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

Shanon X. Ju* Michael J. Black† Allan D. Jepson*

*Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4 Canada
Phone: (416)978–6488, FAX: (416)978–1455
Email: {juxuan/jepson}@vis.toronto.edu

†Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304
Phone: (650)812–4745, FAX: (650)812-4334
Email: black@parc.xerox.com

Abstract

This paper describes a method 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 patchs. 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 bones and transparent regularization, is formulated as an objective function that is minimized using a variant of the EM-algorithm. Experiments with synthetic and natural images are provided throughout the paper to illustrate the method.

Submitted as a **Regular Paper.**

Index terms: Optical flow, Layers, Regularization, Affine motion, Robust estimation, Transparency.

Contact Information:

Shanon X. Ju
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4
Canada

Phone: (416)978–7726
FAX: (416)978–1455
Email: juxuan@vis.toronto.edu

1 Introduction

Recent work on optical flow can be seen as trying to find a balance between dense optical flow schemes that use regularization and parameterized approaches that use regression [3, 10, 44, 51]. Dense optical flow methods, as epitomized by the method of Horn and Schunck [25], require only local image measurements and integrate information over larger areas via regularization. These 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 [9, 23, 35, 36, 42]. 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 [6]. 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) 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. Jepson and Black [26] referred to this problem surrounding the appropriate choice of region size as the *generalized aperture problem*. The work described here combines 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.

We consider first a basic approach that estimates multiple motions within small patches. The approach tiles the image with a fixed set of patches that are 32 pixels on each side. 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" [3, 16, 26, 48, 51]. Pixels are assigned to layers and the motion of each layer is estimated using a robust mixture model formulation [3, 12, 26, 34] that accounts for "outliers" [22] 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 EM algorithm [34]. In addition, an incremental

revision process is used to find the most appropriate number of layers needed in a patch. This basic approach recovers layered affine motions simultaneously within a patch, and is applied to each image patch independently.

Within patches of this size, the affine motion models may be underconstrained due to insufficient spatial variation in the image, and hence we add a regularization term that embodies the assumption that the affine motions at the boundary of a patch should be smooth. Since each image patch may have multiple motion estimates associated with it, how can one regularize such a set of layered measurements? If the motions could somehow be grouped into consistent layers (e.g. [27, 48]) 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*.

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 models within a patch and a prior term that enforces spatial smoothness between the estimated affine motions and those of neighboring patches.

The following section reviews related work on layered motion estimation. The algorithm is

then developed in pieces with examples to motivate and illustrate each piece. First, robust layered affine motion estimation (bones) is introduced in Section 3. In Section 4 we first propose the 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".

2 Related Work

Parameterized optical flow methods [1, 6, 17, 18, 29, 32, 49] 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. 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. Bergen *et al.* [7] use an iterative registration algorithm to account for multiple global motions in the scene. Darrell and Pentland [16] introduce the idea of estimating global motions in *layers* and present an optimization scheme using ideas from robust statistics. Jepson and Black [26] assume that the motion in the scene can be represented by a mixture of distributions and they use the EM algorithm to decompose the motion into a fixed number of layers. Black and Anandan [9] use robust statistics (M-estimation) [22] to estimate a dominant motion in the scene and then fit additional motions to outlying measurements. Bab-Hadiashar and Suter

[4] develop a similar, but more robust, approach using the the optimally robust Least Median of Squares technique [40] to estimate the dominant motion in a region. Yuille et. al [53] 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. All of these 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.

Another set of approaches apply parametric models to coarse flow fields by grouping flow vectors into consistent regions. Wang and Adelson [47] 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 [32]. Unlike the method described here, only a single translational motion is computed in each patch rather than multiple affine motions. They then use K-means clustering to group motion estimates into regions of consistent affine motion. Similarly, Adiv [1] uses a Hough technique to group flow measurements into regions consistent with the motion of planar surfaces. These approaches, like the regression approaches above, assume that the image motion can be represented by a small number of layers within an image region.

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. What the above methods no not address is how to estimate the correct number of layers or how to deal with motions that are significantly more complex than simple polynomials (like affine).

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 [16] and Ayer and Sawhney [3] 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. 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" [40] that pull the solution away from the desired local motion. Weiss and Adelson [52] 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.

To model complex motions in layers Weiss [51] 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 are needed still needs to be addressed. This approach gives up one of the benefits of parameterized motions models which is that they provide a concise description of the motion in a region which can be used for recognition [13]. To model complex motions with concise parameterized models, Black *et. al* [14] propose "learning" these models from examples.

In this paper we argue that what is needed are methods that have the accuracy of the parameterized area-based approaches but which are applied locally. There have been a number of recent attempts to apply parameterized motion models to smaller, more local, image regions. For example Black and Jepson [10] 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.

Bab-Hadiashar and Suter [4] apply their robust Least Median of Squares technique *independently* within image patches centered on each pixel in the image. This very 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 provide a confidence measure that can be used to ignore these unreliable estimates.

Szeliski and Shum [44] 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. This is similar to the adaptive window idea used by Okutomi and Kanade for stereo matching [38]. While the approach moves towards our goal of a local parameterized motion estimate, it has two serious limitations. First, only a single motion can be present within a patch which would preclude the representation of transparent motion. With transparent motion or fragmented occlusion, the region size would have to shrink to a point. This brings to light the second limitation which is that the spline-based representation does not readily admit spatial discontinuities. With this method it does not appear feasible to represent a highly discontinuous or transparent flow field.

Instead we take fixed sized regions of the image that are sufficiently large to estimate affine flow (these can be overlapping as in [4] or non-overlapping as in our experiments). We then estimate multiple affine motions using a layered motion estimation scheme similar to those used by [3, 9, 12, 26, 41, 53] and we automatically estimate the number of layers necessary for each region. Since the motion of some patches may be underconstrained, we are faced with the problem of enforcing spatial smoothness between the motions of neighboring patches.

We define a spatial smoothness constraint on the affine parameters of neighboring patches. This is similar in spirit to the constraints used in the oriented particle systems of Szeliski and Tonnesen [45]. 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.* [33] 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 [9] provides a simple and elegant solution to the problem.

3 Mixture of Locally Affine Motions (Bones)

We first introduce the affine flow model and then describe a layered representation based on mixture models [34].

3.1 Affine Motion Estimation

Within a given image region, $\mathcal{R}$, we model the image motion using a low-order polynomial [6]. When the image region is small, an affine (linear) transformation can well approximate the image flow for a smooth surface [30]. The affine model of image motion is defined as

$$u(x,y) = a_0 + a_1(x - x_c) + a_2(y - y_c), \quad (1)$$

$$v(x,y) = a_3 + a_4(x - x_c) + a_5(y - y_c), \quad (2)$$

where $u(x,y)$ and $v(x,y)$ are the horizontal and vertical components of the image velocity at the image point (x,y), and the a_i are parameters to be estimated relative to some region center (x_c, y_c). Using vector notation this can be rewritten as

$$\mathbf{u}(\mathbf{x};\mathbf{a}) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} 1 & x - x_c & y - y_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x - x_c & y - y_c \end{bmatrix} \mathbf{a}, \quad (3)$$

where boldface lowercase letters denote vectors and, in particular, $\mathbf{x} = [x,y]^T$ and $\mathbf{a}$ denotes the vector $[a_1, a_2, a_3, a_4, a_5, a_6]^T$.

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

$$I(\mathbf{x},t) = I(\mathbf{x} + \mathbf{u}(\mathbf{x};\mathbf{a}), t-1), \quad \forall \mathbf{x} \in \mathcal{R} \quad (4)$$

where $\mathcal{R}$ denotes the pixels in some region, 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 $\mathbf{u}(\mathbf{x};\mathbf{a})$ in region $\mathcal{R}$. 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 [6, 24]

$$\nabla I \cdot \mathbf{u}(\mathbf{x};\mathbf{a}) + I_t = 0, \quad \forall \mathbf{x} \in \mathcal{R} \quad (5)$$

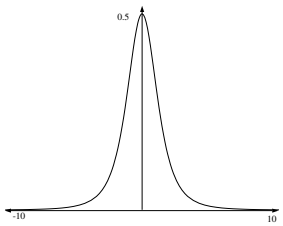


Figure 1: Weight function $l(r, \sigma)$.

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 $(\mathbf{x}, t)$.

3.2 Mixtures of Robust Bones

We deal with several motions within a single region using a straightforward extension of the mixture model approach described in [26]. 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 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:

$$E(\mathbf{a}_i) = \sum_{\mathbf{x} \in \mathcal{R}} m_i(\mathbf{x}, \sigma)(\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t)^2 \tag{6}$$

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 6. We provide the details of each of these steps below.

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

$$\frac{1}{K} l(r(\mathbf{x}; \mathbf{a}_i), \sigma) = p(I(\mathbf{x}, t) \mid \mathbf{a}_i), \tag{7}$$

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 ([26, 34]) or a robust distribution as in ([12, 28]). 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 1.

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

be

$$l_i(\mathbf{x}, \sigma) = l(\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t, \sigma) = \frac{\sigma^3}{(\sigma^2 + (\nabla I \cdot \mathbf{u}(\mathbf{x}; \mathbf{a}_i) + I_t)^2)^2}. \tag{8}$$

Notice in Figure 1 that constraints having a 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 [21, 26] 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. [40, 41]) and hence the likelihood of an outlier is

$$l_{\mathcal{L}+1}(\sigma) = \frac{\sigma^3}{(\sigma^2 + (2.5\sigma)^2)^2}. \tag{9}$$

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,

$$m_i(\mathbf{x}, \sigma) = l_i(\mathbf{x}, \sigma)/M(\mathbf{x}), \tag{10}$$

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 [34]). Note that the computation of ownership weights (Equation 10) is independent of the uniform scaling constant K in equation 7.

3.2.2 Layer Parameters

The ownership weights in Equation (10) 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

$$E(\mathbf{a}) = \sum_{\mathbf{x} \in \mathcal{R}} \sum_{i=1}^{\mathcal{L}} m_i(\mathbf{x}, \sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x}, \mathbf{a}_i) + I_t)^2 \tag{11}$$

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 [26] in two ways. The first is that in place of the likelihood function in Equation (8) 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 probability, 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.

3.2.3 Estimating and Annealing the Scale Parameter

The scale parameter, σ, controls the shape of the likelihood function and hence the assignment of constraints to layers. A small value of σ 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. We would like to estimate this parameter automatically from the data (cf. [41]).

We can estimate the scale parameter 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 $\hat{\sigma}_i$ to be

$$\hat{\sigma}_i^2 = \frac{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x};\mathbf{a}_i) + I_t)^2}{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)}. \tag{12}$$

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

$$l(r,\sigma) = \frac{2\sigma^3}{\pi(\sigma^2 + r^2)^2} \tag{13}$$

and hence the estimate $\hat{\sigma}_i$ will be biased. To estimate the correct σ for our distribution, we seek a constant κ such that $\kappa\sigma_i^2 = \hat{\sigma}_i^2$. Integrating an approximate version of Equation 12 over the range of inliers (values between $[-2.5\sigma, 2.5\sigma]$) gives

$$\int_{-2.5\sigma}^{2.5\sigma} l(r,\sigma) r^2 \, dr \; / \; \int_{-2.5\sigma}^{2.5\sigma} l(r,\sigma) \, dr \approx 0.54\sigma^2. \tag{14}$$

Therefore, the new scale parameter $\tilde{\sigma}_i$ for layer i can be estimated using:

$$\tilde{\sigma}_i = \sqrt{\frac{\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)(\nabla I \cdot \mathbf{u}(\mathbf{x},\mathbf{a}_i) + I_t)^2}{0.54\sum_{\mathbf{x}\in\mathcal{R}} m_i(\mathbf{x},\sigma_i)}}. \tag{15}$$

Finding the global minimum of Equation (11) when σ is small is complicated by the existence of local minima. Jepson and Black [27] used an EM-algorithm coupled with deterministic annealing to estimate a layered representation of a gray-level image. The idea is to start with a sufficiently large σ so that no data are treated as outliers and constraints are shared by layers. Then as σ decreases the influence of outliers is gradually reduced and constraints are assigned to layers.

In this paper, we use the following annealing approach. At each iteration we compute the current value of σ_i by taking into account the estimated $\tilde{\sigma}_i$ in Equation (12) and a maximum and minimum value (σ_s and σ_f respectively).

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

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 2) and in our experiments we take $\sigma_{init} = 10$ and $\sigma_{min} = 1.5$. The estimated σ is bounded so that it will not decrease too fast or too slow. The annealing rate r_s is 0.95, and r_f is 0.5. These parameters are the same for all the experiments in this paper.

3.2.4 Implementation

To estimate the affine transformation of each layer we first construct a Gaussian pyramid representation of the images under consideration. Spatial and temporal derivatives are computed using the simple approach in [24]. The affine transformations are computed at a coarse level and then, at the next finer level, the estimated transformations are used to register the two images by warping one

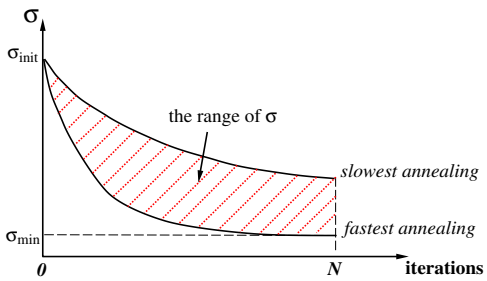


Figure 2: σ annealing.

towards the other using bilinear interpolation (note that this must be done for each of the layers, and that the affine parameters a_1 and a_3 in each layer are multiplied by a factor of two). The affine transformations at this finer level are estimated and then composed with the affine transformations from the coarse level (in our implementation, we simply add them). This process is repeated down to the finest level in the pyramid while the transformation is updated at each stage.

In this coarse-to-fine strategy, the regions in which the motions are estimated are also reduced in size by a factor of two for each coarser level in the pyramid. 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. 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. The process iterates between the steps while the σ_i are annealed as described above. This gradient descent scheme in the E-step is a straightforward variation of that used in [9]. An alternative to gradient descent would be to solve the Equation (6) using weighted least squares estimation.

3.2.5 Example

An example should help clarify how the layered method behaves. Consider the synthetic "Skaters" image sequence shown in Figure 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

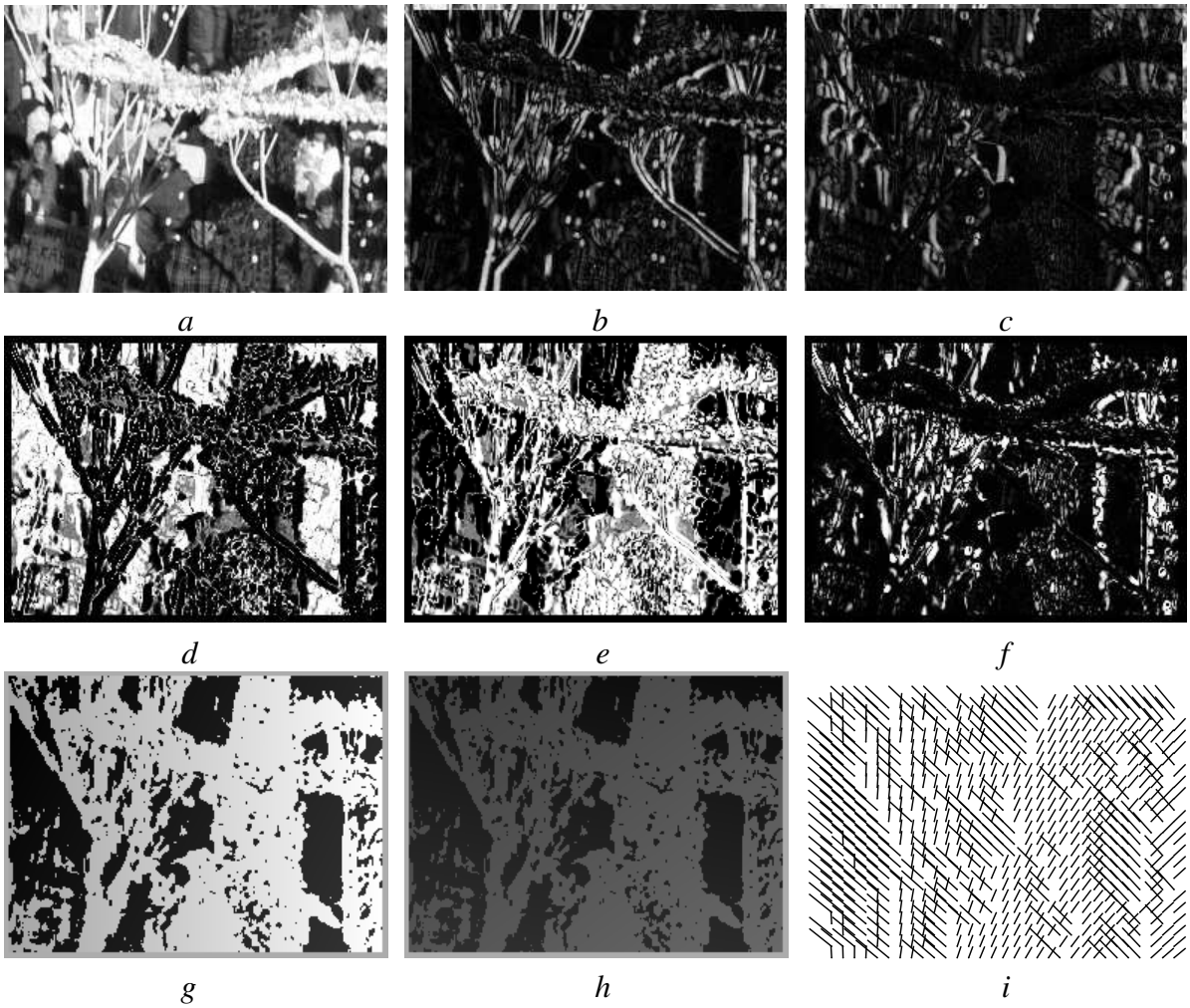


Figure 3: Fitting multiple motions (Skaters sequence). (a) First image in the sequence; (b) Absolute difference between the original image and the second image stabilized with respect to layer 1; (c) Absolute difference image with respect to layer2; (d) Weights for layer 1; (e) Weights for layer 2; (f) Weights for outlier layer; (g) Horizontal component of flow; (h) Vertical component of flow; (i) Vector field.

(a) (b)

Figure 4: Flower garden image sequence. (a) one image in the sequence; (b) grid of patches superimposed on the image.

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 pixelwise absolute intensity differences between these stabilized pairs are shown in Figure 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 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 predominantly correspond to areas near depth discontinuities or shadow boundaries. Figure 3 (g) and (h) show the horizontal and vertical motion at every pixel and Figure 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.

3.3 Tiling the Image

Consider applying the method in small patches over the image; for example, we choose non-overlapping 32×32 pixel patches for all experiments in this paper. We apply the method to the "flower garden" sequence shown in Figure 4. As shown in the figure, the image is segmented into fixed size patches which may span surfaces at a number of depths. The affine motions of each patch are estimated independently as described above.

In the case where just one layer is used in each image region, the robust bones method will tend

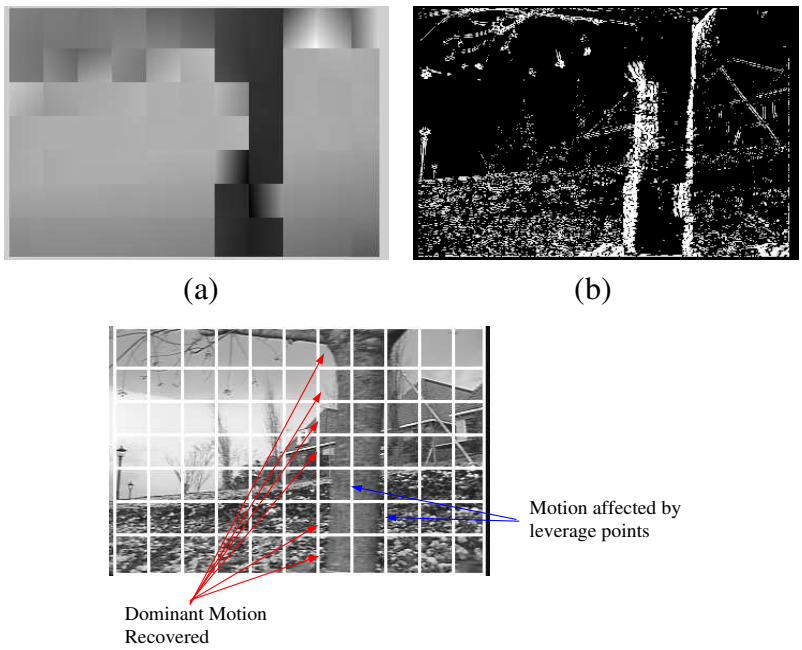


Figure 5: Flower garden; single-layer bones. (a) One layer: Horizontal component of flow; (b) One layer: Outliers where the estimated motion did not conform to the parameterized model; (c) some patches recover the dominant motion while others try to fit both foreground and background.

to recover the dominant motion within a region. This can be seen in the horizontal flow estimates in Figure 5 (a) (there is very little vertical motion), and the outliers in Figure 5 (b). Notice that in regions which span the tree boundary, one of two things occur. In some cases, the algorithm choses 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 white in Figure 5 (b). In the other cases, the algorithm attempts to fit both the foreground and background regions. For these cases, the motion constraints of the background can be thought of as leverage points that pull the solution away from the dominant motion (Figure 5 (c)).

When two layers are used, the motion at the boundary of the tree is estimated more accurately (Figure 6 (a)). Figure 6 (c) and Figure 6 (d) show the weights for the two motion layers within each region. Gray areas correspond to a weight near 0.5 and these regions indicate places where only one motion was present or where there is little texture so that the constraints could equally well belong to either layer (for example, in the sky). Regions that span a motion boundary have two

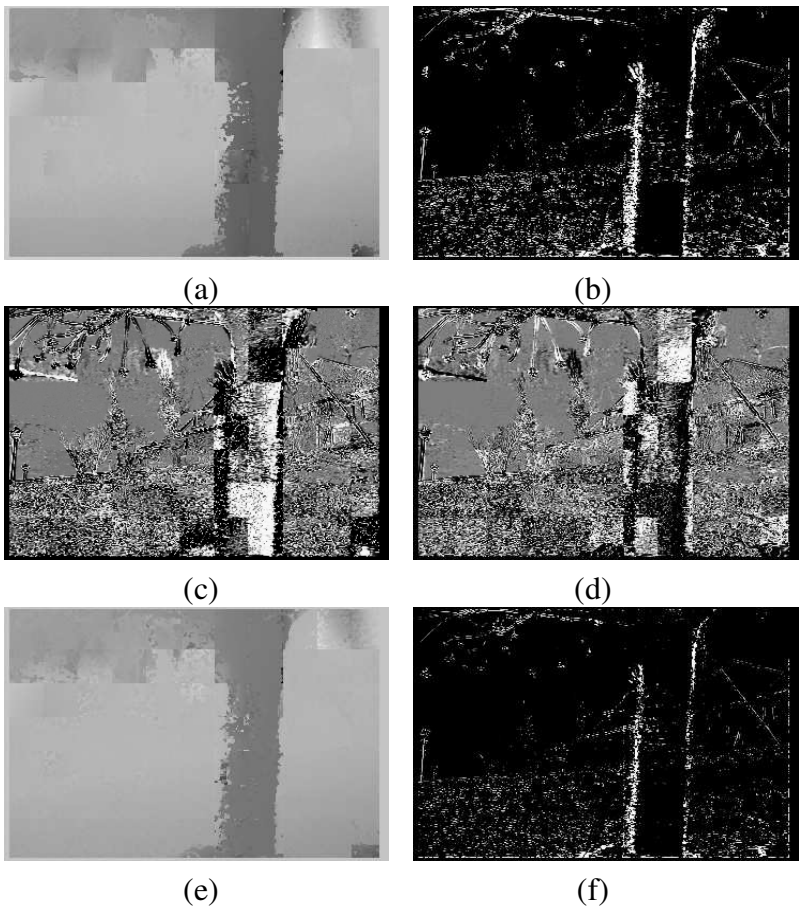


Figure 6: Flower garden: multi-layer bones. (a) Two layers: Horizontal flow; (b) Two layers: Weights for outlier layer; (c) Two layers: Weights for layer 1; (d) Two layers: Weights for layer 2; (e) Four layers: Horizontal flow; (f) Four layers: Weights for outliers layer.

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. Note that the motion of the flower-bed is not affine and hence increasing the number of layers improves the motion estimates.

While increasing the number of layers can improve the motion estimation, it also results in greater computational cost. However, when we increase the number of layers (Figure 6 (e) and (f)) to four, the improvement obtained diminishes. Clearly, only one or two distinct motions are estimated in most regions. In the next section we discuss a simple approach for choosing the appropriate number of layers in a region.

3.3.1 Estimating the number of layers

We consider an incremental approach that will add a new layer if the revision can improve the motion estimates significantly. In our implementation, the Minimum Description Length (MDL) principle is used to compare results with differing numbers of layers (see [3, 16, 31]). We will choose the number of layers that have the minimum encoding cost while explaining the observations best. For this purpose, the number of the bits required to encode the layered motions and the residuals are used.

The encoding has three parts, the first part consists of the encoding of the estimated affine motions. According to coding theory, the optimal coding cost of a continuous motion parameter a, in terms of some discrete set $\{a^1, a^2, ..., a^n\}$ of alternatives is given by:

$$C(a) = -\sum_{k=1}^{n} p(a^k) \log_2(p(a^k)), \tag{16}$$

where a^k are the discrete states used to approximate the parameter a, and $p(a^k)$ is the prior probability for discrete state a^k. As a rough approximation, we assume 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 1/100th of a pixel. We also assume an independent uniform prior for each coefficient. This gives an encoding cost for the six affine motion parameters of

$$C_{motion} = -2 \log_2 \frac{1}{3200} - 4 \log_2 \frac{1}{100} \approx 50. \tag{17}$$

The second part consists of the encoding of which layer each pixel is assigned to. If we ignore spatial correlations, then the optimal coding cost is:

$$C_{layers} = -|\mathcal{R}| \sum_{i=1}^{\mathcal{L}+1} p(layer_i) \log_2(p(layer_i)), \tag{18}$$

Where $|\mathcal{R}|$ is the number of pixels in region $\mathcal{R}$. We take the $p(layer_i)$ to be a uniform distribution, i.e., $p(layer_i) = \frac{1}{\mathcal{L}+1}$. In this case, the C_{layers} is simply $|\mathcal{R}| \log_2(1 + \mathcal{L})$, Where $\mathcal{L}$ is the number of layers in the region $\mathcal{R}$.

The last part consists of the encoding of pixel intensity $I(\mathbf{x}, t)$ of second image given the first image, the layer i, and the motion parameters $\mathbf{a}_i$. 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, $r(\mathbf{x}; \mathbf{a}_i)$. The residual r only needs to be encoded to within -0.5 and 0.5 grey levels, 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,

$$p(r^k(\mathbf{x}; \mathbf{a}_i) \mid \sigma_i) = \frac{2}{\pi}\, l(r^k(\mathbf{x}; \mathbf{a}_i),\ \sigma_i)\ \delta r \tag{19}$$

where $\delta r = r^{k+1} - r^k = 1$. This gives the cost to encode a residual at pixel $\mathbf{x}$ in layer i:

$$\begin{aligned} C_i(\mathbf{x}) &= -\log_2(p(r^k(\mathbf{x}; \mathbf{a}_i) \mid \sigma_i)) & (20)\\ &= 2\log_2(\sigma_i^2 + r^k(\mathbf{x}; \mathbf{a}_i)^2) - 3\log_2(\sigma_i) + \log_2(\frac{2}{\pi}). & (21) \end{aligned}$$

Finally, for the outlier layer, $C_{\mathcal{L}+1}(\mathbf{x}) = 8$. 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.

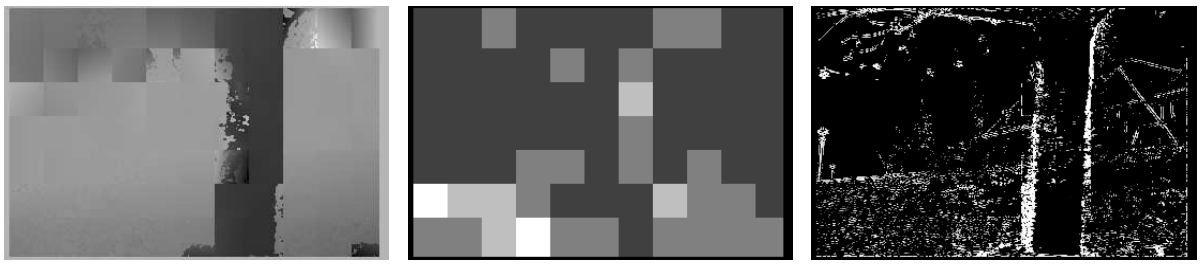
The overall code length for a region, $\mathcal{R}$ is defined to be the sum of all three parts:

$$C_{\mathcal{R}} = \mathcal{L} * C_{motion} + C_{layers} + \sum_{\mathbf{x}\in\mathcal{R}} C_{i(\mathbf{x})}(\mathbf{x}). \tag{22}$$

The incremental revision process starts with one layer in each patch and the motion for that layer is estimated. 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.

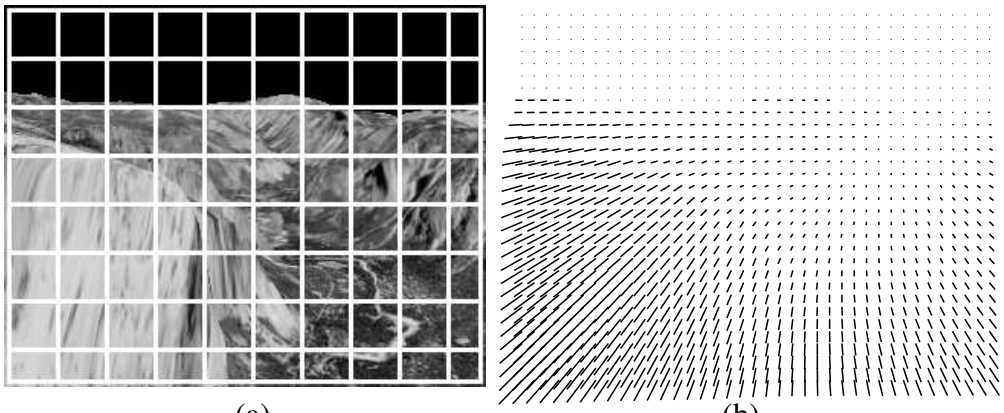
3.4 Examples: Multi-layer Bones

The example of using this incremental layered motion estimation for the "flower garden" sequence is given in Figure (7). Figure 7(b) shows that in about half of the regions that span the motion



(a) (b) (c)

Figure 7: Flower garden – estimating the number of layers. (a) Horizontal component of flow; (b) number of layers (one=dark-gray, two=medium-gray, three=light-gray, four=white). (c) Outliers (white).



(a) (b)

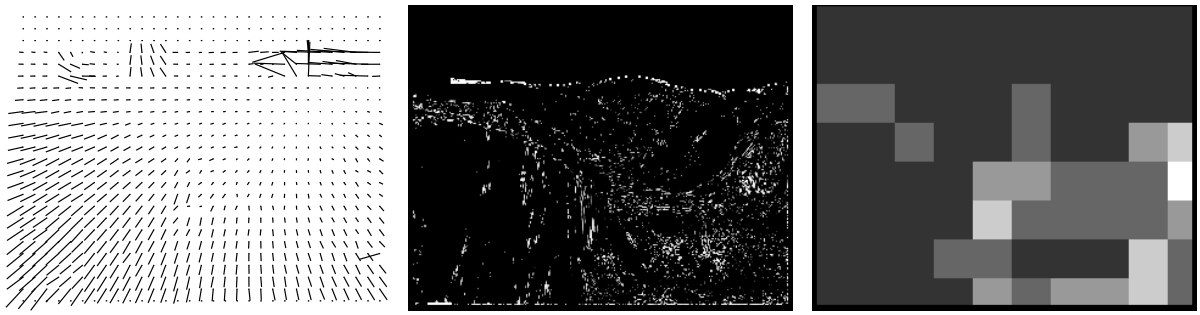
Figure 8: Yosemite Sequence, ground truth. (a) Image 11 in the sequence; (b) vector field.

boundaries, two layers (medium gray) are selected. While one layer (dark gray) is chosen for most of regions in sky and two or more layers is selected for the ground plane, since the motion of the flower-bed is more complex than affine.

To illustrate the accuracy of local affine "bones" we apply the method to two images in the synthetic Yosemite sequence[1], the first of which is shown in Figure 8(a) with the grid of patches superimposed. Figure 8(b) shows the vector-field version of the true motion.

The image was divided into equally sized patches (32×32 pixels at the finest level), and a three-level Gaussian pyramid was used in the coarse-to-fine processing. At the coarsest level, the size of each region is 4×4 pixels which is not sufficient to reliably fit more than one affine flow model. Therefore, for regions of this size, we fit only translational models (the parameters a_0 and

[1] This sequence was generated by Lynn Quam and provided by David Heeger.



(a) (b) (c)

Figure 9: Yosemite results – multi-layer bones. (a) Vector field; (b) Outliers (white). (c) number of layers (one=dark, two=gray, three=white).

a_3).

In order to display the results, at each pixel we show the motion for the layer which has the maximum ownership. Figure 9 (a) shows the vector field of the estimated flow. The pixels that were treated as outliers during the robust estimation are shown in Figure 9 (b). Figure 8 (c) shows the number of layers chosen to encode each region. In some cases, the affine model is a good approximation to the image motion and a single layer is sufficient. Some regions contain depth discontinuities and more than one layer is chosen. In some regions (particularly in the valley floor) more layers than necessary were chosen to capture the image motion. The regularization described in the following section will address this problem.

Since the sequence is synthetic, we can compute the error in the flow using the angular error measure of Barron *et al.* [5]. 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 $\arccos(\mathbf{v}_t \cdot \mathbf{v}_e)$. The performance of the algorithm can be quantified in Table 1. [2] Note that

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Bones:	0.163	3.80°	4.84°	18.2%	40.5%	57.0%	76.5%	94.2%

Table 1: Error results for the Yosemite fly-through sequence: Bones.

"Pixel Error" is the mean of the Euclidean distance (i.e. $\|\mathbf{u}_e - \mathbf{u}_t\|$, in pixels per frame) between

[2] Flow errors were not computed in the sky area since there is no ground truth for this region.

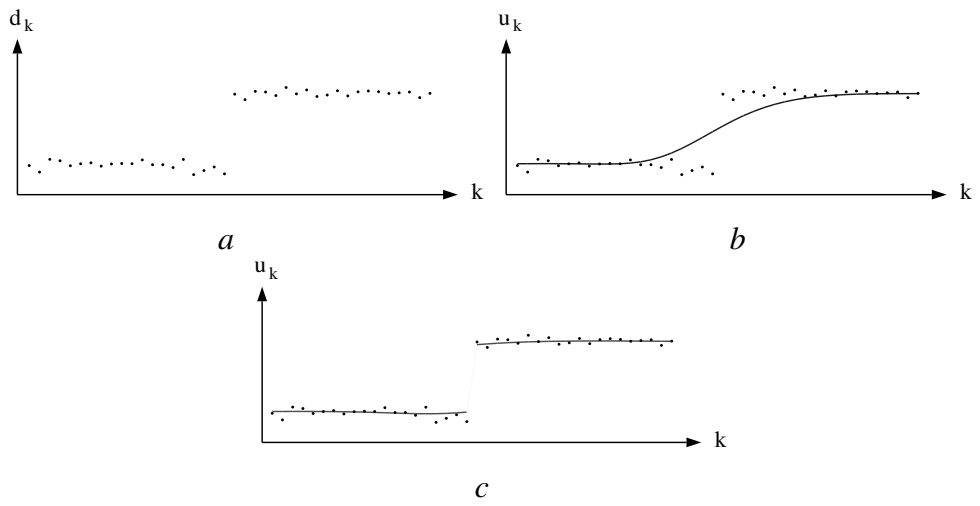


Figure 10: Regularizing discontinuous data. (a) Noisy data; (b) Least-squares fit to the data; (c) Piecewise smooth fit (robust regularization).

the estimated flow vector and the true flow vector [39], while "Average Error" refers to the mean angular error. Both measures are computed over the non-sky portion of the image.

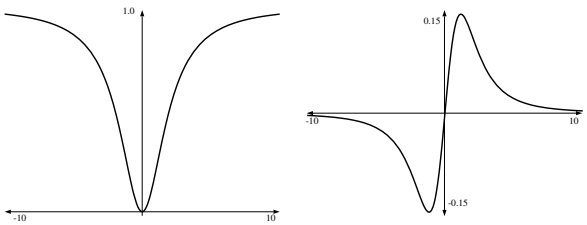
By visual inspection, it is clear that the motion field in Figure 9 is not as smooth as the true flow and shows a block structure. In some regions, most notably at the boundaries and the sky regions, the estimated motion is incorrect. The following section illustrates how a regularization term (skin) improves these locally affine estimates (bones).

4 Regularization with Transparency (Skin)

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 that extends a simple regularization problem to the transparent case.

4.1 Standard Regularization

We first consider the standard regularization problem without transparency. Figure 10*a* shows an example of noisy and discontinuous data (cf. Blake and Zisserman [15]). Given noisy data measurements, d_k, $1 \leq k \leq K$, our goal is to estimate a piecewise-smooth approximation, u_k, of



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

Figure 11: Robust error norm ρ and influence function ψ.

the true function. The standard regularization problem can be formalized as finding the u_k that minimize

$$E(\mathbf{u},\mathbf{d}) = \sum_{k=1}^{K} \left[(u_k - d_k)^2 + (u_k - u_{k-1})^2 \right], \tag{23}$$

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 10*b* which does not preserve the spatial discontinuity in the data. To account for discontinuities and outlying data measurements, we can convert this to the robust estimation problem [11, 19]

$$E(\mathbf{u},\mathbf{d}) = \sum_{k=1}^{K} \left[\rho(u_k - d_k, \sigma_D) + \rho(u_k - u_{k-1}, \sigma_S) \right], \tag{24}$$

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

$$\rho(x,\sigma) = \frac{x^2}{\sigma^2 + x^2} \tag{25}$$

which is used in [9, 12, 20] and is shown in Figure 11.[3] The shape of the ρ function in Figure 11 is such that it "rejects", or downweights, large residual errors. The function $\psi(x,\sigma)$, also shown in Figure 11, is the derivative of ρ and characterizes the influence of the residuals [22]. As the magnitudes of residuals grow beyond a point their influence on the solution begins to decrease and the value of $\rho(\cdot)$ approaches a constant. The value σ effects this point at which the influence of outliers begins to decrease. Minimizing the robust formulation in Equation (24) results in the piecewise smooth fit shown in Figure 10*c*.

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

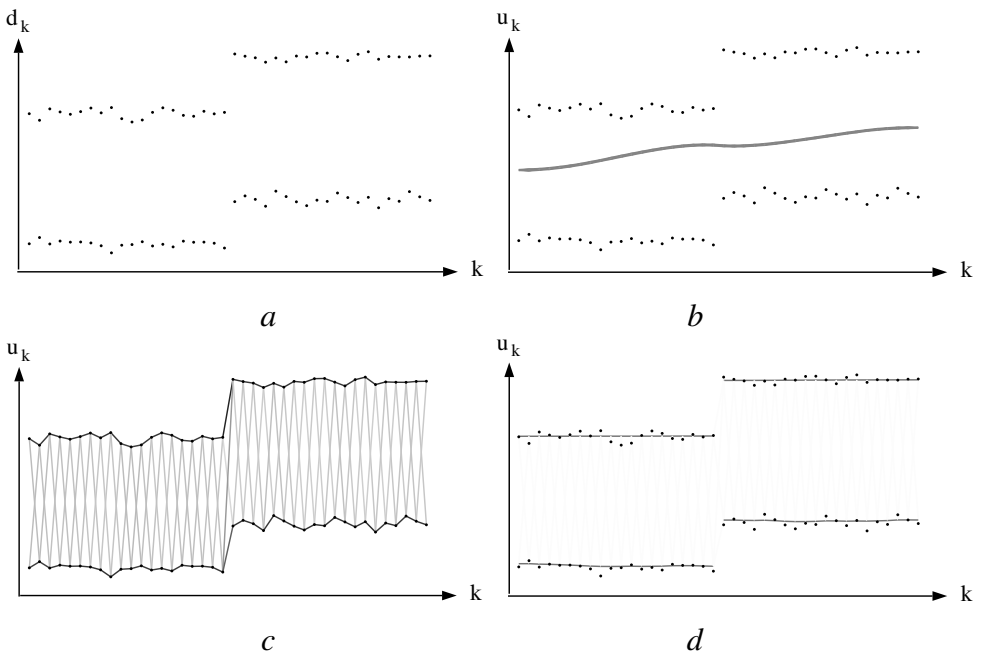


Figure 12: Transparent regularization. (a) Transparent data; (b) Single-layer regularization; (c) Weight of the connection between neighboring points in all layers (large σ); (d) Transparent regularization, piecewise smooth result (small σ).

4.2 Regularization with Transparency

The robust formulation can be extended naturally to cope with transparency. Consider the noisy data in Figure 12*a*. 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 12*b*.

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

$$E(\mathbf{u}, \mathbf{d}) = \sum_{k=1}^{K} \sum_{i=1}^{\mathcal{L}} \left[\rho(u_{k,i} - d_{k,i}, \sigma_D) + \sum_{j=1}^{\mathcal{L}} \rho(u_{k,i} - u_{k-1,j}, \sigma_S) \right], \tag{26}$$

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 (26) smoothes the data without explicitly assigning data to particular layers.[4]

To illustrate this, Figure 12*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 (26) 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 12*d* shows the result of minimizing Equation (26) using gradient descent with a continuation method. The solution converges to the desired piecewise-smooth, and transparent, surface interpretation.

4.3 Optical Flow

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. In previous formulations, 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

[4] Equation (26) can be reformulated as a weighted least squares problem much like Equation (11) and solved using an EM algorithm.

problem in a straightforward way to allow the regularization of multi-layer bones.

We define the Skin & Bones model by adding a spatial coherence term to the multi-layer data term in Equation (11). The smoothness term is defined to examine all neighboring layers as described above. The new objective function for layer i of an image patch $\mathbf{s}$ is

$$E_i(\mathbf{a}_i(\mathbf{s})) = \left[\frac{1}{|\mathcal{R}(\mathbf{s})|}\sum_{\mathbf{x}\in\mathcal{R}(\mathbf{s})} m_i(\mathbf{x},\sigma_i(\mathbf{s}))(\nabla I\cdot\mathbf{u}(\mathbf{x},\mathbf{a}_i(\mathbf{s}))+I_t)^2\right] + \frac{\lambda}{|\mathcal{G}(\mathbf{s})|}\left[\sum_{\mathbf{y}\in\mathcal{G}(\mathbf{s})}\sum_{\mathbf{t}\in\mathcal{N}(\mathbf{s},\mathbf{y})}\sum_{l\in\mathcal{L}(\mathbf{t})}\rho(\mathbf{u}(\mathbf{y},\mathbf{a}_i(\mathbf{s}))-\mathbf{u}(\mathbf{y},\mathbf{a}_l(\mathbf{t})),\sigma_S)\right]. \quad (27)$$

where $\mathbf{s}$ is an image region, λ controls the relative importance of the two terms, $\mathcal{R}(\mathbf{s})$ are the pixels of region $\mathbf{s}$, $\mathbf{a}_i(\mathbf{s})$ are the affine parameters of layer i in patch $\mathbf{s}$, $\mathcal{G}(\mathbf{s})$ is the set that contains the pixels at the boundaries of patch $\mathbf{s}$, $\mathcal{N}(\mathbf{s},\mathbf{y})$ are the neighboring patches connected to patch $\mathbf{s}$ at pixel $\mathbf{y}$. The two terms of E (data and spatial) are normalized with respect to the size of $\mathcal{R}(\mathbf{s})$ and $\mathcal{G}(\mathbf{s})$ 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 paper, ρ is taken to be the function given in Equation (25).

The first term of Equation (27) is simply the multi-layer bone from Section 3. The smoothness term 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. Note that there may be a different number of layers in neighboring patches; the smoothness term is computed with respect to any and all neighboring layers.

While Equation (27) 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 described in the previous section. This process alternates between solving for the $\mathbf{a}_i(\mathbf{s})$ in each layer taking into account the smoothness term and solving for the weights $m_i(\mathbf{x},\sigma_i(\mathbf{s}))$. We use a fixed rate for annealing the smoothness term; σ_S initially starts with a maximum value of 1.0 is and reduced by a factor of 0.95 at each iteration until it reaches the lower bound of 0.2. λ is taken to be 6.0. These parameters remain fixed for the experiments in this paper.

Note that we could estimate the σ_S of the smoothness term as we did for the data term in Section 3.2.3. However, we found that the fixed annealing scheme works well in practice.

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.

4.4 Experimental Results: Skin & Bones

This section presents experiments with synthetic and natural image sequences. In the current implementation, the algorithm takes approximately 2 minutes on a SGI R10000 workstation with a 194 MHz processor to compute the optical flow between two images of approximately 256×256 pixels.

Yosemite Sequence

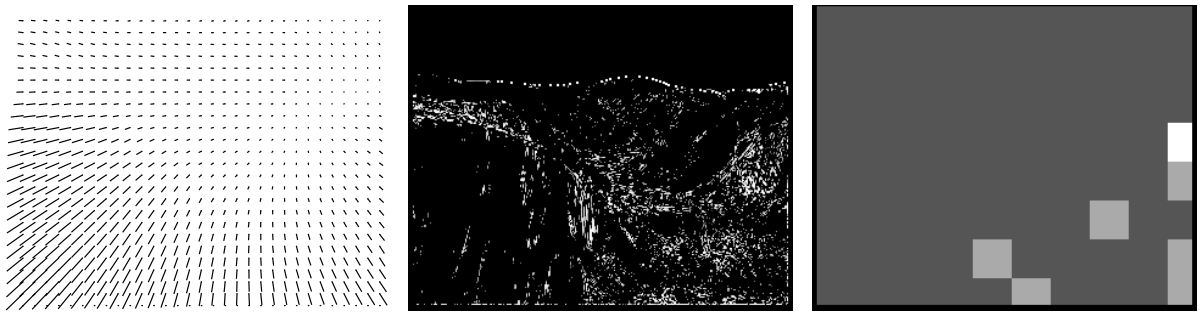
To illustrate the effect of regularization we compare results for the Yosemite sequence example with and without skin. The recovered optical flow using Equation (27) is shown in Figure 13. Comparing the results to those in Figure 9 one can see that unstable results near the boundaries are gone and that the flow appears much smoother. The addition of skin regularizes the results and this helps the algorithm choose the appropriate number of layers in most of the image patches. A single layer is chosen in most of the patches; compare this with Figure 9(c). The improvement is compared quantitatively in Table 2. Note that the addition of "skin" reduces the average angular

	Pixel Error	Average Error	Standard Deviation	Percent of flow vectors with error less than:				
				$< 1°$	$< 2°$	$< 3°$	$< 5°$	$< 10°$
Bones:	0.163	3.80°	4.84°	18.2%	40.5%	57.0%	76.5%	94.2%
Skin&Bones:	0.088	1.97°	1.70°	33.4%	62.6%	79.9%	94.8%	99.7%

Table 2: Error results for the Yosemite fly-through sequence: Skin and Bones.

error by 48%.

The results of the Skin & Bones approach are compared with other published results for the



(a) (b) (c)

Figure 13: Yosemite sequence: Skin & Bones. (a) Estimated flow; (d) Outliers; (c) Number of layers (one for dark grey, three for white).

Yosemite sequence in Table 3 [5]. In [5], when the sky is omitted, the errors for the Lucas and Kanade [32] and Fleet and Jepson [18] methods improve to $3.37°$ and $2.97°$ respectively, although the density remains low. The accuracy of the other approaches in [5] 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).

In [10], Black and Jepson perform a similar parameterized 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. If we allow similar local deformations from the Skin & Bones fit, the average angular error decreases to $1.82°$ with as standard deviation of $1.58°$ and 100% density.

Flower Garden Sequence

Figure 14 revisits the flower garden sequence with the same region size as in previous section. With the skin, the estimated optical flow is smoother at the tree boundaries and branches. This can be seen in the horizontal motion at the boundary and center of the tree in Figure 14 (c). The regions bordering the tree have two clearly distinct motions which are smoothly connected to their neighbors. Figure 14 (d) shows those points that were not accounted for by either layer and were treated as outliers. Notice that these occur predominantly around the occlusion/disocclusion boundary of the tree and the background. The horizontal image motion (Figure 14 (c)) is generated

Technique	Average Error	Standard Deviation	Density
Anandan [2]	15.84°	13.46°	100%
Singh [43]	13.16°	12.07°	100%
Nagel [37]	11.71°	10.59°	100%
Horn and Schunck (modified) [25]	11.26°	16.41°	100%
Uras *et al.* [46]	10.44°	15.00°	100%
Bab-Hadiashar and Suter [4] *	3.17°	6.46°	100%
Fleet and Jepson [18]	4.29°	11.24°	34.1%
Lucas and Kanade [32]	4.10°	9.58°	35.1%
Weber and Malik [50]	3.42°	5.35°	45.2%
Black and Anandan [9] *	4.46°	4.21°	100%
Black [8] *	3.52°	3.25°	100%
Black and Jepson [10] *	2.9°	3.2°	100%
Bones *	3.80°	4.84°	100%
Skin & Bones *	1.97°	1.70°	100%

Table 3: Comparison of various optical flow algorithms. The "*" indicates those results computed without the sky.

by taking the most likely motion at each pixel (given the weights $m_i(\mathbf{x}, \sigma_i(s))$). Comparing these results with those in Figure 7 (no skin), we can see that the motion estimated in the upper right of the image is more stable. The vector field is also shown in Figure 14 (a).

Notice that adding skin not only improves the accuracy of motion estimation, but also helps to select the right number of layers in the regions. In comparison with the Figure 7(b), the placement of multiple layers for the version with skin (Figure 14(b)) appears more appropriate.

SRI Tree Sequence

Figure 15 shows skin and bones results for the SRI tree sequence. From the horizontal motion in Figure 15 (b), we see that the branches and the background are recovered simultaneously. Examining the the number of layers selected in Figure 15 (c), we find that the ground plane is essentially treated to have two layers, due to the fact that that an affine motion model is not sufficient for recovering the planar motion of the ground. The branches of the tree and the background, however, are also assigned to different layers when they both appear in the same region.

5 Issues and Future Work

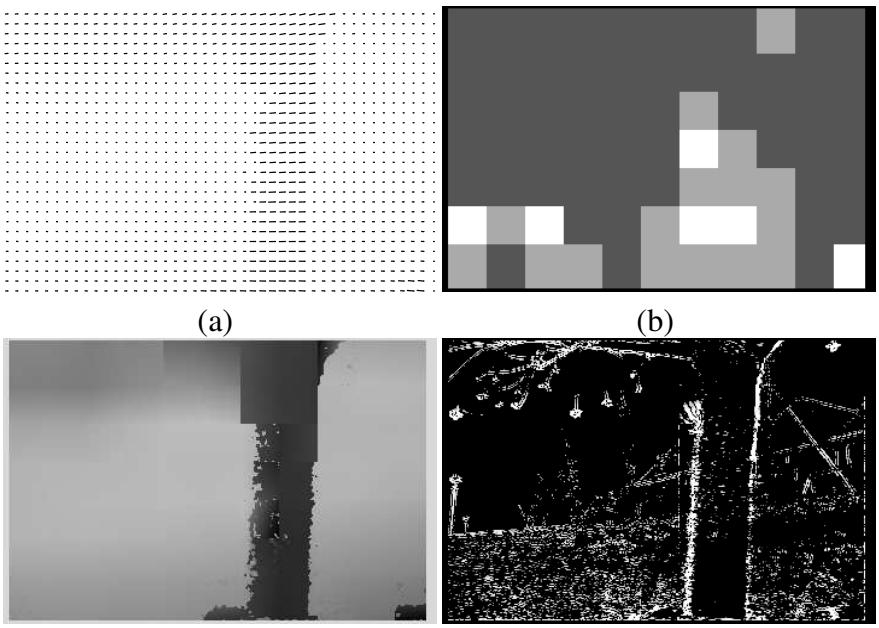


Figure 14: Flower Garden (skin&bones) (a) vector field; (b) number of layers (c) horizontal component of flow; (d) outliers.

The combination of local, layered, motion estimates with a regularizing term shows promise for optical flow estimation. There are situations, however, where the basic scheme described here may fail. Consider the experimental results depicted in Figure 16 in which a diamond shape 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, a single layer may be chosen. This is most likely to occur when the small region contains very little texture.

In our work, we have explored a simple strategy with fixed size patches placed on a regular grid. Some of the problems with this method might be reduced by simply sampling the image more finely with overlapping layers. This would be a trivial extension but would increase the computational cost of the method. More sophisticated methods might be used to adaptively select an appropriate patch size or shape [38, 44]. It might be possible to pose this problem within the MDL framework described here.

Another issue which should be explored in future work involves the addition of a spatial prior

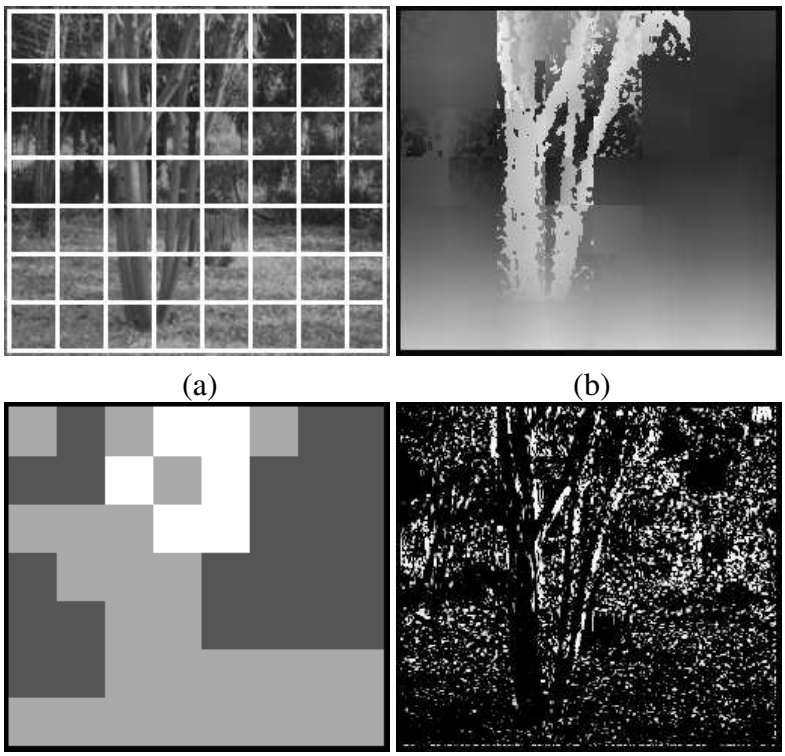


Figure 15: SRI Tree sequence (skin&bones) (a) Image with segmented regions shown; (b) Horizontal component of flow; (c) number of layers; (d) Weights for outlier layer.

on the ownership weights [52]. Notice in Figure 15 that the estimated flow is "speckled" in places. So, for example, the motion of the tree branch may fit portions of the background as well if these regions have little texture. A term to encourage spatial coherence of the weights could be added to Equation (27).

We have also left for future work the integration of information over time. One approach would be to add a temporal coherence prior [8] to the objective function. An alternative direction would be to allow the "bones" to move with the underlying image data. In this tracking formulation, the skin would dynamically "connect" bones within a spatial neighborhood.

6 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

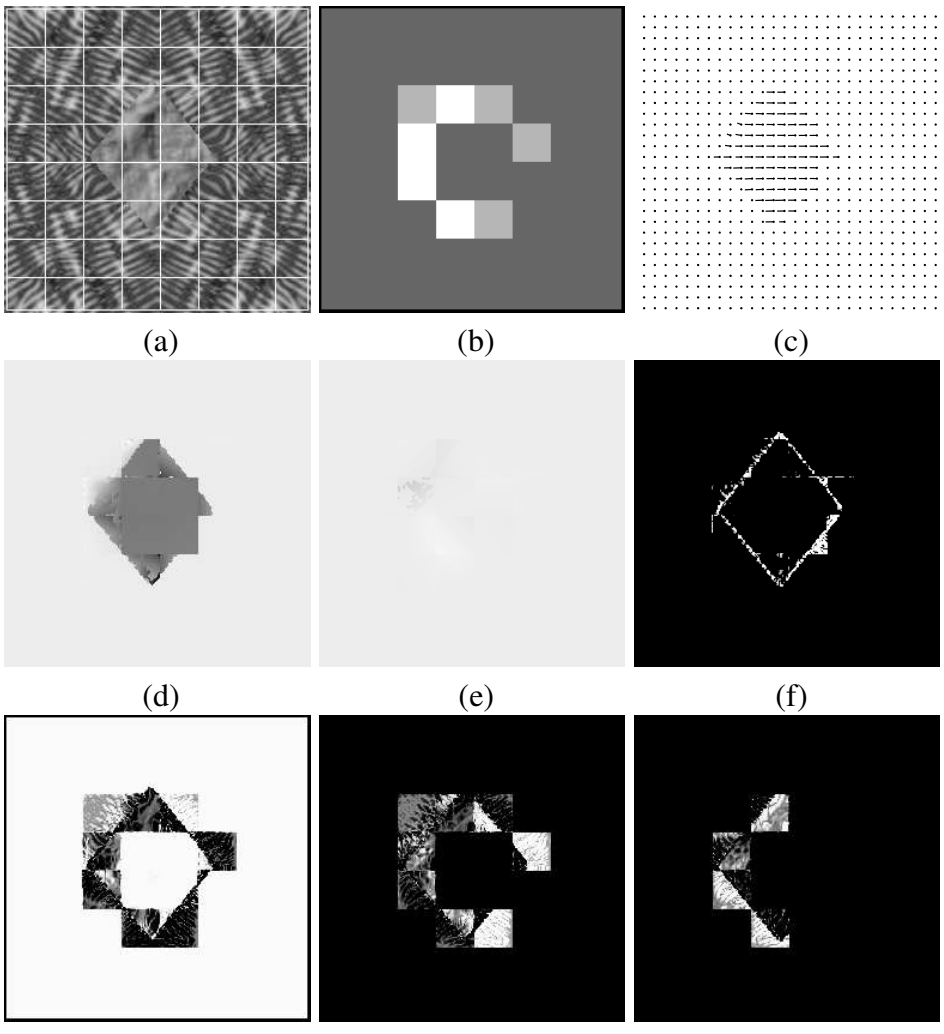


Figure 16: (a) Image with segmented regions shown; (b) number of layers; (c) vector field; (d) horizontal component of flow; (e) vertical component of flow; (f) Weights for outlier layer. (g) Weights for layer 1; (h) Weights for layer 2; (i) Weights for layer 3;

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. Moreover, we have provided a general framework for regularization with transparency that extends regularization to cope with multiple local motion estimates. The methods have been tested on synthetic and natural images and provide accurate flow estimates common to parametric approaches, while maintaining the flexibility of regularization schemes.

The results of the method are compelling and suggest a number of future directions for research. For example, 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 bones. The coarse bones provide a coarse estimate and finer bones, 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.

Acknowledgements. The authors would like to thank Eric Mjolsness for his helpful suggestions, Jim Bergen for the "Skaters" sequence, D. Heeger for the Yosemite sequence[5], and H. Baker and R. Bolles for the SRI tree sequence.

References

[1] 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.

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

[3] 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] The Yosemite sequence was generated by L. Quam

[4] A. Bab-Hadiashar and D. Suter. Optic flow calculation using robust statistics. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 988–993, Puerto Rico, June 1997.

[5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. Technical Report RPL-TR-9107, University of Western Ontario, July 1993.

[6] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *Proc. ECCV*, pages 237–252, May 1992.

[7] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. *Proc. ICCV*, pages 27–32, December 1990.

[8] 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.

[9] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[10] 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, October 1996.

[11] M. J. Black and A. Rangarajan. The outlier process: Unifying line processes and robust statistics. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 15–22, Seattle, WA, June 1994.

[12] M. J. Black and R. Rosenholtz. Robust estimation of multiple surface shapes from occluded textures. In *International Symposium on Computer Vision*, pages 485–490, Miami, FL, November 1995.

[13] M. J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. In *Proceedings of the International Conference on Computer Vision*, pages 374–381, Boston, Mass., June 1995.

[14] M. J. Black, Y. Yacoob, A. D. Jepson, and D. J. Fleet. Learning parameterized models of image motion. In *Proc. Computer Vision and Pattern Recognition, CVPR-97*, pages 561–567, Puerto Rico, June 1997.

[15] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, 1987.

[16] 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.

[17] C. L. Fennema and W. B. Thompson. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9:301–315, 1979.

[18] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *Int. J. of Computer Vision*, 5(1):77–104, 1990.

[19] D. Geiger and R. A. M. Pereira. The outlier process. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 61–69, 1991.

[20] S. Geman and D. E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, LII-4:5–21, 1987.

[21] 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.

[22] 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.

[23] J. G. Harris, C. Koch, E. Staats, and J. Luo. Analog hardware for detecting discontinuities in early vision. *Int. Journal of Comp. Vision*, 4(3):211–223, June 1990.

[24] B. K. P. Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.

[25] B. K.P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, August 1981.

[26] 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.

[27] A. Jepson and M. J. Black. Mixture models for image representation. Technical Report ARK96-PUB-54, University of Toronto, March 1996.

[28] 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.

[29] 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.

[30] 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.

[31] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3(1):73–102, 1989.

[32] 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.

[33] 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.

[34] G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc,, N.Y., 1988.

[35] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(2):220–228, March 1987.

[36] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 8(5):565–593, September 1986.

[37] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299–324, November 1987.

[38] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, 1992.

[39] M. Otte and H. H. Nagel. Optical flow estimation: Advances and comparisons. In J. Eklundh, editor, *European Conf. on Computer Vision, ECCV-94*, volume 800 of *LNCS-Series*, pages 51–60, Stockholm, Sweden, 1994. Springer-Verlag.

[40] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[41] 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.

[42] D. Shulman and J. Hervé. Regularization of discontinuous flow fields. In *Proc. Workshop on Visual Motion*, pages 81–85, Irvine, CA, March 1989. IEEE Computer Society Press.

[43] A. Singh. *Optical Flow Computation: A Unified Perspective*. IEEE Computer Society Press, Los Alamitos, California, 1991.

[44] R. Szeliski and H. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1211, 1996.

[45] R. Szeliski and D. Tonnesen. Surface modeling with oriented partical systems. *Computer Graphices*, 26(2):185–194, July 1992.

[46] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion preception. *Biological Cybernetics*, 60:79–97, 1989.

[47] J. W. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *CVPR*, pages 361–366, June 1993.

[48] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.

[49] 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.

[50] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *Proc. ICCV*, pages 12–20, May 1993.

[51] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, June 1997.

[52] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 321–326, San Francisco, June 1996.

[53] A. Yuille, T. Yang, and D. Geiger. Robust statistics, transparency and correspondence. Technical Report 90-7, Harvard Robotics Laboratory.

Biographies

Shanon X. Ju

Shanon Xuan Ju received her B.Sc. in 1992 from Tsinghua University in China, her M.S. in 1994 from University of Toronto, and is currently a Ph.D. candidate in the department of Computer Science, University of Toronto. During the summers of 1995 and 1996, she was a visiting researcher at the Xerox Palo Alto Research Center. Her research interests include motion estimation and its applications in the video databases, rigid and non-rigid object tracking, and recognition of activities.

Michael J. Black

Michael Black received his B.Sc. in 1985 from the University of British Columbia, his M.S. in 1989 from Stanford University, and his Ph.D. in 1992 from Yale University. Between 1990 and 1992, he was a visiting researcher at the NASA Ames Research Center, and from 1992 to 1993 he was an Assistant Professor in the Department of Computer Science at the University of Toronto where he is currently an Adjunct Professor. In 1993 Dr. Black joined the Xerox Palo Alto Research Center and is currently the head of the image understanding research group. In 1991 he won the IEEE Computer Society Outstanding Paper Award (CVPR'91) for his work with P. Anandan on robust motion estimation. His research interests include optical flow estimation, applications of robust statistics in computer vision, and human motion and gesture understanding.

Allan D. Jepson

Allan Jepson received the B.Sc. degree in 1976 from the University of British Columbia and the Ph.D. degree in Applied Mathematics in 1981 from the California Institute of Technology. He spent two years as a postdoctoral fellow at Stanford University in the Mathematics Department. He is now a Professor in the Department of Computer Science at the University of Toronto. He was also a member of the Canadian Institute of Artificial Intelligence during 1989-1995. His current research interests include various aspects of computer vision (see http://www.cs.toronto.edu/~jepson).