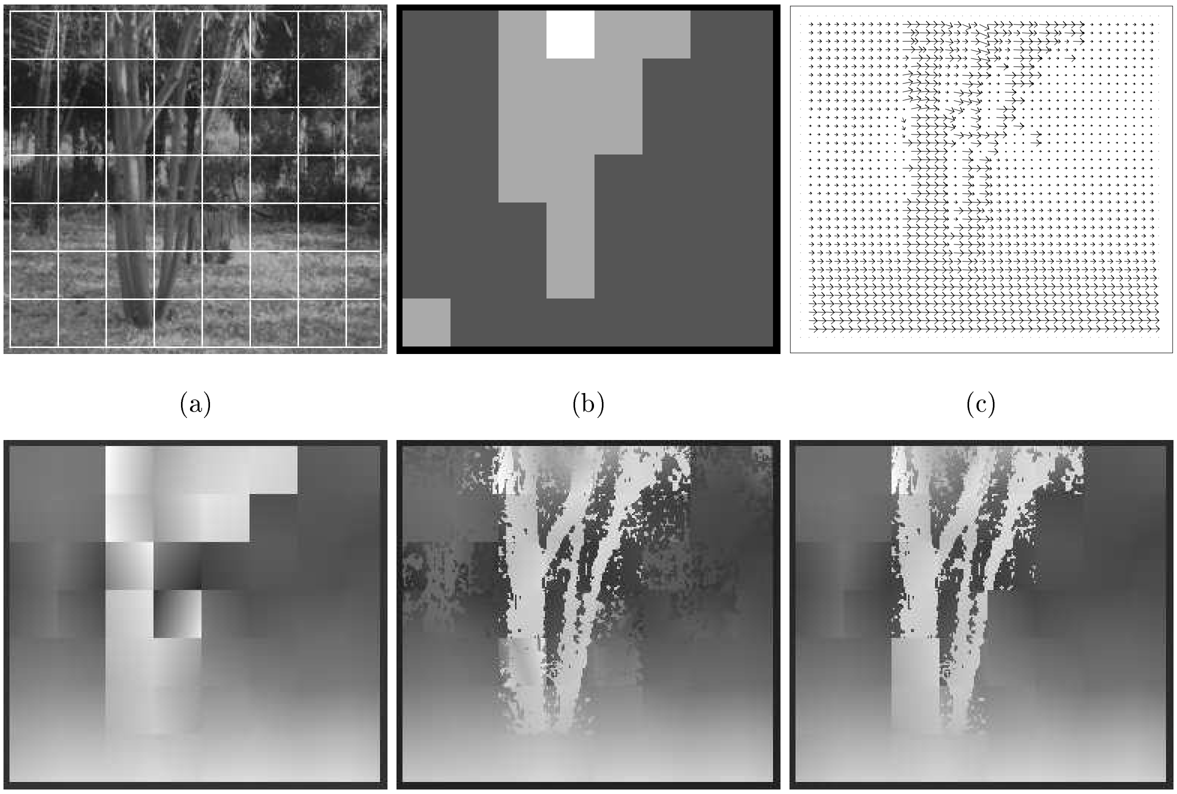
From the final horizontal motion estimated with the MDL criterion shown in Figure 6.10 (f), we see that the tree branches and the background are recovered simultaneously. Examining the number of layers selected in each patch which is shown in Figure 6.10 (b), we find that the ground plane is essentially treated as a single layer. The branches of the tree and the background, however, are assigned to different layers when they both appear in the same region. The optimal coding cost of the sequence is 5.367Bpp, which



(a) (b) (c)

(d) (e) (f)

Figure 6.9: **Flower Garden Sequence: Skin & Bones;** (a) image one with segmented region shown; (b) number of layers estimated in each patch (black: one layer; white: two layers); (c) vector field of estimated flow; (d) horizontal flow with one layer in each patch; (e) horizontal flow with two layers in each patch; (f) horizontal flow with the number of layers estimated automatically.



(a) (b) (c)

(d) (e) (f)

Figure 6.10: **SRItree Garden Sequence: Skin & Bones;** (a) image one with segmented region shown; (b) number of layers estimated in each patch (black: one layer; gray: two layers; white: three layers); (c) vector field of estimated flow; (d) horizontal flow with one layer in each patch; (e) horizontal flow with two layers in each patch; (f) horizontal flow with the number of layers estimated automatically.

saves about 0.57 bits per pixel compared with the codec performance of the global layered affine model.

Chapter 7

Estimating Image Motion Over Time

The motion estimation methods described in the previous chapters are two-frame based methods, while methods that use longer image sequences are described in this Chapter. From many frames, we may reduce the ambiguity of motion estimation and segmentation obtained from two images, and recover a layered motion representation with accuracy and high confidence. Section 7.1 reviews the *temporal coherence constraint*, and discusses one scheme that employs the constraint on image velocities or layer ownerships over time. In Section 7.2, the same constraint is applied to the layer ownerships. We propose an incremental estimation process to predict and estimate the ownership weights. We formulate the problem based on a Bayesian statistical decision theory [48], and show that the prediction and estimation equation reduce to the Kalman equations when the measurement and dynamic model are both modeled by Gaussian functions. Section 7.3 extends the formulation of multi-layer motion estimation described in Chapter 4 to add a temporal prior on the ownership weights in the mixture model. Section 7.4 presents the experimental results. Finally, Section 7.5 draws a summary of the chapter and discusses a number of issues that have not be explored yet.

7.1 Temporal Coherence Constraint

The "Skin and Bones" model can be extended over time to improve the motion estimation with a temporal coherence constraint, which states that the motion of surfaces in the world are predictable over short periods of time due to the law of physics. This section describes one of the solutions, which applies the temporal coherence constraint on image velocities over time

7.1.1 Temporal Coherence Constraint on Image Motion

The temporal coherence constraint has been exploited in a various ways to estimate spatio-temporal image motion. It has been formulated in terms of image motion at the pixel level using an incremental approach [16, 91], or scene motion at the region level using a parametric spatio-temporal approach [6, 26, 98]. Section 2.3 provides a review of these methods.

Current parametric spatio-temporal approaches assume that the motion trajectories are continuous, provided that they can be approximated by polynomial functions of time. The approaches [26, 98] that use least squares approximation can not handle images that contain multiple motions or occlusion and disocclusion. Ayer *et al.* [6] proposed to use robust error functions to estimate the dominant parametric spatio-temporal motion in the scene; pixels that belonged to the other motions were treated as outliers. The success of their approach depends on the existence of a dominant motion. With these approaches, the order of the polynomial model is either preset [98] or determined by sequential testing of goodness-of-fit [6, 26].

For parametric spatio-temporal approaches to converge to the correct model, expensive computation may be involved if there is no prior knowledge of the temporal motion model present in the scene. For example, when a camera is mounted on a moving vehicle, the camera undergoes involuntary vibration, and the motion trajectories of the acquired image sequences are portions of some "narrow-band" functions. Therefore motion trajectories can not be efficiently described by ordinary polynomial models of time. One solution is to use a trigonometric polynomial model, since the combination of a few sine and cosine curves can approximate the vibrational motion better. On the other hand, current incremental approaches [16, 91] use either a constant velocity model or a constant acceleration model to estimate image motion over time. The model is given *a priori*. These models are too simply to account for complex motion trajectories caused by vibration or elliptic motion.

All the above approaches apply the *temporal coherence constraint* directly to the image velocities. A critical issue related to the accuracy and robustness of the methods is how to determine an appropriate temporal model. The motion trajectories can be complex, particularly over a relatively long period of time. Thus how to adaptively select the temporal models and their orders, particularly for the incremental approaches, is important and still an open question.

7.1.2 A Multi-frame "Skin and Bones" Model

With the two-frame based "Skin and Bones" formulation, the use of many frames for motion estimation leads to several distinct layered estimates between consecutive pairs of images. Black and Anandan [14] proposed an approach that added a temporal coherence prior on image motion to the objective function of the two-frame formula. The constraint is applied at the pixel level. We can extend their incremental prediction and estimation method to the layer level so that it can fit into the multi-layer "Skin and Bones" model.

We treat temporal continuity as a constraint on image velocity, formulate it to be robust to allow temporal discontinuities, and incorporate it into the two-frame based "Skin and Bones" model.

Prediction

We can assume the affine motion model of one layer is constant over time. If we know the model $\mathbf{a}_i(s,t)$ of layer i for patch s at time t, we can predict the model at the next time instant to be:

$$\mathbf{a}_i^-(s,t+1) = \mathbf{a}_i(s,t), \tag{7.1}$$

where $\mathbf{a}^-$ is the predicted affine motion model.

Temporal coherence constraint with layer consistency

Given prediction of an affine motion, we can formulate the temporal coherence constraint as follows:

$$E_{T_i}(\mathbf{a}_i(s,t), \mathbf{a}_i^-(s,t)) = \lambda_T \ \rho(\mathbf{a}_i(s,t) - \mathbf{a}_i^-(s,t), \sigma_{temporal}(s)), \tag{7.2}$$

The equation constrains the current estimate $\mathbf{a}_i$ to be close to the predicted affine motion $\mathbf{a}_i^-$. ρ is a robust error function, which allows the estimate to differ from the prediction in cases where the motion model is not predicted by the temporal model. For example, when a new layer enters the patch, or an old layer leaves the patch.

Multi-frame "Skin and Bones" Model

We can add the temporal energy term into the "Skin and Bones" objective function:

$$E_i(\mathbf{a}_i(s)) = E_{D_i}(\mathbf{a}_i(s)) + E_{S_i}(\mathbf{a}_i(s)) + E_{T_i}(\mathbf{a}_i(s,t), \mathbf{a}_i^-(s,t)), \tag{7.3}$$

where $E_{D_i}(\mathbf{a}_i(s))$ and $E_{S_i}(\mathbf{a}_i(s))$ are the multi-layer data term and the regularization term defined in Equation (5.5). λ_T in Equation (7.2) controls the relative importance of the temporal term.

Temporal coherence constraint with layer transparency

Equation (7.3) implicitly assumes a layer coherence over time. Namely, the number of layers in each patch remains fixed over time, and the motion of each layer i is consistent over time. These assumptions may be violated, such as when a new layer is present. We proposed the MDL criterion to automatically select the number of layers in Chapter 6. Considering the same criterion to be used in multiple frames, the above formulation of the temporal coherence constraint is not appropriate. Since the number of layers of a patch at time t and time $t+1$ can be different. One solution is to use the regularization with transparency framework to smooth the differences of the current affine model and the affine models predicted from all the layers at the previous time:

$$E_{T_i}(\mathbf{a}_i(s,t), \mathbf{a}_i^-(s,t)) = \lambda_T \sum_{j \in \mathcal{L}(\mathrm{s},t-1)} \rho(\mathbf{a}_i(s,t) - \mathbf{a}_j^-(s,t), \sigma_{temporal}(s)), \tag{7.4}$$

where $\mathcal{L}(s,t-1)$ denotes the number of layers in patch s at time $t-1$, and $\mathbf{a}_j^-(s,t)$ denotes the prediction at time t from any layer j at time $t-1$. By using the robust error function ρ, the current affine motion estimate is smoothed with respect to the similar predictions, while dissimilar predictions will be ignored as outliers.

One could formulate a temporal constraint on the motions in the "Skin and Bones" model as described above. In this thesis, we, instead, explore the addition of a temporal constraint on layer ownerships, which is similar to the intra-patch spatial smoothness prior. The advantage of this formulation over the constraint on the motions is that it can improve the motion segmentation over time. The formulation of this method and the experimental results will be described in the remaining sections of this chapter.

7.2 Incremental Estimation

Instead of assuming the predictability of the image motion, we assume that the layer ownership at an image position is predictable over time. In this section, we propose to apply the temporal coherence constraint to the layer ownerships (i.e., the ownership weights used in the mixture models) in an incremental estimation framework.

There are two steps involved in the incremental estimation: 1) given layer ownerships at the previous frames, a predication of the ownership at the current frame is made according a temporal model; 2) given the observed layer ownership at the current frame, an estimation of the ownership is computed according the prediction and the certainty of the prediction.

Kalman filtering is the standard technique using prediction to improve state estimation over time. It is also a special case of a more general probability density propagation process. In continuous time this process can be described in terms of a dynamic model that consists of a stochastic component, which leads to a diffusion of the density function, and a deterministic component, which causes a translation of the mass of the density function. In this section, we formulate our problem based on a general Bayesian statistical decision theory.

7.2.1 Temporal Coherence Constraint on Layer Ownerships

We must estimate, 1) a set of motion models, and 2) the layer ownerships for each motion model in order to represent image motion in layers. The formulation of multi-frame and layered motion estimation should consider how the layer ownerships can be constrained over time. Since which layer a pixel belongs to remain the same in space and time in the direction of image motion, if the pixel is not occluded or disoccluded. We, therefore, assume a constant model of layer ownerships in the incremental prediction and estimation process described below.

7.2.2 The Propagation Process

We formulate the process in discrete time t for computational purposes. The state of the dynamic system at time t is denoted by z_t, and $Z_t = (z_1, z_2, \cdots, z_t)$ represents the set of all known states at time t. Similarly, the measurement at time t is w_t, and $W_t = (w_1, w_2, \cdots, w_t)$ is the set of measurements up to time t. The state in our problem stands for the estimated ownership weights of one pixel from multiple frames, while the measurement represents the observed ownership weights for the pixel using a modified two-frame method described in Section 7.3.

Stochastic dynamics

We assume that the new state is directly conditioned on its immediately preceding state only, that is:

$$p(z_{t+1}|Z_t) = p(z_{t+1}|z_t). \tag{7.5}$$

Therefore, the stochastic dynamics are entirely determined by the conditional density $p(z_{t+1}|z_t)$. We take:

$$p(z_{t+1}|z_t) = \frac{1}{\sqrt{2\pi} D_t} \exp -\frac{(z_{t+1} - z_t)^2}{2D_t^2} \quad \sim N(z_t, D_t), \tag{7.6}$$

which represents a one-dimensional random walk with D_t (diffusion variance) as the variance of the Gaussian density function. The dynamic model forms the correct state to be the same as the previous stage with the additional of Gaussian noise. For complicated problems, z may be is multi-dimensional and the density can be complex. For example, it was learned from training sequences in [54]. In our problem, since a constant temporal model of layer ownerships is assumed, the Gaussian defined in Equation (7.6) is adequate. The stochastic dynamics is used as a prior model with a time-independent diffusion variance, D_t, in the prediction step.

Measurement

Observations w_t are assumed to be mutually independent, that is:

$$p(W_t|Z_t) = \prod_{i=1}^{t} p(w_t|z_t), \tag{7.7}$$

The observation is therefore defined by the probability distribution, or likelihood function, $p(w_t|z_t)$, which represents the likelihood of an observation w_t given the state z_t. We take $p(w_t|z_t)$ to be a Gaussian as well:

$$p(w_t|z_t) = \frac{1}{\sqrt{2\pi}L_t} \exp - \frac{(w_t - z_t)^2}{2L_t^2} \qquad \sim N(z_t, L_t), \tag{7.8}$$

where L_t (likelihood variance) is the variance of the Gaussian function. In our problem, L_t is a time-dependent value. If a pixel is treated as an outlier in the two-frame based motion estimation process at time t, the observation w_t is probably unreliable and L_t should be set to a large value, such that any observation w_t is equally likely given the state z_t.

Prediction and Estimation

Given the likelihood function $p(w_t|z_t)$ for the measurement conditioned on the state, and the prior probability $p(z_{t+1}|z_t)$ for the state at $t+1$ conditioned on the state at t, we let $p(z_t|W_t)$ be the estimated probability density function of the ownership weights at time t conditioned on all the available measurements W_t. To propagate $p(z_t|W_t)$ over time, we need a prediction step:

$$p(z_{t+1}|W_t) = \int_{z_t} p(z_{t+1}|z_t)p(z_t|W_t), \tag{7.9}$$

and an estimation step:

$$p(z_{t+1}|W_{t+1}) = k_{t+1}p(w_{t+1}|z_{t+1})p(z_{t+1}|W_t), \tag{7.10}$$

where k_{t+1} is a normalization term that is independent of z_{t+1}. The estimation step uses Bayes' rule to define the *a posterior* density function. The prior $p(z_{t+1}|W_t)$ in Equation (7.10) is the prediction taken from the posterior $p(z_t|W_t)$ from the previous time and the stochastic dynamic model $p(z_{t+1}|z_t)$.

The first prediction before any measurements have been made, $p(z_1|w_0)$, is also defined by a Gaussian density with an initial state z_0 and prediction variance P_0. From Equation

(7.5)–(7.10), we can compute the density functions recursively as follows:

$$
\begin{array}{rclrcl}
p(z_1|w_0) & \sim & N(z_0, P_0), & & & \\
p(z_1|W_1) & \sim & N(\frac{w_1 P_0 + z_0 L_1}{P_0 + L_1}, E_1) & E_1 & = & \frac{P_0 L_1}{P_0 + L_1}, \\
p(z_2|W_1) & \sim & N(z_1, P_1) & P_1 & = & E_1 + D_1, \\
\vdots & & & & & \\
p(z_t|W_t) & \sim & N(\frac{w_t P_{t-1} + z_{t-1} L_t}{P_{t-1} + L_t}, E_t) & E_t & = & \frac{P_{t-1} L_t}{P_{t-1} + L_t}, \\
p(z_{t+1}|W_t) & \sim & N(z_t, P_t) & P_t & = & E_t + D_t, \\
\vdots & & & & &
\end{array}
\tag{7.11}
$$

where P_t and E_t denote prediction and estimation variances respectively. Note that this is a simple Gaussian case, where the diffusion is purely linear. Since only Gaussian densities are involved in integrating and multiplying, the density functions remain Gaussian over time. In what follows, we show that using a Kalman filter can lead to the same update equations.

7.2.3 Kalman Filter

The basic discrete Kalman filter defines a *system model*:

$$z_t = \Phi_t z_{t-1} + d_t, \qquad d_t \sim N(0, D_t), \tag{7.12}$$

where Φ_t is the transition matrix. It also defines a *measurement model* to be:

$$w_t = H_t z_t + l_t, \qquad l_t \sim N(0, L_t), \tag{7.13}$$

where H_t is the measurement matrix. In addition, the model at the initial state is given by:

$$z_0 \sim N(z_0^-, E_0), \tag{7.14}$$

where E_0 is the initial variance, z_t^- is the predicted estimate, and we use z_t to denote the current estimate in the following. The Kalman filter [61] is summarized as follows:

$$
\begin{aligned}
z_t^- &= \Phi_{t-1} z_{t-1}, \\
P_{t-1} &= \Phi_{t-1} E_{t-1} \Phi_{t-1}^T + D_{t-1}, \\
K_t &= P_{t-1} H_t^T (H_t P_{t-1} H_t^T + L_t)^{-1},
\end{aligned}
$$

$$\begin{aligned} z_t &= z_t^- + K_t(w_t - H_t z_t^-), \\ E_t &= (I - K_t H_t)P_{t-1}, \end{aligned}$$

where K_t is the Kalman filter *gain matrix*, P_t and E_t correspond to the prediction variance and estimation variance respectively. Consider the standard 1D Kalman filter equations, where the transition matrix and measurement matrix are omitted since they are identical, we can simplify the above update equations to:

$$\begin{aligned} z_t^- &= z_{t-1}, && \\ P_{t-1} &= E_{t-1} + D_{t-1}, && \\ K_t &= P_{t-1}(P_{t-1} + L_t)^{-1} &&= \tfrac{P_{t-1}}{P_{t-1}+L_t}, \\ z_t &= z_t^- + K_t(w_t - z_t^-) &&= \tfrac{w_t P_{t-1} + z_{t-1} L_t}{P_{t-1}+L_t}, \\ E_t &= (I - K_t)P_{t-1} &&= \tfrac{P_{t-1}L_t}{P_{t-1}+L_t}. \end{aligned} \tag{7.15}$$

These equations are exactly the same as those derived from the Bayesian approach. Therefore, if the measurement model and the dynamic model are both Gaussian functions, Equation (7.10) and (7.9) of the Bayesian statistical decision reduce to the standard Kalman equations [48].

7.2.4 Implementation

We apply Equations (7.11) to update the mean and variance of ownership weights over time at every pixel for all the layers *independently*. Let $w_i^-(\mathbf{x})$ denote the prediction at the current time for layer i at pixel $\mathbf{x}$; $w_i(\mathbf{x})$ denote the observation corresponding to the estimated ownership weights $w_i(\mathbf{x}, \sigma_i)$ (defined in Equation (7.24)) using the motion estimation method described in the following section; and $w_i^+(\mathbf{x})$ denote the smoothed ownership weights. In what follows, the subscript i indicates the layer index.

Given the number of layers, $\mathcal{L}$, the initial prediction $w_i^-(\mathbf{x})$ and its variance $P_i(\mathbf{x})$, and the diffusion variance $D_i(\mathbf{x})$ of layer i at pixel $\mathbf{x}$ are preset to be:

$$\begin{aligned} w_i^-(\mathbf{x}) &= 1/\mathcal{L}, \\ P_i(\mathbf{x}) &= 1.0, \\ D_i(\mathbf{x}) &= 0.1, \end{aligned} \tag{7.16}$$

Note that $w_i^+(\mathbf{x})$, which corresponds to the state in the propagation process described above, represents temporally smoothed ownership weights of layer i. Thus one can only determine which layer pixel $\mathbf{x}$ belongs to after comparing $w_j^+(\mathbf{x})$ of all the layers $j = 1, \cdots, \mathcal{L}$. We start with a prediction $w_i^-(\mathbf{x})$ that implicitly assigns pixel $\mathbf{x}$ equally to all the layers, and a large prediction variance $P_i(\mathbf{x})$ that represents a low confidence of the prediction. Using Equations (7.16) as the initial setting, the recursive process first updates $w_i^+(\mathbf{x})$ and $P_i(\mathbf{x})$ in the estimation step given the current observed weight $w_i(\mathbf{x})$, then updates $w_i^-(\mathbf{x})$ in the prediction step given $w_i^+(\mathbf{x})$, and repeats the two steps recursively.

If pixel $\mathbf{x}$ is very likely to belong to layer i, the observation, $w_i(\mathbf{x}, \sigma_i)$, obtained from a two-frame motion estimation process[1], will be close to 1.0 and $w_j(\mathbf{x}, \sigma_j)$ $(j \in \mathcal{L}, j \neq i)$ will be close to 0.0. When the estimated weights of layer i for several consecutive frames are consistently large, the smoothed weights $w_i^+(\mathbf{x})$ will converge to 1.0 with high confidence (or small variance), while $w_j^+(\mathbf{x})$ $(j \in \mathcal{L}, j \neq i)$ will converge to 0.0 with high confidence as well, but in the separated prediction/estimation process for layer j.

If pixel $\mathbf{x}$ is treated as an outlier in the motion estimation process, the likelihood variance $L_i(\mathbf{x})$ is set to 1.0, which indicates the uncertainty of the current measurement. Otherwise, $L_i(\mathbf{x})$ of layer i at pixel $\mathbf{x}$ is defined to be:

$$L_i(\mathbf{x}) = (w_i(\mathbf{x}) - w_i^-(\mathbf{x}))^2, \tag{7.17}$$

so that similar weights over time will be reinforced and the estimation/prediction variances will decline rapidly, while dissimilar weights will be smoothed and the estimation/prediction variances may decrease slower or increase depending on the magnitude of the difference.

The current state is used to predict the next state whose spatial location has moved in time. Therefore, after the prediction at all pixels is performed, the predicted weights and their variances need to be warped with respect to the affine motion model of the corresponding layer. The warping process is done independently for each layer. At an image position, only one prediction is close to 1.0 among all the predicated weights before

[1] Though we use a modified formulation described in the next section, the meaning of $w_i(\mathbf{x}, \sigma_i)$ remains the same as those defined in Equation 4.19.

warping. However, after the warping process, at occluded pixels more than one warped prediction can be near 1.0. Thus, we rescale the warped predictions at pixel $\mathbf{x}$,

$$w_i^-(\mathbf{x}) = w_i^-(\mathbf{x})/M, \qquad M = \sum_i^{\mathcal{L}} w_i^-(\mathbf{x}). \tag{7.18}$$

This rescaling is particularly useful at the occluded locations where the pixel is predicted to belong to multiple layers. In such a situation, the reweighting can reduce the certainty of the predicted weights, so that the estimation will be determined according the observation, rather than the prediction. The rescaling will not affect the predicted weights significantly at other locations, where only one layer has a high warped weight and the warped weights of all the other layers are close to 0.0. Note that the variances should not be rescaled. Since $P_i(\mathbf{x})$ of all layer i are equally small at an inlier pixel $\mathbf{x}$, therefore rescaling will reduce the confidence of the correct predictions.

7.3 Temporal Smoothness Prior

The previous section proposed an incremental approach to predict and estimate the ownership weights over time. As with the spatial smoothness prior on the ownership weights, the predicted weights can be used as a temporal smoothness prior to constrain the motion estimation process. In this section, the multi-layer motion estimation method proposed in Section 4.3 is extended to incorporate a temporal smoothness prior on the ownership weights.

7.3.1 A Posterior Probability

We need to define the conditional probability of assigning a pixel to a model given the observed motion constraint, the ownership weights of its neighbors, and the predicted ownership weights. A posterior probability $p(H_i|D_i, C_i)$ is defined to be the same as Equation (4.18),

$$p(H_i|D_i, C_i) \propto p(H_i|C_i) \ * \ p(D_i|H_i, C_i) \tag{7.19}$$

The background information (context), C_i, now contains, 1) the ownership weights $w_i(\mathbf{y}, \sigma_i)$, $\mathbf{y} \in \mathcal{N}(\mathbf{x})$, where $\mathcal{N}(\mathbf{x})$ is the set of neighboring pixels of $\mathbf{x}$, and 2) the predicted weights $w_i^-(\mathbf{x})$ with prediction variance $P_i(\mathbf{x})$. The prior probability, $p(H_i|C_i)$, is defined to be the optimal estimate of the probability that pixel $\mathbf{x}$ belongs to layer i, given the background C_i only. For simplicity, we assume the probabilities at one spatio-temporal location are independent of those at another, and Gaussian distributions are assumed for both the weights in the neighborhood and the predicted weights. The probability that pixel $\mathbf{x}$ belongs to layer i, given the background C_i is defined as follows:

$$p(H_i|C_i) = K_i(\mathbf{x})\ N_1(\hat{w}_i(\mathbf{x}) - w_i^-(\mathbf{x}), P_i(\mathbf{x}))\ N_2(\hat{w}_i(\mathbf{x}) - \mu_i(\mathbf{x}), R_i(\mathbf{x})), \tag{7.20}$$

where $N_1(\cdot)$ and $N_2(\cdot)$ are Gaussians and $K_i(\mathbf{x})$ is a normalization factor that depends on the means and variances of $N_1(\cdot)$ and $N_2(\cdot)$. The predicated weights $w_i^-(\mathbf{x})$ and variances $P_i(\mathbf{x})$ are computed in the incremental prediction step. The mean $\mu_i(\mathbf{x})$ and variance $R_i(\mathbf{x})$ are estimated given the weights of the neighbors:

$$\mu_i(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} w_i(\mathbf{y}, \sigma_i), \tag{7.21}$$

$$R_i(\mathbf{x}) = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} (w_i(\mathbf{y}, \sigma_i) - \mu_i(\mathbf{x}))^2. \tag{7.22}$$

We maximum the log likelihood of Equation (7.20), which gives rise to the optimal estimate of the prior probability:

$$\hat{w}_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x})P_i(\mathbf{x}) + w_i^-(\mathbf{x})R_i(\mathbf{x})}{P_i(\mathbf{x}) + R_i(\mathbf{x})}. \tag{7.23}$$

Given the prior probability $\hat{w}_i(\mathbf{x})$, the ownership weight $w_i(\mathbf{x}, \sigma_i)$ is determined by rescaling the posterior probabilities so that the weights sum to one. That is:

$$w_i(\mathbf{x}, \sigma_i) = \frac{\hat{w}_i(\mathbf{x}) * l_i(\mathbf{x}, \sigma_i)}{\mathcal{M}(\mathbf{x})}, \tag{7.24}$$

$$\mathcal{M}(\mathbf{x}) = \left[\sum_{i=1}^{\mathcal{L}} \hat{w}_i(\mathbf{x}) * l_i(\mathbf{x}, \sigma_i)\right] + l_{\mathcal{L}+1}, \tag{7.25}$$

where l_i is the likelihood function for layer i defined in Section 4.2. Given the ownership weights $w_i(\mathbf{x}, \sigma_i)$, layer parameters can be updated by minimizing the Equation

$$E(\mathbf{a}) = \sum_{\mathbf{x}\in\mathcal{R}} \sum_{i=1}^{\mathcal{L}} w_i(\mathbf{x}, \sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}_i) + I_t)^2 \tag{7.26}$$

```
for each $i$ in $\mathcal{L}$ layer                              ;; $\mathcal{L}$ is a fixed small number of layers
    for each pixel $\mathbf{x}$ in the patch                     ;; set the initial values
        $w_i^-(\mathbf{x}) \leftarrow 1/\mathcal{L};\ P_i(\mathbf{x}) \leftarrow 1.0;\ D_i(\mathbf{x}) \leftarrow 0.1$
    end for
    $\mathbf{a}_i^- \leftarrow \mathbf{0}$                       ;; initial estimate of an affine motion
end for
for each image in a sequence
    $(\mathbf{a}, \mathbf{w}) \leftarrow minimize(E; \mathbf{a}^-, \mathbf{w}^-, \mathbf{P})$   ;; apply the two-frame method
    for each $i$ in $\mathcal{L}$ layer
        for each pixel $\mathbf{x}$ in the patch
            if pixel is an outlier
                $L_i(\mathbf{x}) \leftarrow 1.0$                 ;; set likelihood variance for outliers
            else
                $L_i(\mathbf{x}) \leftarrow (w_i(\mathbf{x}) - w_i^-(\mathbf{x}))^2$   ;; set likelihood variance for inliers
            end if
            $w_i^+ \leftarrow \frac{w_i(\mathbf{X})*P_i(\mathbf{X})+w_i^-(\mathbf{X})*L_i(\mathbf{X})}{P_i(\mathbf{X})+L_i(\mathbf{X})}$   ;; update the prediction using Equations 7.11
            $P_i(\mathbf{x}) \leftarrow \frac{P_i(\mathbf{X})*L_i(\mathbf{X})}{P_i(\mathbf{X})+L_i(\mathbf{X})} + D_i(\mathbf{x})$   ;; update the variance using Equations 7.11
        end for
    end for
    for each $i$ in $\mathcal{L}$ layer
        $\mathbf{w}_i^- \leftarrow warp(\mathbf{a}_i, \mathbf{w}_i^+)$   ;; warp smoothed weights by affine motion
        $\mathbf{P}_i \leftarrow warp(\mathbf{a}_i, \mathbf{P}_i)$       ;; warp variances
    end for
    for each $i$ in $\mathcal{L}$ layer
        $\mathbf{w}^- \leftarrow rescale(\mathbf{w}^-)$           ;; rescale predictions
        $\mathbf{a}_i^- = \mathbf{a}_i$                           ;; set initial estimate of the affine motion
    end for
end for .
```

Figure 7.1: The incremental algorithm.

7.3.2 The Algorithm

The incremental algorithm is summarized in Figure 7.1, where the boldface letter $\mathbf{w}_i$ and $\mathbf{P}_i$ represent the set of data for layer i at all pixels. In the 8^{th} line of the algorithm, the initial estimates of all affine motions ($\mathbf{a}^-$), the predicated weights and their variances at all pixels for all the layers ($\mathbf{w}^-$ and $\mathbf{P}$) are provided as the inputs[2] of the two-frame motion estimate method, which returns $\mathbf{a}$ and $\mathbf{w}$ for all the layers as the current estimates.

7.4 Experimental Results

We extended the two-frame based method over time by adding the temporal smoothness prior described in Section 7.3 and the incremental updating process described in Section 7.2. The effect of the multi-frame method will be illustrated through three examples in this section. We focus on demonstrating the improvement of the motion segmentation when integrating information from multiple frames. In all the experiments, the method is applied to one patch covering the entire image region, and the number of layers is provided.

Pepsi Can sequence

The first experiment involves an image sequence consisting of ten 201×201 images, the first and the last image of the sequence are shown in the last row of Figure 7.2. The images contain a soda can in the foreground; the motion of which is about 1.57 pixels to the left between each frame. The can is moving in front of a textured background that also moves to the left with approximately 0.75 pixel between frames. There is no vertical motion. We estimate two affine motion layers using the incremental approach described in the previous two sections. The horizontal and vertical components of flow field, computed after the ninth frame is shown in the last row of Figure 7.2.

The results at frame 1, 2, 4, and 9, using the multi-frame method described in this chapter and two-frame based multi-layer method described in Chapter 4, are compared in the first four rows of Figure 7.2 respectively. In each row, the left two images are re-

[2]The previous image $I(t-1)$ and the current image $I(t)$ are also provided as the inputs.

sults from the two-frame based method, while the right two images are the corresponding results from the multi-frame based method. We show the ownership weights of layer one and the binary ownership map for each experiment. Note that when image motions are recovered from a pair of consecutive images, the ownership maps contain little segments that are assigned to the wrong layer in both background and foreground. By integrating motion information over time, the ownership weights are smoothed over time, and the incorrect segments in the binary ownership map disappear eventually. A good segmentation of the scene is obtained after the fourth frame. Table 7.1 shows the recovered affine motion coefficients for the ninth frame. The true motion is not available since it is a real sequence. However, we notice that a_1 and a_2 of layer one are reduced to 0.0 when integrating the motion information.

	a_0	a_1	a_2	a_3	a_4	a_5
Two-frame method, Layer 1:	-1.497	0.002	-0.001	-0.004	0.0	0.0
Multi-frame method, Layer 1:	-1.561	0.0	0.0	-0.001	0.0	0.0
Two-frame method, Layer 2:	-0.738	0.0	-0.001	0.003	0.0	0.0
Multi-frame method, Layer 2:	-0.763	0.0	-0.001	-0.003	0.0	0.0

Table 7.1: Recovered affine motion coefficients for the Pepsi Can sequence: at the 9^{th} frame.

SRI Tree sequence

We now consider the more complex SRI tree sequence containing 20 images of size 256×230 pixels. This sequence is more challenging than Pepsi sequence in that there is significant image noise and fragmented occlusion. We also estimate two affine motion layers in this experiment. The results are summarized in Figure 7.3 which shows the ownership weights of both layers, and the estimated horizontal and vertical image velocities at frame 1, 2, 5 and 10. The results of the two-frame based method on frame 10 and 11 are also shown in the last row of Figure 7.3. Overall, the performance is significantly improved over the two frame algorithm. As with the Pepsi Can sequence, most noisy segments in the binary ownership map and the estimated velocity images are corrected after about 5 frames. The images are segmented into two layers with spatially coherent support maps. Note that the recovered vertical velocity ($a_3 = 0.397$) of the tree is different from the true velocity, which is about 0 (Table 7.2). The affine motion of layer one fits both the front tree and the left smaller tree.

	a_0	a_1	a_2	a_3	a_4	a_5
Two-frame method, Layer 1:	1.679	0.009	0.001	0.024	0.0	0.002
Multi-frame method, Layer 1:	0.585	0.0	0.006	0.084	0.0	0.0
Two-frame method, Layer 2:	1.572	0.007	0.001	0.397	0.003	0.003
Multi-frame method, Layer 2:	0.602	0.0	0.007	0.030	0.0	0.0

Table 7.2: Recovered affine motion coefficients for the SRI Tree sequence: at the 10^{th} frame.

Flower Garden Sequence

The last set of results is shown on the Flower Garden sequence where three affine motion layers are estimated over time. Figure 7.4 shows the ownership weights obtained over time at frame 1, 3, and 6 in the first three rows. The horizontal and vertical components of the flow field and the binary ownership map at frame 6 are shown in the last row of Figure 7.4. We get qualitatively equivalent segmentation results in the later frames. The fourth row in Figure 7.4 shows the images of the prediction variances at frame 6. The white indicates a high variance near 0.5, while black indicates a low variance near 0.0. We can see that the variance is high near motion boundaries and the outliers.

7.5 Discussion

There are a number of issues need to be addressed regarding the incremental approach described. First, the current implementation employs only the temporal coherence constraint on the layer ownerships. It can be also applied to the affine motion models as proposed in Section 7.1.1. While the current method produces better segmentation results in our experiments, an extended method that is in progress is likely to have the advantage of applying the temporal coherence constraint on both the affine motion models and the layer ownerships.

A second issue that must be addressed is the robustness of the predication and estimation process. For simplicity we only consider Gaussian densities, which have closed form solutions for Equations (7.10) and (7.9). Gaussian distributions, however, are nonrobust and an incorrect measurement can seriously distort the estimate of the true state. Therefore, to solve the data association problem well, the measurement models (or likelihood functions) need to be robust. In [54, 109], truncated Gaussian functions were used to allow for the outliers in the measurements. Due to the lack of closed form solutions, special algorithms, such as the CONDENSATION algorithm in [54] and the network model based algorithm in [109], are used to find the approximations of the prediction distribution $p(x_{t+1}|W_t)$. There is no need for the dynamic system model to be robust, however, the model needs to be given a prior as in our formulation, or learned from the training data as in [54].

Another important issue regards how to include a mechanism to allow the number of layers to change over time, in order to account for the appearance and disappearance of objects and surfaces. A possible solution is to combine the multi-frame method with a MDL criterion similar to the one described in the previous chapter.

It is also worth mentioning that the computational cost of the multi-frame method is less than those of the two-frame based methods applied to every pair of images in the sequence. Note that in the algorithm described in Figure 7.1, the current affine motion models are used as the initial guess at the next frame. The idea is that $\mathbf{a}^{-}$ provides a good initial estimate of the affine motion model, hence, the motion estimation algorithm should converge quickly. However, due to the continuous method and the annealing step in the minimization process, a complete minimization will still be performed regardless of the initial estimates. In order to reduce the amount of computation, the annealing of scale parameter is ignored in our algorithm when a good initial estimate is provided.

Finally, building the *layered templates* over time is another issue that can be further studied. A "template" is an intensity image model which may serve as the "memory" of the appearance of one layer. "Templates" can help to solve motion phenomena involving temporal coherence. For example, the template of a surface can be modified continuously even after part of the surface is occluded, therefore the motion of an occluded object can be perceived over time. The fundamental problem is how to build the templates given a number of layered representations estimated from every two consecutive images. A widely used solution is to update the intensity layer model by combining the warped images obtained over time. Simple methods take the mean or median of the intensity values at each pixel in the registered images. Another approach based on the weighted average is proposed in [89]. The accuracy of the layered templates depends critically on the accuracy of the recovered motion models. Particularly when the image sequence is long, small errors in the estimated motion model at each frame may accumulate, which can result in notable distortion in the templates. Baker *et al.* [9] proposed to first refine layer ownerships and intensities by minimizing the differences between the re-synthesized image and the input image. The refined layer estimates are then used to adjust the model parameters. Their method for stereo layered reconstruction may be adapted to motion layered templates.

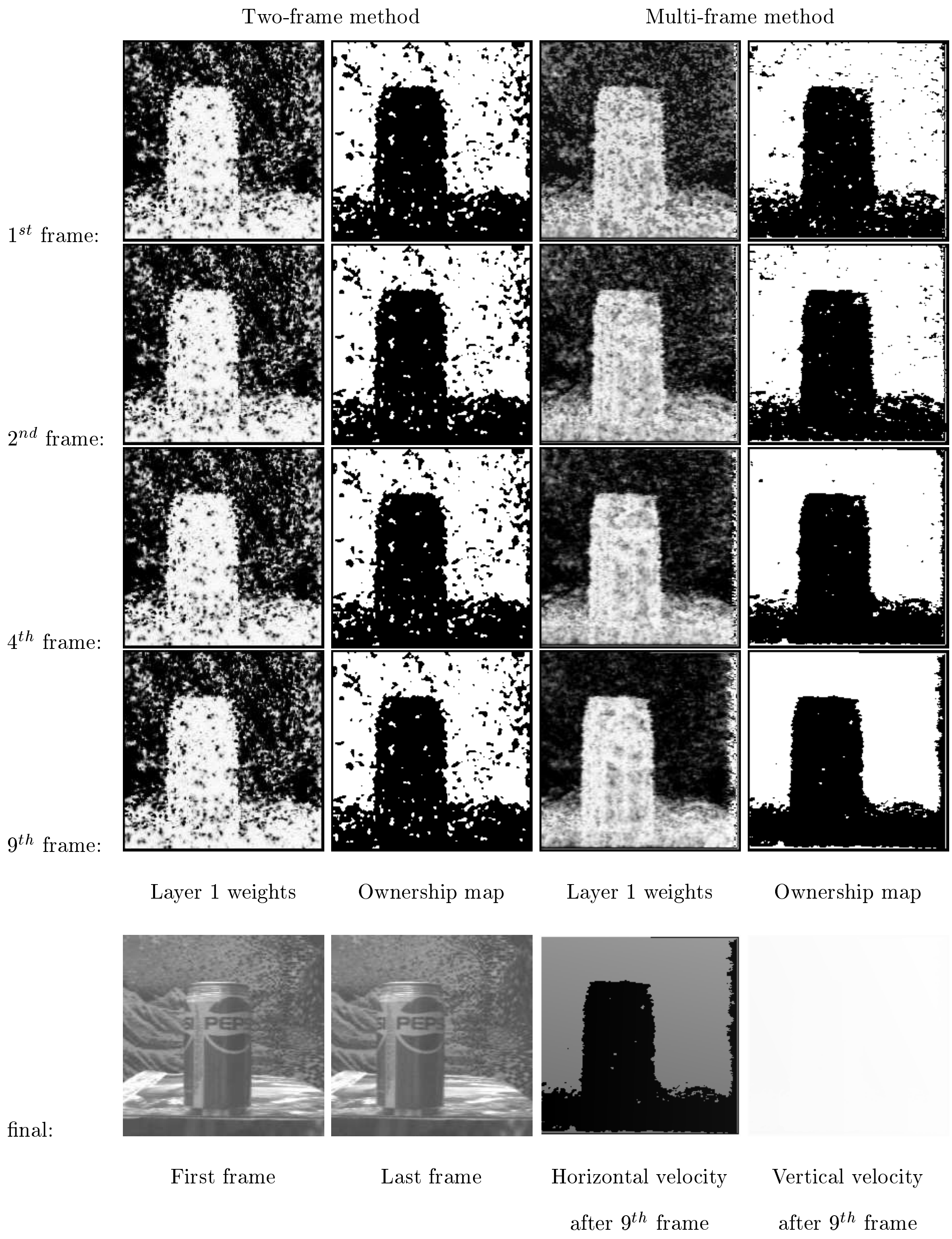


Figure 7.2: **Pepsi Can Sequence:** motion estimation over time.

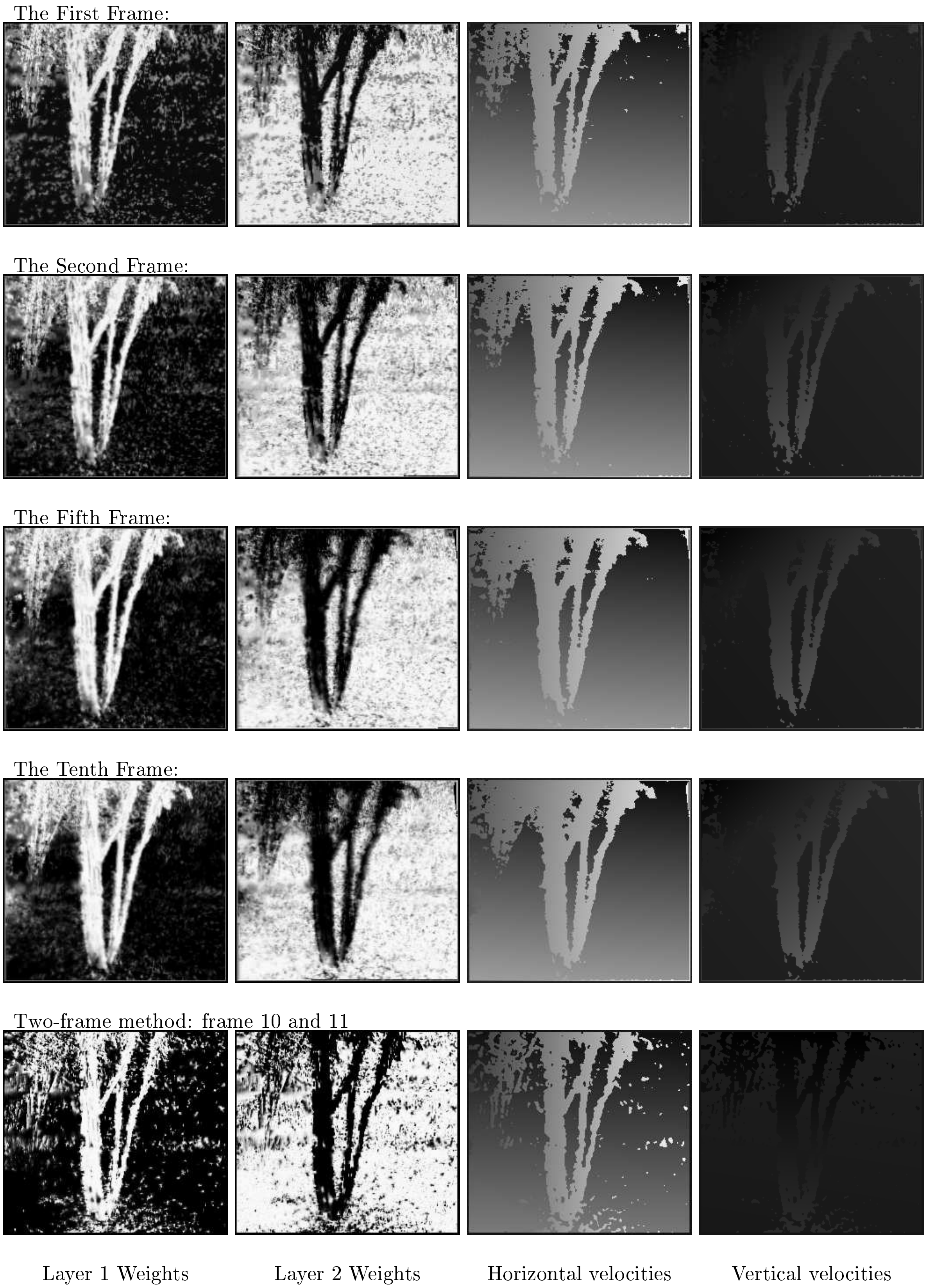


Figure 7.3: **SRI Tree Sequence:** motion estimation over time.

Figure 7.4: **Flower Garden Sequence:** motion estimation over time.

Chapter 8

Cardboard Person Model

The "Skin and Bones" model is proposed as a generic motion estimation model. This model can be specified and applied to particular applications. In this chapter, we present one application of articulated motion estimation based on a specified model of the "Skin and Bones" method. The so-called *cardboard person model* is illustrated in Figure 8.1, in which a person's limbs are represented by a set of connected planar patches. To estimate articulated human motion we approximate the limbs as planar regions and recover the motions of these planes while constraining the motion of the connected patches to be the same at the points of articulation. Experimental results are presented after the introduction of the model.

8.1 Background

The tracking and recognition of human motion is a challenging problem with diverse applications in virtual reality, sports medicine, teleoperations, animation, and human-computer interaction to name a few. The study of human motion has a long history with the use of *images* for analyzing animate motion beginning with the improvements in photography and the development of motion-pictures in the late nineteenth century. Scientists and artists such as Maret [29] and Muybridge [75] were early explorers of human and animal motion in images and image sequences. Today, commercial motion-capture systems can be used to accurately record the 3D movements of an instrumented person, but the motion analysis and motion recognition of an arbitrary person in a video sequence remains an unsolved problem. Most current approaches attempt to fit a model

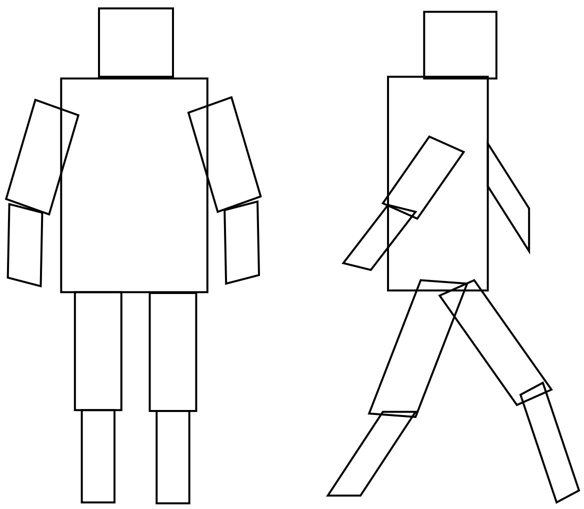


Figure 8.1: The cardboard person model. The limbs of a person are represented by planar patches.

of a person (for example a “stick-figure” or a volumetric 3D model) to image data and to recognize human motion from the changing joint angles of the model. In contrast to this model-based approach recent work by Black and Yacoob [21] focuses on the tracking and recognition of human facial expressions solely from image motion using parameterized models of optical flow.

We extend the work of Black and Yacoob [21] on tracking and recognition of human facial expressions to the problem of tracking and recognizing the articulated motion of human limbs. We make the assumption that a person can be represented by a set of connected planar patches: the *cardboard person model* illustrated in Figure 8.1. In the case of faces, Black and Yacoob [21] showed that a planar model could well approximate the motion of a human head and that it provides a concise description of the optical flow within a region. This motion can be estimated robustly and it can be used for recognition. This chapter explores the extension of this planar approximation to articulated human limb motion.

In the case of faces, the motions of the mouth, eyes, and eyebrows are represented relative to the motion of the face. To extend the approach in [21] to track articulated human motion we approximate the limbs as planar regions and recover the motions of these planes while constraining the motion of the connected patches to be the same at the points of articulation. To recognize articulated motion we will need to know the relative motion of each of the limbs. Given the computed motions of the thigh and calf, for example, we can solve for the relative motion of the calf with respect to the thigh. We posit that this relative image motion of the limbs is sufficient for recognition of human activity.

Consider, for example, a person walking towards the camera. If the sequence is stabilized with respect to the torso, the thigh regions will expand and contract cyclicly while also undergoing perspective distortions. In the thigh-stabilized sequences, the calf regions will exhibit similar optical flow patterns. Note that in this scenario, where the motion is towards the camera, stick figure models that match the stick figure to image data will likely not provide the information necessary for recognition.

The tracking of human motion using these parameterized flow models is more challenging than the previous work on facial motion tracking. In the case of human limbs, the motion between frames can be very large with respect to the size of the image region, the deformations of clothing as a person moves make tracking difficult, and the human body is frequently self-occluding and self-shadowing. In the following sections we focus on the problem of tracking the limbs of a person using articulated planar patches. At the end of the chapter we discuss how this special model can be extended.

8.2 Estimating Articulated Motion

For an articulated object, such as the Cardboard Person model, we assume that each patch is connected to only one preceding patch and one following patch, that is, the patches form a chain structure (see Figure 8.2). For example, a "thigh" patch may be connected to a preceding "torso" patch and a following "calf" patch. Each patch is represented by its four corners. Our approach is to simultaneously estimate the motions, $\mathbf{a}(s)$, of all the patches. We minimize the total energy of the following equation to estimate the motions of each patch (from 0 to n) based on Equation (3.9).

$$E = \sum_{s=0}^{n} E(s) = \sum_{s=0}^{n} \sum_{\mathbf{x} \in \mathcal{R}(s)} \rho(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}(s)) + I_t, \sigma(s)) \tag{8.1}$$

where the energy term $E(s)$ is just the data term of the single-layer bone described in Section 3.1, with the motion model $\mathbf{a}(s)$ representing a planar model defined in Equation (3.6).

8.2.1 Articulated Constraint

Equation (8.1) may be ill-conditioned due to the lack of sufficient brightness variation within the patch. The articulated nature of the patches provides an additional constraint on the solution. It is therefore useful to regularize the optical flow problem by adding a spatial coherence constraint that favors solutions which are "smooth", that is, where the spatial variation of the flow field is small. In the formulation described in Chapter 3, this constraint has been formulated to minimize the difference between optical flow vectors at the boundary of the region for *all* neighboring patches. In this section, we present a smoothness constraint on the articulated points only.

This articulation constraint is added to Equation (8.1) as follows

$$E = \sum_{s=0}^{n}\left(\frac{1}{|\mathcal{R}(s)|}E(s) + \lambda \sum_{\mathbf{x}'\in\mathcal{A}(s)} \|\mathbf{u}(\mathbf{x},\mathbf{a}(s)) - \mathbf{u}(\mathbf{x}',\mathbf{a}')\|^2\right), \tag{8.2}$$

where $|\mathcal{R}(s)|$ is the number of pixels in patch s, λ controls relative importance of the two terms, $\mathcal{A}(s)$ is the set of articulated points for patch s, $\mathbf{a}'$ is the planar motion of the patch which is connected to patch s at the articulated point $\mathbf{x}'$, and $\|\cdot\|$ stands for the norm function. There are two differences between the regularization term used here and term defined in single layer "Skin and Bones" model (Equation (3.18)). First, the set of articulated points, $\mathcal{A}(s)$, is only a subset of $\mathcal{G}(s)$, which contains all the boundary pixels. Second, a robust error function is used in the "Skin and Bones" model to allow spatial discontinuities, but here we use a quadratic function for the spatial coherence term, which indicates that no outlier is allowed.

Instead of using a constraint on the image velocity at the articulation points, we can make use of the distance between a pair of points. Assuming $\mathbf{x}'$ is the corresponding image point of the articulated point $\mathbf{x}$, and $\mathbf{x}'$ belongs to the patch connected to patch s at point $\mathbf{x}$ (see Figure 8.2), Equation (8.2) can be modified as

$$E = \sum_{s=0}^{n}\left(\frac{1}{|\mathcal{R}(s)|}E(s) + \lambda \sum_{\mathbf{x}\in\mathcal{A}(s)} \|\mathbf{x} + \mathbf{u}(\mathbf{x},\mathbf{a}(s)) - \mathbf{x}' - \mathbf{u}(\mathbf{x}',\mathbf{a}')\|^2\right) \tag{8.3}$$

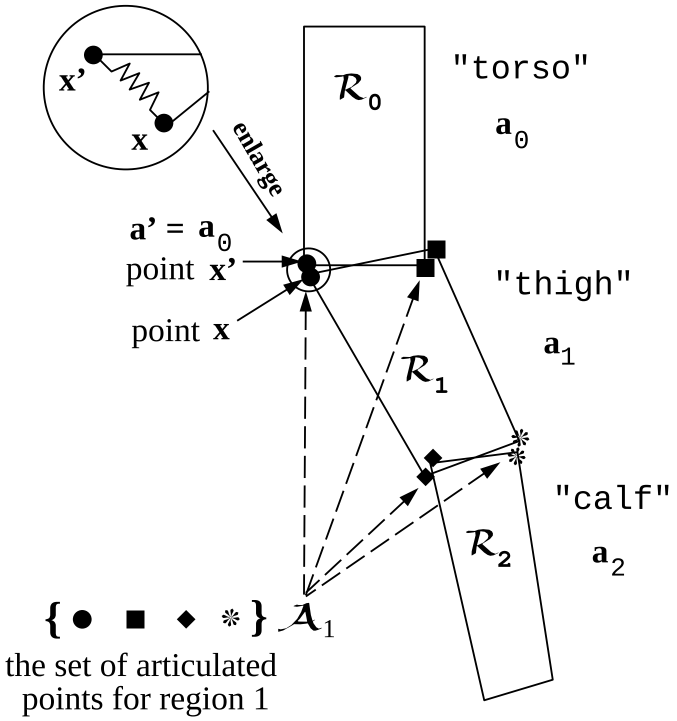


Figure 8.2: The "chain" structure of a three-segment articulated object.

This formulation has the advantage that the pair of articulated points, $\mathbf{x}$ and $\mathbf{x}'$, will always be close to each other at any time. The second energy term (the "smoothness" term) in Equation (8.3) can also be considered as a spring force energy term between two points (Figure 8.2).

We minimize Equation (8.3) using the simple gradient descent scheme with a continuation method [16, 23], which is reviewed in Section 3.1. This involves in taking derivatives of the equation with respect to each of the planar motion parameters. At each step, we take into account both the optical flow constraints within the patch and the motion parameters of the connected patches.

8.2.2 Estimation of Scale Parameters

The estimated the value of $\sigma(s)$ is defined to be the same as in Section 3.1.3, that is,

$$\sigma_{est} = 1.4826\ median_{\mathbf{x}} |\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}(s)) + I_t| \tag{8.4}$$

Equation (8.3) is minimized using continuation method described in Section 3.1, which begins with a large σ and lowers it gradually. We use a different annealing approach [60]

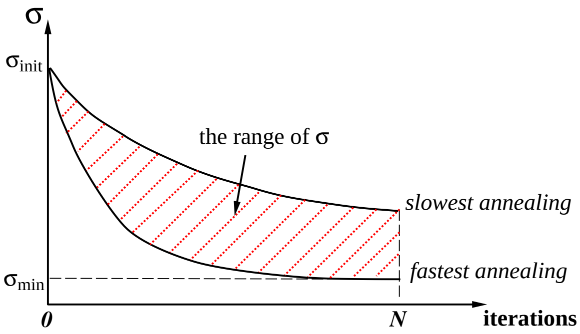


Figure 8.3: σ annealing.

described below. At each iteration we compute the current value of σ by taking into account the estimated σ_{est} in Equation (8.4) and a maximum and minimum value (σ_s and σ_f respectively).

$$\begin{aligned}\sigma &= \max(\min(\sigma_{est}, \sigma_f), \sigma_s) \\ \sigma_f &\longleftarrow \max(r_f\sigma, \sigma_{min}) \\ \sigma_s &\longleftarrow \max(r_s\sigma, \sigma_{min})\end{aligned}$$

where r_f and r_s are the fastest and the slowest annealing rates respectively. The initial scale parameter is set to a large value σ_{init} in the first iteration. The σ_{min} provides a lower bound. These parameters define a valid range of σ (Figure 8.3) and in our experiments we take $\sigma_{init} = 10\sqrt{3}$ and $\sigma_{min} = 2\sqrt{3}$. The estimated σ is bounded so that it will not decrease too fast or too slow. The annealing rate r_s is 0.97, and r_f is 0.9. These parameters are the same for all the experiments in this chapter.

8.2.3 Relative Motions

The planar motions estimated from the Equation 8.3 are absolute motions. In order to recognize articulated motion, we need to recover the motions of limbs which are relative to their preceding (parent) patches. We define

$$\mathbf{u}(\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}(s-1)), \mathbf{a}(s)^r) = \mathbf{u}(\mathbf{x}, \mathbf{a}(s)) - \mathbf{u}(\mathbf{x}, \mathbf{a}(s-1)), \tag{8.5}$$

where $\mathbf{a}(s)^r$ is the relative motion of patch s, $\mathbf{u}(\mathbf{x}, \mathbf{a}(s)) - \mathbf{u}(\mathbf{x}, \mathbf{a}(s-1))$ is the relative displacement at point $\mathbf{x}$, and $\mathbf{x} + \mathbf{u}(\mathbf{x}, \mathbf{a}(s-1))$ is the new location of point $\mathbf{x}$ under motion $\mathbf{a}(s-1)$. A planar motion has eight parameters, therefore four different points of patch s are sufficient to solve $\mathbf{a}(s)^r$ given the linear equations (8.5). In our experiments, we use the four corners of the patches.

8.2.4 Tracking the articulated object

In the first frame, we interactively define each patch by its four corners. For each patch, the first two corners are defined as the articulated points, whose corresponding points are the last two corners of its preceding patch. The corresponding points of the last two corners of this patch are the first two corners of its following patch (See Figure 8.2). This definition of articulated points shows that two connected patches share one common "edge". Once the "chain" structure is defined, the object is automatically tracked thereafter. Tracking is achieved by using the articulated motion between two frames to predict the location of each patch in the next frame. Each part of the articulated object is a quadrilateral. Since a line on a plane remains a line under the planar motion $\mathbf{a}$, these patches remain quadrilaterals. We update the location of each of the four corners of each patch by applying its estimated planar motion to it.

8.3 Experimental Results

In this section we illustrate the performance of the tracking algorithm on several image sequences of lower body human movement. We focus on "walking" (on a treadmill, for simplicity) and provide the recovered motion parameters for two leg parts during this cyclic activity. Notice that during "walking" the upper body plays only a minor role in recognition (it can, however, be appreciated that the movement of the torso and the arms can be used in determining heading, speed of "walking" and clues regarding the positions of lower body parts). To facilitate the use of our gradient-based flow estimation approach, we use a 99Hz video-camera to capture a few cycles of "walking" (lower frame rates would make it necessary to employ a correlation based approach to overcome the large inter-frame displacements of body parts). We assume that body parts were initially located and delineated by a polygon in the first frame.

We took several sequences from different viewports. Each sequence contains 500 to

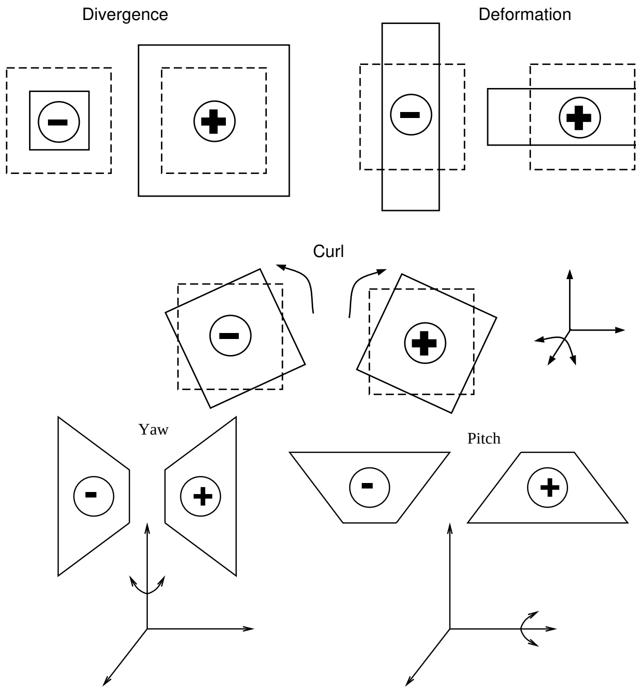


Figure 8.4: **Interpretations of the motion parameters:** divergence $(a_1 + a_5)$, deformation $(a_1 - a_5)$, curl $(-a_2 + a_4)$, image yaw (a_6) and image pitch (a_7). Note that the divergence, deformation and curl are approximations.

800 frames. All the parameters used in the motion estimation algorithm were exactly the same in all the experiments. In particular, for each pair of images, 30 iterations of gradient descent were used at each level, and 3 levels were used in the coarse-to-fine strategy. The parameters to control σ annealing were described in the previous section. The value of λ is 0.005. In the following results no temporal smoothing of the motion parameters was done.

Figures 8.5, 8.7, and 8.9 demonstrate three "walking" sequences taken from different view-points. (walking parallel to the image plane, near 45 degrees, and away from the image plane, respectively). The left column in each figure shows three input images some frames apart, the right column shows the tracking of two parts (the "thigh" and "calf").

The coefficients of a planar transformation (Equation (3.6)) can be used to interpret the motion within each region. Various, low-level, interpretations of the motion coefficients are shown in Figure 8.4. To illustrate the recovered planar motion models, we show various motion parameters for these sequences in Figures 8.6, 8.8, and 8.10. The first row in Figures 8.6 and 8.8 shows the horizontal and vertical translation (left most graph, dashed line is the vertical translation) and “curl” (right graph) for the “thigh”. The second row shows the graphs for the “calf.” In Figure 8.10 the “curl” graphs are replaced by the “deformation” and “divergence” and “image pitch”. These graphs are only meant to provide an idea about the effectiveness of our tracking model and its ability to capture meaningful parameters of the body movement.

In Figures 8.6 and 8.8 it is clear that the horizontal translation and “curl” parameters capture quite well the cyclic motion of the two parts of the leg. The translation of the “calf” is relative to that of the “thigh” and therefore it is significantly smaller. On the other hand, the rotation (i.e., “curl”) is more significant at the “calf”. Notice that Figures 8.6 and 8.8 are qualitatively quite similar despite the difference in viewpoint. Notice that as the “curl” changes signs (for example about frame 300), there is considerable translation in the “calf” since the “thigh” is almost In both cases the motions measured at the “calf” are slightly more pronounced than the motions measured at the “thighs.” Notice that the vertical translation is minimal. In Figure 8.10 the translations are smaller than before but still disclose a cyclic pattern. The “deformation,” “divergence,” and pitch capture the cyclic motion of the “walking away” on the treadmill. Notice that the pitch measured at the two parts is always reversed since when the “thigh” rotates in one direction the “calf” is bound to be viewed to be rotating in a opposite way.

In summary, the reported experiments show that the image motion models are capable of tracking articulated motion quite accurately over long sequences and recovering a meaningful set of parameters that can feed into a recognition system. For related work see [32].

8.4 Discussion

The approach described in this chapter uses a specific approximation of the single-layer “Skin and Bones” model, and extends previous work on facial motion [21] to articulated motion. It shows promise for tracking and recognition of human activities. There are, however, a number of issues that still need to be addressed. First, the motion of human limbs in NTSC video (30 frames/sec) can be very large. For example, human limbs often move distances greater than their width between frames. This causes problems for a hierarchical gradient-based motion scheme such as the one presented here. To cope with large motions of small regions we will need to develop better methods for long-range motion estimation.

Unlike the human face, people wear clothing over their limbs which deforms as they move. The "motion" of the deforming clothing between frames is often significant and, where there is little texture on the clothing, may actually be the dominant motion within a region. A purely flow-based tracker such as the one here has no "memory" of what is being tracked. So if it is deceived by the motion of the clothing in some frame there is a risk that tracking will be lost. One possible solution is to add a template-style form of memory to improve the robustness of the tracking.

Self occlusion is another problem typically not present with facial motion tracking. Currently we have not addressed this issue, preferring to first explore the efficacy of the parameterized tracking and recognition scheme in the non-occlusion case. In extending this work to cope with occlusion, the template-style methods mentioned above may be applicable.

To conclude, we have presented a method for tracking articulated motion in an image sequence using parameterized models of optical flow in this chapter. Unlike previous work on recovering human motion, this method assumes that the activity can be described by a the motion of a set of planar patches with constraints between the patches to enforce articulated motion. No 3D model of the person is required, features such as edges are not used, and the optical flow is estimated directly using the parameterized model. An advantage of the 2D parameterized flow models is that recovered flow parameters can be interpreted and used for recognition as described in [21]. Prior knowledge of the segmentation and the structure of articulation is required to initialize the cardboard people model. We manually initialize the model in the first frame, which should be automated in future work.

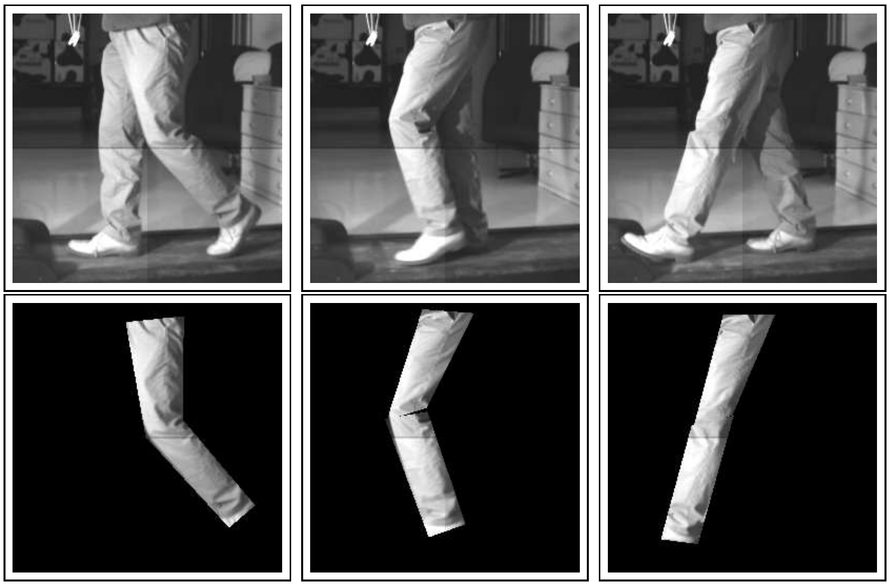


Figure 8.5: Walking parallel to the imaging plane. Three frames shown twenty frames apart.

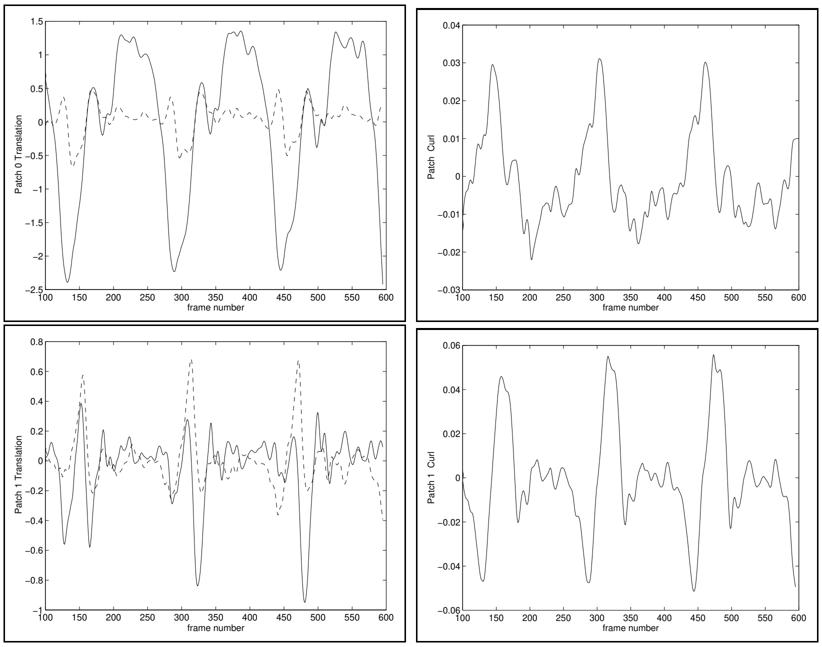


Figure 8.6: Motion parameters for walking parallel to the imaging plane (Figure 8.5). The sequence contains 500 frames, approximately three cycles.

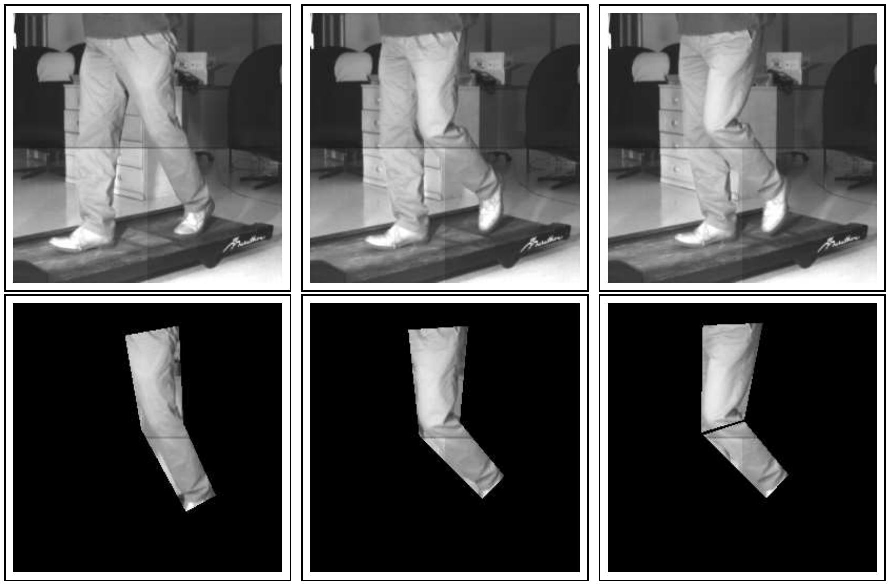


Figure 8.7: Walking 45 degrees relative to the imaging plane.

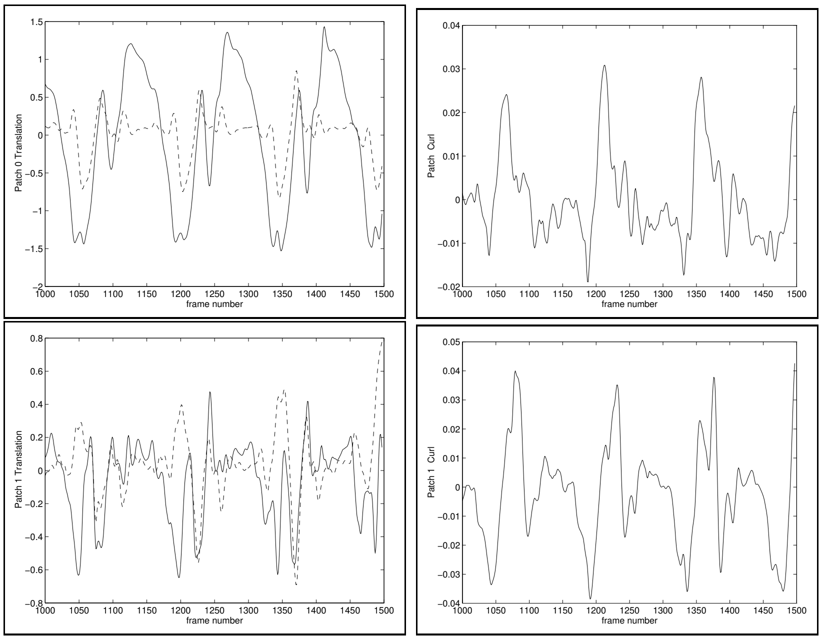


Figure 8.8: Motion parameters for walking 45 degrees relative to the imaging plane (Figure 8.7). The sequence contains 500 frames, approximately three cycles.

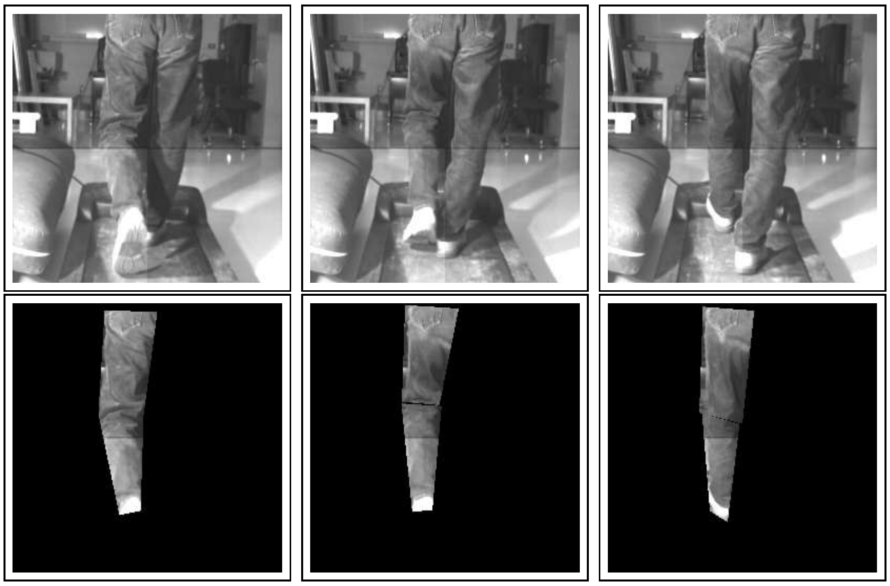
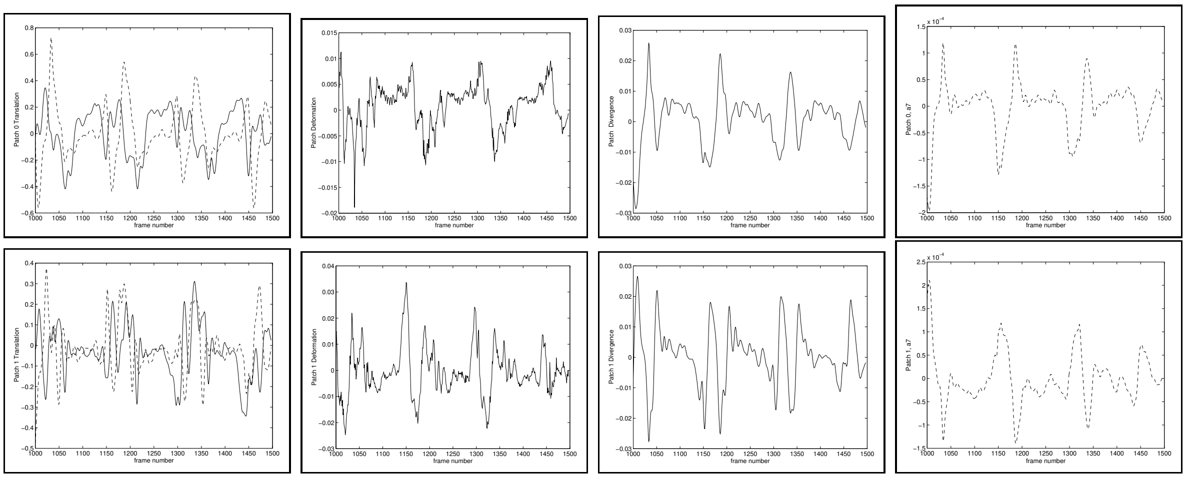


Figure 8.9: Walking perpendicular to the imaging plane.

Thigh:

Calf:



Translation Deformation Divergence Image Pitch

Figure 8.10: Tracking results for Figure 8.9. The sequence contains 500 frames, approximately three cycles.

Chapter 9

Conclusions

Estimating optical flow accurately involves pooling information over a large area. Parametric motion models do this well and can cope with multiple motions in certain cases but are not applicable globally. When applied locally, however, insufficient constraints may result in an unstable solution. We have shown how regularization can be extended to constrain these local affine flow parameters in a novel motion estimation method proposed in this thesis. This chapter concludes the dissertation with a summary of the main contributions of this work in Section 9.1. In Section 9.2, we summarize the open questions and possible directions for future work.

9.1 Contributions

There have been two main themes pursued through out this thesis. The first is how to accurately estimate multiple motions in a local region, and the second is how to constrain local parameterized motion estimates. We propose the "Skin and Bones" model to estimate image motion in layers. The following part of this section summarizes the contributions of this work.

The "Skin and Bones" Model

In Chapter 3, we developed the simplified "Skin and Bones" model, which assumes that only a single motion is present in a patch. To motivate the need for estimating image motion locally, we have shown in this chapter that a global parametric motion model is often not sufficient to model image motion in the entire image. Comparisons of estimated image motion using one global affine model and local affine models have shown that local models outperform one global model. To motivate the need for regularizing the local affine models, we have shown that the motion estimation problem in local patches may be under-constrained due to the lack of brightness variation. The "Skin and Bones" model, which combines the features of both the regularized approaches and the parameterized approaches, has been proposed to meet these two needs. The model has the accuracy of the parameterized methods and the generality and flexibility of the regularized methods. The following issues were also addressed in Chapter 3:

1. We introduced an approach to determine the scale parameters at each iteration, by integrating the estimated scale parameter with an annealing scheme.

2. The method's accuracy and robustness were demonstrated through a series experiments involving several synthetic image sequences and sequences with additive noise. Our experimental results were also compared with other published results, and the comparison indicated that the "Skin and Bones" method produced results that fall into the most accurate category.

3. We also illustrated the effect of tiling the images with different sized patches by rigorous comparison of the estimated flow fields.

The single-layer model has a limitation in that it cannot handle multiple motions within a patch. Thus it is generalized to allow simultaneous estimation of multiple affine motions in Chapter 4. We have presented an approach for computing a layered description based on mixture models and the EM algorithm. The image motion at a pixel is assumed to be modeled by one of $\mathcal{L}$ affine layers or an outlier layer. $\mathcal{L}$ is a preset number. In contrast to traditional mixture of Gaussian distributions, a mixture of robust likelihood functions is used, which fall off more sharply than those of normal distributions. When applied globally, the algorithm has shown promising results on some sequences, but was not very successful on the other sequences. We believe that the multi-layer method should also be applied to local patches, and employed it *independently* in each 32×32 patch. However, locally affine estimates are possibly incorrect in regions that contain single oriented motion constraints or little texture. In Chapter 5, we developed the complete multi-layer "Skin and Bones" model, which used the regularization term to improve local affine estimates. Furthermore, the following issues were also presented in Chapter 4:

1. We introduced a spatial smoothness prior on the ownership weights, which will be summarized in detail below.

2. We demonstrated a series of experiments on the same sequences, each of which simultaneously estimate image motion in the entire image region given a fixed number of layers. Our experimental results showed that approaches based on mixture models could cope with a small number of motions within a region. Estimating a large number of layers did not make notable improvements. This also motivates the need to apply the multi-layer method locally.

3. We illustrated the problem caused by tiling the images with non-overlapping patches, namely, the layer that has little support may not be recovered. This is most likely to occur when a layer occupies a small boundary region that contains very little texture. We proposed to use overlapped patches and showed how it could overcome this problem partly, but not completely. This motivates the need to regularize the local motion estimates.

Regularization with Transparency Framework

Considering the regularization problem when there are multiple measurements at a given point, we have proposed a general framework for regularization with transparency that extends regularization to cope with multiple measurements in Chapter 5. The framework is used to regularize multiple local motion estimates. The experiments demonstrated the robustness and accuracy of the "Skin and Bones" model through flow fields produced from both synthetic and real image sequences. By comparing with the experimental results shown in Chapter 4, we observed that the regularization term could result in a more stable optimization problem and more accurate motion estimates.

Estimating the Number of Layers

In Chapter 6, we have explored and presented a solution to the problem of estimating the number of layers within a patch. We formulated the problem using the Minimum Description Length principle. To solve for all unknown parameters, we proposed a practical method that contained two stages. The first stage computes, given the number of models, the ML estimates of the mixture model parameters and the ownership weights using the "Skin and Bones" method. These solutions in turn are used to incrementally test for the most appropriate number of models by computing the encoding cost of the model parameters, model structures, and residuals in the second stage.

Spatial and Temporal Smoothness Prior

The mixture model of affine motions (Section 4.2) assumed that each layer is equally likely, that is, the mixture proportions are equal for all layers. In other word, this assumes a type of independence in the ownership weights, so that knowing the membership of a particular location yields no information on the membership of all other locations in the image. With this formulation, the estimated optical flow fields were noticed to be "speckled" in some sequences. Moreover, "leverage points" can have strong influence on the estimated motion that pull the solution away from the desired local motion.

Like Weiss and Adelson [105], we added a spatial coherence constraint to the ownership weights, namely nearby pixels are likely to belong to the same model. This constraint has two advantages. First, it is likely to reduce the effect of leverage points by encouraging layers to have spatially coherent support. Second, it is likely to assign ambiguous regions, where the motion constraints can be equally well assigned to any layer, to the layer of its neighbors. We used a prior in the mixture models that enabled the pixel to prefer a layer over others. The prior was determined given the ownership weights of the neighboring pixels. This formulation fits naturally into the EM framework, and it does not require extra and extensive computations. Furthermore, it depends on the static intensity information implicitly through the likelihood function, hence we avoid using any ad hoc function.

In Chapter 7, a similar temporal smoothness prior has been applied to constrain the motion estimation problem at a current frame, given the predicted ownership weight from previous frames.

Applications and Specified Models

The generality of the “Skin and Bones” model was illustrated by a special application based on a specialization of the generic “Skin and Bones” model. In Chapter 8, we proposed the “Cardboard People” model, in which a person’s limbs are represented by a set of connected planar patches. To estimate articulated human motion we approximate the limbs as planar regions and recover the motions of these planes using the robust motion estimation method, while constraining the motion of the connected patches to be the same at the points of articulation using a regularization term.

9.2 Open Questions and Future Directions

The work presented in this thesis provides a novel framework for estimating accurate optical flow common to parameterized schemes, while maintaining the flexibility of regularization schemes. The results of the method are compelling, however, a number of questions are still open for further exploration.

Accuracy vs. Efficiency

Both accuracy and efficiency of optical flow algorithms are important as far as real world applications are concerned. We emphasize higher accuracy, and thus have given less weight to the considerations of efficiency. Experiments reported in this thesis were performed on a Pentium II personal computer with a 233 MHz processor. It takes approximately 2 minutes to compute the optical flow field of a pair of 256×256 images using the two-layer “Skin and Bones” method described in Chapter 5. Obviously, this is currently an off-line process. The “Skin and Bones” model is a general purpose optical flow method, which provides motion estimates at all pixels. The algorithm is inherently parallel, but our current implementation is sequential, which uses a single processor. Note that we use the first-order system in the inter-patch smoothness term, such that each patch is dependent on its four nearest neighbors. A parallelism can be simply implemented to

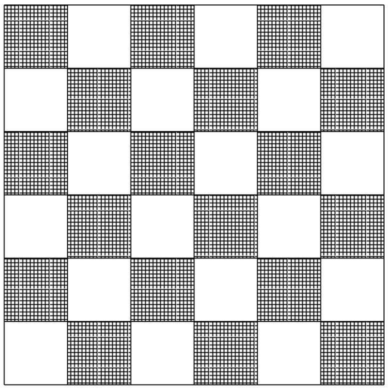


Figure 9.1: A parallelism scheme for the "Skin and Bones" model.

update the motion estimates of all the patches, such that each processor updates one of the black patches in the first iteration, then updates one of the white patches in the second iteration (see Figure 9.1). Such a parallelism would reduce the computational cost to approximately 3.75 seconds using 32 Pentium II 233 MHz processors to update 64 patches of size 32×32.

How to realize a faster algorithm of the model from software viewpoint needs to be further addressed. Bab-Hadiashar and Suter [8] use a subset of randomly selected motion constraints to compute K temporary motion estimates. Within a local window centered on each pixel, one of the K estimates is selected and used to remove the outliers in the window. The final motion estimate is the least squares approximation of the motion of inliers. Their algorithm has the complexity $O(K * W * N)$, where W is the size of the local window and N is the size of the image. Using a subset of motion constraints can save computation time, however, the random sampling scheme is not appropriate for the "Skin and Bones" model. Instead, sampling should be done in different orientations for different layers. Moreover, when a patch contains little texture, its motion is determined primarily by the regularization term (skin). If these patches are detectable so that their data terms can be dropped in the optimization process, the computational cost would be reduced. To detect unreliable patches, we would need to define a relative reliability measure that takes into account both the data term and the regularization term.

Motion Segmentation from Global Motions

The optical flow field is commonly considered a low-level representation. On the other hand, the layer representations can be used as intermediate representations for recognition, navigation, and video compression. It is well known that the estimation of the optical flow on one hand, and the segmentation of the image with respect to the apparent motion on the other hand, are two important issues in motion analysis. Solutions of these problem should compliment each other to provide accurate results as well as rich structure information. The advantage of layered representations is to separate the scene into coherent regions with homogeneous motion. The problem of motion estimation and motion segmentation can thus be solved simultaneously with, for example, methods based on mixture models.

We have shown numerous examples using the multi-layer motion estimation method globally in this thesis. To evaluate a method of recovering a global layered representation within the entire image region, a very important measure is the spatial support maps corresponding to each motion component. There are often two types of support maps, the *estimation* and the *a posteriori* support maps. The former is the support map used while computing the motion estimates, which corresponds to the ownership weights. The latter is usually computed once the number of motion components and the parametric motion model corresponding to each component have been estimated[1]. Our method only provides the estimation maps, which are likely to be smoothed due to the spatial smoothness prior used in the objective function. However, these ownership maps may still be locally "speckled". Given the estimation maps and motion estimates, MRF models are widely used to produce a posteriori support maps through modeling smoothness of layer ownerships. Such method can be used to produce a better segmentation of the scene.

Global+Local Layers

In this thesis, we demonstrated the need to estimate parametric motion models locally, and did so by tiling images into small patches. The "Skin and Bones" model is a local

[1] MRF-based methods do not make any distinction between these two support maps.

layered representation of the image motion, which is motivated by the need of an accurate, robust, and general motion estimation method. The relationship between this local layered representation and the global layered representation still needs to be exploited.

For example, considering the scene with an independently moving object and a background with complicated structures, such as a sequence taken by a moving camer at the Yosemite valley when an aircraft is flying through, a global layer representation is not appropriate since it will not recover the Yosemite valley well (see Figure 4.15). However, the local layer representations may not recover the aircraft with a coherent support map. What is desired is a combined representation that defines the motion of the Yosemite valley with local motion layers and the motion of the aircraft with a global layer and a segmentation map. Such a representation should result in accurate and flexible motion estimates for motions of complex surfaces, as well as a layered representation at the object/surface level for motions of rigid objects.

Other Formulations of the "Skin"

In our previous formulation [58] of the regularization term (skin), differences between affine parameters in neighboring patches were employed. This formulation may result in slightly blocked estimates, therefore, the smoothness is not achieved at the pixel level. Instead, we formulated the regularization term using differences between image velocities at the boundaries of patches in this thesis. Flow is smoothed between patches, however, it may not be smoothed inside the patch that contain little texture (see example of the Marbled Block sequence in Chapter 5). Other formulations of the "skin" still need to be investigated. One possible approach is to apply the transparent regularization term at every pixels within the patch. Obviously, it is the most computationally expensive formulation. However, such a term, which treats all the pixels same, favors a smoothed flow at the pixel level.

Confidence Measures

We have not addressed the issue of assigning confidence measures to the estimated motion vectors. In regions that lack brightness variation, the motion estimates may be poor. In this situation, a confidence estimate is useful, and can be exploited by processes that rely on optical flow as input.

Hierarchy of Motion Models

The accuracy of the method could be improved by extending the affine model to a quadratic flow model to account for the motion of planar regions. This suggests an algorithm in which there is a hierarchy of models, with varying complexity, operating at different spatial scales. In regions of decreasing size we might have planar, affine and translational patches. The coarse scale patches provide a coarse estimate and finer resolution patches, with more general flow models, are used to refine the solution taking the coarse level as a prior constraint. Such a hierarchical system could be implemented within the framework described here.

Summary

This thesis has developed a new motion estimation model that combines the regularization techniques and the area-based regression techniques. The "Skin and Bones" model, we believe, is an *accurate, dense, flexible, and robust* optical flow method. However, the problem of image motion estimation has not been solved completely. We have discussed some of the open issues and future directions above, which are primarily concerned with the early (the optical flow field) and intermediate (layered representations) stages of motion processing and understanding. To obtain high-level interpretations from the video sequences, such as recognition of *activities* [60, 69] and extraction of *content* [59], motion information has been used in applications defined in constrained domains. The model developed in this thesis is a general purpose approach that can be adapted to special applications.

Bibliography

[1] E. H. Adelson and H. R. Bergen. Spatio-temporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(1):284–299, 1985.

[2] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):381–401, 1985.

[3] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[4] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proc. of IEEE International Conference on Computer Vision*, pages 777–784, June 1995.

[5] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proc. of European Conference on Computer Vision*, volume 2, pages 316–327, May 1994.

[6] S. Ayer, P. Schroeter, and J. Brigger. Time-varying motion estimation using orthogonal polynomials and applications. In *Proc. of International conference on Pattern Recognition*, pages 409–414, Oct 1994.

[7] Serge Ayer. *Sequential and Competitive Methods for Estimation of Multiple Motions.* PhD thesis, Swiss Federal Institute of Technology, 1996.

[8] A. Bab-Hadiashar and D. Suter. Optic flow calculation using robust statistics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 988–993, 1997.

[9] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 434–441, Jun 1998.

[10] J.L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1), 1994.

[11] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *Proc. of European Conference on Computer Vision*, pages 237–252, May 1992.

[12] J. R. Bergen, P. J. Burt, , R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896, Sep 1992.

[13] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, B*, 48(3):259–302.

[14] M. J. Black. Recursive non-linear estimation of discontinuous flow fields. In *Proc. of European Conference on Computer Vision*, volume 1, pages 138–145, May 1994.

[15] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1991.

[16] M. J. Black and P. Anandan. The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan 1996.

[17] M. J. Black, D. Fleet, and Y. Yacoob. A framework for modeling appearance changes in image sequences. In *Proc. of IEEE International Conference on Computer Vision*, India, Jan 1998.

[18] M. J. Black and A. D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, Oct 1996.

[19] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–92, July 1996.

[20] M. J. Black and R. Rosenholtz. Robust estimation of multiple surface shapes from occluded textures. In *International Symposium on Computer Vision*, Miami, FL, November 1995.

[21] M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[22] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 561–567, 1997.

[23] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, 1987.

[24] P. Bouthemy and E. Francois. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *International Journal of Computer Vision*, 10(2):157–182, 1993.

[25] M. Campani and A. Verri. Computing optical flow from an over-constrained system of linear algebraic equations. In *Proc. of IEEE International Conference on Computer Vision*, pages 22–26, 1990.

[26] W. Chen, G. B. Giannakis, and N. Nandhakumar. Spatio-temporal approach for time-varying image motion estimation. In *Proc. of International conference on Image Processing*, pages 411–416, 1994.

[27] I. Cohen and I. Herlin. Optical flow and phase portrait methods for environmental satellite image sequence. *Proc. of European Conference on Computer Vision*, 2:141–150, 1996.

[28] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John wiley and Sons Inc., 1991.

[29] Franccois Dagognet. *Etienne-Jules Marey: A Passion for the Trace.* Zone Books, New York, 1992.

[30] T. Darrell and A. Pentland. Robust estimation of multi-layer motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, Princeton, NJ, October 1991.

[31] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, May 1995.

[32] J. W. Davis. Appearance-based motion recognition of human actions. Technical report #387, MIT Media Lab, 1996.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B(39):1–38, 1977.

[34] M. Etoh and Y. Shirai. Segmentation and 2D motion estimation by region fragment. In *Proc. of IEEE International Conference on Computer Vision*, pages 192–199, 1993.

[35] C. L. Fennema and W. B. Thompson. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9:301–315, 1979.

[36] D. Fleet. *Measurement of image velocity.* Kluwer Academic Press, Norwell, MA, 1992.

[37] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[38] E. Francois and P. Bouthemy. Multiframe-based identification of mobile components of a scene with a moving camera. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 166–172, 1991.

[39] D. Geiger and R. A. M. Pereira. The outlier process. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 61–69, 1991.

[40] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[41] S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, LII-4:5–21, 1987.

[42] A. Giachette and V. Torre. Refinement of optical flow estimation and detection of motion edges. In *Proc. of European Conference on Computer Vision*, volume 2, pages 151–160, 1996.

[43] S. Gold, C. P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness. Fast algorithms for 2D and 3D point matching: Pose estimation and correspondence. Technical Report YALEU/DCS/RR-1035, Yale University, May 1994.

[44] H. Gu, M. Asada, and Y. Shirai. The optimal partition of moving edge segments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 367–372, 1993.

[45] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: the Approach Based on Influence Functions.* Probability and mathematical statistics. John Wiley & Sons, 1986.

[46] J. G. Harris, C. Koch, E. Staats, and J. Luo. Analog hardware for detecting discontinuities in early vision. *International Journal of Computer Vision*, 4(3):211–223, Jun 1990.

[47] D. J. Heeger. Optical flow from spatiotemporal filter. *International Journal of Computer Vision*, 1:279–302, 1988.

[48] Y. C. Ho and R. C. K. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Trans. on Automatic Control*, 9:333–339, Oct 1964.

[49] B. K. P. Horn. *Robot Vision.* The MIT Press, Cambridge, Massachusetts, 1986.

[50] B. K.P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, August 1981.

[51] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representation. In *Proceedings 12th International Conference on Pattern Recognition*, pages 743–746, 1994.

[52] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. of European Conference on Computer Vision*, pages 282–287, 1992.

[53] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, 1994.

[54] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conference on Computer Vision*, pages 343–356, 1996.

[55] A. Jepson and M. J. Black. Mixture model for optical flow computation. *Ingmer Cox, Pierre Hansen, and Bela Julesz, editors, Partiting Data Sets: With Applications to Psychology, Vision and Target Tracking*, pages 271–286, April 1993. DIMACS Workshop.

[56] A. Jepson and M. J. Black. Mixture models for image representation. Technical Report ARK96-PUB-54, University of Toronto, March 1996.

[57] S. X. Ju and M. Black. Time-to-contact from active tracking of motion boundaries. *Intelligent Robots and Computer Vision XIII: 3D Vision, Product Inspection, and Active Vision, SPIE 2354*, Oct 1994.

[58] S. X. Ju, M. Black, and A. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 307–314, June 1996.

[59] S. X. Ju, M. J. Black, S Minneman, and D Kimber. Summarization of video-taped presentations: Automatic analysis of motion and gesture. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):686–696, Sep 1998.

[60] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Second International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Killington, VM, October 1996.

[61] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans ASME Journal of Basic Engineering*, 1960.

[62] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:229–244, 1987.

[63] J. J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica acta*, 22(9):773–791, 1975.

[64] J. Konrad and E. Dubois. Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[65] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989.

[66] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, BC, Canada, 1981.

[67] W. J. Maclean, A. D. Jepson, and R. C. Frecker. Recoverying of egomotion and segmentation of independent object motion using the EM algorithm. In *BMVC*, 1994.

[68] S. Madarasmi, D. Kersten, and T. C. Pong. Multi-layer surface segmentation using energy minimization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 774–775, New York, June 1993.

[69] R. Mann, A. Jepson, and J. M. Siskind. Computational perception of scene dynamics. In *Proc. of European Conference on Computer Vision*, pages 528–539, 1996.

[70] J. L. Marroquin. Surface reconstruction preserving discontinuities. Technical Report A.I. Memo 792, MIT, August 1984.

[71] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithm for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236, Sep 1989.

[72] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc,, N.Y., 1988.

[73] E. Memin and P. Perez. A multigrid approach for hierarchical motion estimation. In *Proc. of IEEE International Conference on Computer Vision*, pages 933–938, India, Jan 1998.

[74] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, Mar 1987.

[75] Eadweard Muybridge. *The Human Figure in Motion*. 1955, Dover Publications, New York.

[76] H. Nagel. Optical flow estimation and the interaction between measurement errors as adjacent pixel positions. *International Journal of Computer Vision*, 15(5):271–288, 1995.

[77] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, September 1986.

[78] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, Nov 1987.

[79] J. M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models in complex image sequence. In *7th European Conference on Signal Processing*, pages 411–414, Edinburgh, Scotland, Sep 1994.

[80] J. M. Odobez and P. Bouthemy. MRF-based motion segmentation exploiting a 2D motion model robust estimation. In *Proc. of International conference on Image Processing*, 1995.

[81] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, 1992.

[82] M. Otte and H. H. Nagel. Optical flow estimation: Advances and Comparisons. In *Proc. of European Conference on Computer Vision*, pages 51–60, 1994.

[83] J. Rissanen. *Encyclopedia of Statistical Sciences*, volume 5, chapter Minimum Description Length Principle, pages 523–527. John Wiley and Sons, 1985.

[84] J. Rissanen. *Stochastic Complexity In Statistical Inquiry.* World Scientific, 1989.

[85] A. Rognone, M. Campani, and A. Verri. Identifying multiple motions from optical flow. *Proc. of European Conference on Computer Vision*, pages 258–266, May 1992.

[86] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiley & sons, 1987.

[87] H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–831, 1996.

[88] D. Shulman and J. Hervé. Regularization of discontinuous flow fields. In *Proc. Workshop on Visual Motion*, pages 81–85, Irvine, CA, Mar 1989.

[89] H. Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proc. of IEEE International Conference on Computer Vision*, Jan 1998.

[90] A. Singh. *Optical Flow Computation: A Unified Perspective.* IEEE Computer Society Press, Los Alamitos, California, 1991.

[91] A. Singh. Incremental estimation of image flow using a Kalman filter. *J. of Visual Communication and Image Representation*, 3(1):39–57, Mar 1992.

[92] R. Szeliski and J. Coughlan. Spline-based image registration. *International Journal of Computer Vision*, 22(3):199–218, 1997.

[93] R. Szeliski and H. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1211, 1996.

[94] R. Szeliski and D. Tonnesen. Surface modeling with oriented partical systems. *Computer Graphics*, 26(2):185–194, July 1992.

[95] D. Terzopoulos. Regularization of inverse visual problem involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.

[96] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–97, 1989.

[97] N. Vasconcelos and A. Lippman. Empirical Bayesian EM-based motion segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 527–532, 1997.

[98] N. Vasconcelos and A. Lippman. A spatiotemporal motion model for video summarization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1998.

[99] A. Verri and T. Poggio. Against quantitative optical flow. In *Proc. of IEEE International Conference on Computer Vision*, pages 171–180, 1987.

[100] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, Sep 1994.

[101] A. M. Waxman and K. Wohn. Contour evolution, neighbourhood deformation and global image flow: Planar surfaces in motion. *Int. J. of Robotics Research*, 4:95–108, 1985.

[102] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *Proc. of IEEE International Conference on Computer Vision*, pages 12–20, May 1993.

[103] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, Jun 1997.

[104] Y. Weiss and E. H. Adelson. Perceptually organized EM: a framework for motion segmentation that combines information about form and motion. In *Proc. of IEEE International Conference on Computer Vision*, page submitted, 1995.

[105] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 321–326, San Francisco, CA., 1996.

[106] P. Werkhoven and J. J. Koenderink. Extraction of motion parallax structure in the visual system. *Biological Cybernetics*, 63:185–199, 1990.

[107] Y. Wu, T. Kanade, J. Cohn, and C. Li. Optical flow estimation using wavelet motion model. In *Proc. of IEEE International Conference on Computer Vision*, India, Jan 1998.

[108] Y. Yacoob and L. Davis. Temporal multi-scale models for flow and acceleration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 921–927, 1997.

[109] A. Yuille, P. Burgi, and N. M. Grzywacz. Visual motion estimation and prediction: A probabilistic network model for temporal coherence. In *Proc. of IEEE International Conference on Computer Vision*, pages 973–978, Jan 1998.

[110] A. Yuille, T. Yang, and D. Geiger. Robust statistics, transparency and correspondence. Technical Report 7, Harvard Robotics Laboratory, 1990.

[111] H. Zheng and S. D. Blostein. Motion-based object segmentation and estimation using the MDL principle. *IEEE Transactions on Image Processing*, Sep 1995.