A Feature-based Approach for Dense Segmentation and Estimation of Large Disparity Motion.

Josh Wills, Sameer Agarwal and Serge Belongie

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093, USA
{josh,sagarwal,sjb} @ cs.ucsd.edu

Abstract

We present a novel framework for motion segmentation that combines the concepts of layer-based methods and feature-based motion estimation. We estimate the initial correspondences by comparing vectors of filter outputs at interest points, from which we compute candidate scene relations via random sampling of minimal subsets of correspondences. We achieve a dense, piecewise smooth assignment of pixels to motion layers using a fast approximate graphcut algorithm based on a Markov random field formulation. We demonstrate our approach on image pairs containing large inter-frame motion and partial occlusion. The approach is efficient and it successfully segments scenes with inter-frame disparities previously beyond the scope of layer-based motion segmentation methods. We also present an extension that accounts for the case of non-planar motion, in which we use our planar motion segmentation results as an initialization for a regularized Thin Plate Spline fit. In addition, we present applications of our method to automatic object removal and to structure from motion.

1 Introduction

Consider the pair of images shown in Figure 1. These two images were captured by an aquarium webcam on a pan-tilt head. For a human observer, a brief examination of the images reveals what happened from one frame to the next: the lower fish swam down and darted forward and the upper fish moved forward slightly; meanwhile, the camera panned to the left about a third of the image width. Even without color information, this is a simple task for the human visual system. The same cannot be said for any existing computer vision system. What makes this problem difficult from a computational perspective? There are number of complicating factors, including the following: (1) due to the low frame rate, the motion between frames is a significant fraction of the image size, (2) the moving objects are relatively small and have few features compared to the richly textured background, (3) the poses of the fish change as they swim, (4) because of the panning motion of the camera, the second frame has motion blur.

Finding out what went where in two frames of an image sequence is an instance of the motion segmentation problem. Formally, motion segmentation consists of (1) finding groups of pixels in two or more frames that move together, and (2) recovering the motion fields associated with each group. Motion segmentation has wide applicability in areas such as video coding, content-based video retrieval, and mosaicking. In its full generality, the problem cannot be solved since infinitely many constituent motions can explain the changes from one frame to another. Fortunately, in real scenes the problem is simplified by the observation that objects are usually composed of spatially contiguous regions and the number of independent motions is significantly smaller than the number of pixels. Operating under these assumptions, we propose a new motion segmentation algorithm for scenes containing objects with large inter-frame motion. The algorithm leverages and builds upon established techniques for robust estimation of motion fields and discontinuity preserving smoothing in a novel combination that delivers the first dense, layer-based motion segmentation method for the case of large (non-differential) motions.

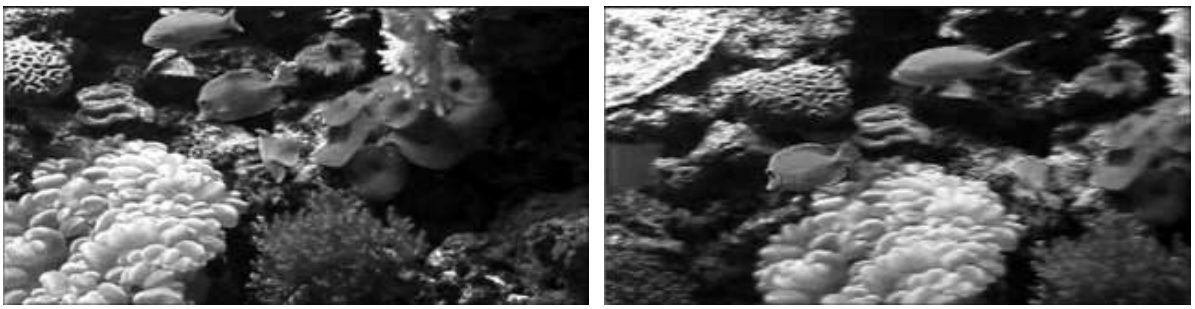

Figure 1: Two consecutive frames from a saltwater aquarium webcam.

The structure of the paper is as follows. We will begin in section 2 with an overview of related work. In section 3, we detail the components of our approach to the problem of motion segmentation. This the primary contribution of this work. We present experimental results for this approach in section 4. An extension to non-planar motion is presented in section 5. Applications of our method are presented in section 6. The paper concludes in section 8

2 Related Works

Early approaches to motion segmentation were based on estimating dense optical flow. The optical flow field was assumed to be piecewise smooth to account for discontinuities due to occlusion and object boundaries, see for example [1, 4, 29]. Darrell & Pentland [12] and Wang & Adelson [46] introduced the idea of decomposing the image sequence into multiple overlapping layers, where each layer is a smooth motion field. Weiss [47] extended this approach to account for flexible motion fields using regularized radial basis functions (RBFs).

Optical flow based methods are limited in their ability to handle large inter-frame motion or objects with overlapping motion fields. Coarse-to-fine methods are able to solve the problem of large motion to a certain extent (see for example [35, 36]), but the degree of sub-sampling required to make the motion differential places an upper bound on the maximum allowable motion between two frames and limits it to about 15% of the dimensions of the image [21]. Also in cases where the order of objects along any line in the scene is reversed and their motion fields overlap, the coarse to fine processing ends up blurring the two motions into a single motion before optical flow can be calculated.

In this paper we are interested in the case of discrete motion, i.e. where optical flow based methods break down. Most closely related to our work is that of Torr [38]. Torr uses sparse correspondences obtained by running a feature detector and matching them using normalized cross correlation. He then processes the correspondences in a RANSAC framework to sequentially cover the the set of motions in the scene. Each iteration of his algorithm finds the dominant motion model that best explains the data and is simplest according to a complexity measure. The set of models and the associated correspondences are then used as the initial guess for the estimation of a mixture model using the Expectation Maximization (EM) algorithm. Spurious models are pruned and the resulting segmentation is smoothed using morphological operations.

In a more recent work [39], the authors extend the model to 3D layers in which points in the layer have an associated disparity. This allows for scenes in which the planarity assumption is violated and/or a significant amount of parallax is present. The pixel correspondences are found using a multiscale differential optical flow algorithm, from which the layers are estimated in a Bayesian framework using EM. Piecewise smoothness is ensured by using a Markov random field prior.

Neither of the above works demonstrate the ability to perform dense motion segmentation on a pair of images with large inter-frame motion. In both of the above works the grouping is performed in a Bayesian framework. While the formulation is optimal and strong results can be proved about the optimality of the Maximum Likelihood solution, actually solving for it is an extremely hard non-linear optimization problem. The use of EM only guarantees a locally optimal solution and says nothing about the quality of the solution. As the authors point out, the key to getting a good segmentation using their algorithm is

to start with a good guess of the solution and they devote a significant amount of effort to finding such a guess. However it is not clear from their result how much the EM algorithm improves upon their initial solution.

A representative work addressing the case of non-planar motion is [40], which shows how the trifocal tensor can be used to cluster groups of sparse correspondences that move coherently across three views. That work deals with similar types of sequences as those in our work, but it does not provide dense assignment to motion layers or dense optical flow. The paper states that it is an initialization and that more work is needed to provide a dense segmentation, however the extension of dense stereo assignment to multiple independent motions is certainly non-trivial and there is yet to be a published solution. In addition, this approach is not applicable for objects with non-rigid motion, as the fundamental matrix and trifocal tensor apply only to rigid motion.

3 Proposed Method

3.1 Our Approach

Our approach is based on a two stage process, the first of which is responsible for motion field estimation and the second of which is responsible for motion layer assignment. As a preliminary step we detect interest points in the two images and match them by comparing filter responses. We then use a RANSAC based procedure for detecting the motion fields relating the frames. Based on the detected motion fields, the correspondences detected in the first stage are partitioned into groups corresponding to each constituent motion field and the resulting motion fields are re-estimated. Finally, we use a fast approximate graph cut based method to densely assign pixels to their respective motion fields. We now describe each of these steps in detail. A reference Matlab implementation of the steps described in this section is available for download at `http://vision.ucsd.edu/motion_seg.html`.

3.1.1 Interest point detection and matching

Many pixels in real images are redundant so it is beneficial to find a set of points that reduce some of this redundancy. To achieve this, we detect interest points using the Förstner operator [17]. To describe each interest point, we apply a set of 76 filters (3 scales and 12 orientations with even and odd phase and an elongation ratio of 3:1, plus 4 spot filters) to each image. The filters, which are at most 31×31 pixels in size, are evenly spaced in orientation at intervals of 15° and the changes in scale are half octave. For each of the scales and orientations, we use a quadrature pair of derivative-of-Gaussian filters corresponding to edge and bar-detectors respectively, as in [23, 18].

To obtain some degree of rotational invariance, the filter response vectors may be reordered so that the order of orientations is cyclically shifted. This is equivalent to filtering a rotated version of the image patch that is within the support of the filter. We perform three such rotations in each direction to obtain rotational invariance up to $\pm 45^\circ$.

We find correspondences by comparing filter response vectors using the L_1 distance. We compare each interest point in the first image to those in the second image and assign correspondence between points with minimal error. Since matching is difficult for image pairs with large inter-frame disparity, the remainder of our approach must take into account that the estimated correspondences can be extremely noisy.

3.1.2 Estimating Motion Fields

Robust estimation methods such as RANSAC [16] have been shown to provide very good results in the presence of noise when estimating a single, global transformation between images. Why can't we simply apply these methods to multiple motions directly? It turns out that this is not as straightforward as one might imagine. Methods in this vein work by iteratively repeating the estimation process where each time a dominant motion is detected, all correspondences that are deemed inliers for this motion are removed [38].

There are a number of issues that need to be addressed before RANSAC can be used for the purpose of detecting and estimating multiple motions. The

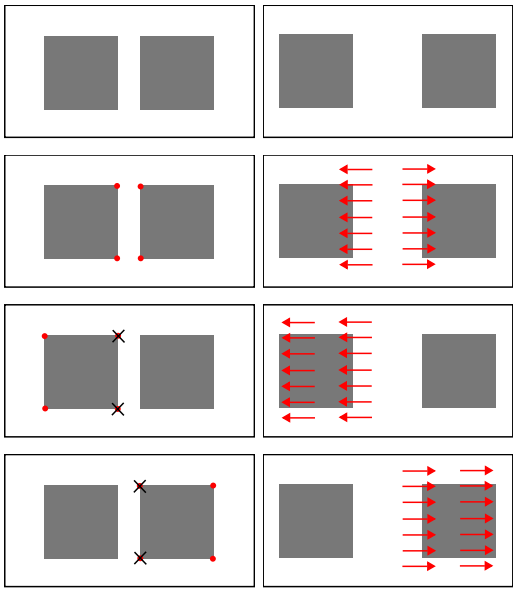

Figure 2: Phantom motion fields. (Row 1) Scene that consists of two squares translating away from each other. (Row 2) Under an affine model, triplets of points that span the two squares will incorrectly propose a global stretching motion. This motion is likely to have many inliers since all points on the inner edges of the squares will fit this motion exactly. If we then delete all points that agree with this transformation, we will be unable to detect the true motions of the squares in the scene (Rows 3 & 4).

first issue is that combinations of correspondences – not individual correspondences – are what promote a given transformation. Thus when "phantom motion fields" are present, i.e., transformations arising from the relative motion between two or more objects, it is possible that the deletion of correspondences could prevent the detection of the true independent motions; see Figure 2. Our approach does not perform sequential deletion of correspondences and thus circumvents this problem.

Another consideration arises from the fact that the RANSAC estimation procedure is based on correspondences between interest points in the two images. This makes the procedure biased towards texture rich regions, which have a large number of interest points associated with them, and against small objects in the scene, which in turn have a small number of interest points. In the case where there is only one global transformation relating the two images, this bias does not pose a problem. However it becomes apparent when searching for multiple independent motions. To correct for this bias we introduce "perturbed interest points" and a method for feature crowdedness compensation.

Perturbed Interest Points If an object is only represented by a small number of interest points, it is unlikely that many samples will fall entirely within the object. One approach for promoting the effect of correct correspondences without promoting that of the incorrect correspondences is to appeal to the idea of a stable system. According to the principle of perturbation, a stable system will remain at or near equilibrium even as it is slightly modified. The same holds true for stable matches. To take advantage of this principle, we dilate the interest points to be disks with a radius of r_p, where each pixel in the disk is added to the list of interest points. This allows the correct matches to get support from the points surrounding a given feature while incorrect matches will tend to have almost random matches estimated for their immediate neighbors, which will not likely contribute to a widely-supported warp. In this way, while the density around a valid motion is increased, we do not see the same increase in the case of an invalid motion; see Figure 3.

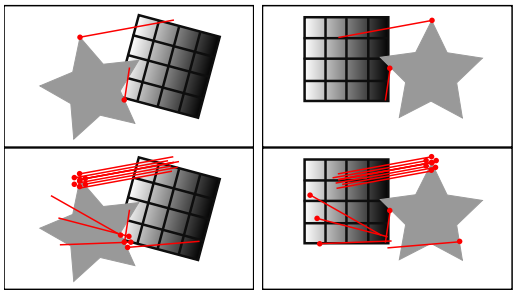

Figure 3: Perturbed Interest Points. Correspondences are represented by point-line pairs where the point specifies an interest point in the image and the line segment ends at the location of the corresponding point in the other image. (Row 1) We see one correct correspondence and one incorrect correspondence that is the result of an occlusion junction forming a white wedge. (Row 2) The points around the correct point have matches that are near the corresponding point, but the points around the incorrect correspondence do not.

Feature Crowdedness Textured regions often have significant representation in the set of interest points. This means that a highly textured object will have a much larger representation in the set of interest points than an object of the same size with less texture. To mitigate this effect, we bias the sampling.

We calculate a measure of crowdedness for each interest point and the probability of choosing a given point is inversely proportional to this crowdedness score. The crowdedness score is the number of interest points that fall into a disk of radius r_c.

Partitioning and Motion Estimation Having perturbed the interest points and established a sampling distribution on them, we are now in a position to detect the motions present in the frames. We do so using a two step variant of RANSAC, where multiple independent motions are explicitly handled, as duplicate transformations are detected and pruned in a greedy manner. The first step provides a rough partitioning of the set of correspondences (motion identification) and the second takes this partitioning and estimates the motion of each group (motion refinement).

First, a set of planar warps is estimated by a round of standard RANSAC and inlier counts (using an inlier threshold of τ) are recorded for each transformation. In our case, we use planar homography which requires 4 correspondences to estimate, however similarity or affinity may be used (requiring 2 and 3 correspondences, respectively). The estimated list of transformations is then sorted by inlier count and we keep the first n_t transformations, where n_t is some large number (e.g. 300).

We expect that the motions in the scene will likely be detected multiple times and we would like to detect these duplicate transformations. Comparing transformations in the space of parameters is difficult for all but the simplest of transformations, so we compare transformations by comparing the set of inliers associated with each transformation. If there is a large overlap in the set of inliers (more than 75%) the transformation with the larger set of inliers is kept and the other is pruned.

Now that we have our partitioning of the set of correspondences, we would like to estimate the planar motion represented in each group. This is done with a second round of RANSAC on each group with only 100 iterations. This round has a tighter threshold to find a better estimate. We then prune duplicate warps a second time to account for slightly different inlier sets that converged to the same transformation during the second round of RANSAC with the tighter threshold.

The result of this stage is a set of proposed transformations and we are now faced with the problem of assigning each pixel to a candidate motion field.

3.1.3 Layer Assignment

The problem of assigning each pixel to a candidate motion field can be formulated as finding a function $l : I \rightarrow \{1, \ldots, m\}$, that maps each pixel to an integer in the range $1, \ldots, m$, where m is the total number of of motion fields, such that the reconstruction error

$$\sum_i [I(i) - I'(M(l(i), i))]^2$$

is minimized. Here $M(p, q)$ returns the position of pixel q under the influence of the motion field p.

A naïve approach to solving this problem is to use a greedy algorithm that assigns each pixel the motion field for which it has the least reconstruction error, i.e.,

$$l(i) = \underset{1 \leq p \leq m}{\operatorname{argmin}} [I(i) - I'(M(p, i))]^2 \qquad (1)$$

The biggest disadvantage of this method as can be seen in Figure 4 is that for flat regions it can produce unstable labellings, in that neighboring pixels that have the same brightness and are part of the same moving object can get assigned to different warps. What we would like instead is to have a labelling that is piecewise constant with the occasional discontinuity to account for genuine changes in motion fields.

Figure 4: Example of naïve pixel assignment as in Equation 1 for the second motion layer in Figure 6. Notice there are many pixels that are erratically assigned. This is why smoothing is needed.

The most common way this problem is solved (see e.g. [47]) is by imposing a smoothness prior over the set of solutions, i.e., an ordering that prefers piecewise constant labellings over highly unstable ones. It

is important that the prior be sensitive to true discontinuites present in the image. In [6], for example, Boykov, Veksler and Zabih have shown that discontinuity preserving smoothing can be performed by adding a penalty of the following form to the objective function

$$\sum_i \sum_{j \in \mathcal{N}(i)} s_{ij}[1 - \delta_{l(i)l(j)}]$$

where $\delta_{..}$ is the Kronecker delta, equal to 1 when its arguments are equal. Given a measure of similarity s_{ij} between pixels i and j, it penalizes pixel pairs that have been assigned different labels. The penalty should only be applicable for pixels that are near each other. Hence the second sum is over a fixed neighborhood $\mathcal{N}(i)$. The final objective function we minimize is

$$\sum_i [I(i) - I'(M(l(i), i))]^2 + \lambda \sum_i \sum_{j \in \mathcal{N}(i)} s_{ij}[1 - \delta_{l(i)l(j)}]$$

where λ is the tradeoff between the data and the smoothness prior.

An optimization problem of this form is known as a *Generalized Potts model* which in turn is special case of a class of problems known as metric labelling problems. Kleinberg & Tardos demonstrate that the metric labelling problems corresponds to finding the maximum *a posteriori* labelling of a class of Markov random field [24]. The problem is known to be NP-complete, and the best one can hope for in polynomial time is an approximation.

Recently Boykov, Veksler and Zabih (BVZ) have developed a polynomial time algorithm that finds a solution with error at most two times that of the optimal solution [7]. Each iteration of the algorithm constructs a graph and finds a new labelling of the pixels corresponding to the minimum cut partition in the graph. The algorithm is deterministic and guaranteed to terminate in $O(m)$ iterations.

Besides the motion fields and the image pair, the algorithm takes as input a similarity measure s_{ij} between every pair of pixels i, j within a fixed distance of one another and two parameters, k the size of the neighborhood around each pixel, and λ the tradeoff between the data and the smoothness term. We use a Gaussian weighted measure of the squared difference between the intensities of pixels i and j,

$$s_{ij} = \exp\left[-\frac{d(i,j)^2}{2k^2} - (I(i) - I(j))^2\right]$$

where $d(i, j)$ is the distance between pixel i and pixel j.

We run the BVZ algorithm twice, once to assign the pixels in the image I to the forward motion field and again to assign the pixels in image I' to the inverse motion fields relating I' and I. If a point in the scene occurs in both frames, we expect that its position and appearance will be related as:

$$\begin{aligned} M(l(p), p) &= p' \\ M(l'(p'), p') &= p \\ I(M(l(p), p)) &= I'(p) \end{aligned}$$

Here, the unprimed symbols refer to image I and the primed symbols refer to image I'. Assuming that the appearance of the object remains the same across the images, the final assignment is obtained by intersecting the forward and backward assignments.

In this simple intersection step, occluded pixels are removed from further consideration. By reasoning about occlusion ordering constraints over more than than two frames, one can retain and explicitly label occluded pixels in the output segmentation; see for example the recent work of Xiao and Shah [51].

```
1.  Detect interest points in I
2.  Perturb each interest point
3.  Find the matching points in I'
4.  For i = 1:N_s
        Pick tuples of correspondences
        Estimate the warp
        Store inlier count
5.  Prune the list of warps
6.  Refine each warp using its inliers
7.  Perform dense pixel assignment
```

Figure 5: Algorithm Summary

4 Experimental Results

We now illustrate our algorithm, which is summarized in Figure 5, on several pairs of images containing objects undergoing independent motions. We performed all of the experiments on grayscale images with the same parameters[1].

Our first example is shown in Figure 6. In this figure we show the two images, I and I', and the assignments for each pixel to a motion layer (one of the three detected motion fields). The rows represent the

[1] $N_s = 10^4, n_t = 300, r_p = 2, r_c = 25, \tau = 10, k = 2, \lambda = .285$. Image brightnesses are in the range $[0, 1]$.

Figure 6: Notting Hill sequence. (Row 1) Original image pair of size 311×552, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

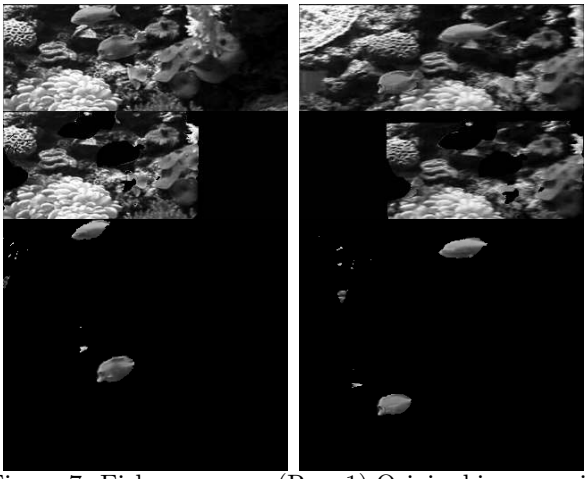

Figure 7: Fish sequence. (Row 1) Original image pair of size 162×319, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

different motion fields and the columns represent the portions of each image that are assigned to a given motion layer. The motions are made explicit in that the pixel support from frame to frame is related exactly by a planar homography. Notice that the portions of the background and the dumpsters that were visible in both frames were segmented correctly, as was the man. This example shows that in the presence of occlusion and when visual correspondence is difficult (i.e. matching the dumpsters correctly), our method provides good segmentation. Another thing to note is that the motion of the man is only approximately planar.

Figure 7 shows a scene consisting of two fish swimming past a fairly complicated reef scene. The segmentation is shown as in Figure 6 and we see that three motions were detected, one for the background and one for each of the two fish. In this scene, the fish are small, feature-impoverished objects in front of a large feature-rich background, thus making the identification of the motion of the fish difficult. In fact, when this example was run without using the perturbed interest points, we were unable to recover the motion of either of the fish.

Figure 8: Flower Garden sequence. (Row 1) Original image pair of size 240×360, (Rows 2-4) Pixels assigned to warp layers 1-3 in I and I'.

Figure 8 shows two frames from a sequence that has been a benchmark for motion segmentation approaches for some time. Previously, only optical flow-based techniques were able to get good motion segmentation results for this scene, however producing a segmentation of the motion between the two frames shown (1 and 30) would require using all (or at least most) of the intermediate frames. Here the only input to the system was the frame pair shown in Row 1. Notice that the portions of the house and the garden that were visible in both frames were segmented accurately as was the tree. This example shows the discriminative power of our filterbank as we were unable to detect the motion field correctly using correspondences found using the standard technique of normalized cross correlation. In addition, this example demonstrates the importance of the perturbed interest points and the sampling based on feature crowdedness as the correct motions were not detected when either of the two techniques were not used.

In Figure 9, a moving car passes behind a tree as the camera pans. Here, only two motion layers were recovered and they correspond to the static background and to the car. Since a camera rotating around its optical center produces no parallax for a static scene, the tree is in the same motion layer as the fence in the background, whereas the motion of the car requires its own layer. The slight rotation in depth of the car does not present a problem here.

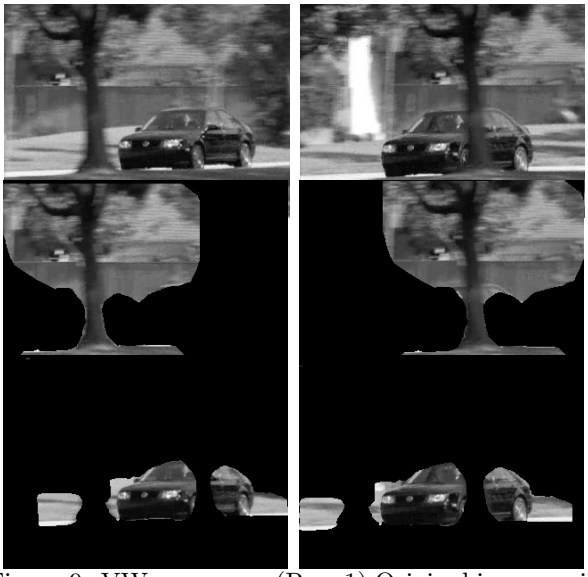

Figure 9: VW sequence. (Row 1) Original image pair of size 240×320, (Rows 2-3) Pixels assigned to warp layers 1-2 in I and I'.

5 Extension to Non-planar motion

Consider the image pairs illustrated in Figure 10. These have a significant component of planar motion but exhibit residual with respect to a planar fit because of either the non-planarity of the object (e.g. a cube) or the non-rigidity of the motion (e.g. a lizard). These are scenes for which the motion can be approximately described by a planar layer-based framework, i.e. scenes that have "shallow structure" [31]. In order to extend our approach to such scenes, we propose an additional stage consisting of a regularized spline model for capturing finer scale variations on top of an approximate planar fit. Our approach is related in spirit to the *deformotion* concept in [34], developed for the case of differential motion, which separates overall motion (a finite dimensional group action) from the more general deformation (a diffeomorphism).

It is important to remember that optical flow does not model the 3D motion of objects, but rather the changes in the image that result from this motion. Without the assumption of a rigid object, it is very difficult to estimate the 3D structure and motion of an object from observed change in the image, though there is recent work that attempts to do this [8, 9, 41, 42, 43, 44, 50]. For this reason, we choose to do all estimation in the image plane (i.e. we use 2D models), but we show that if the object is assumed to be rigid, the correspondences estimated can be used to recover the dense structure and 3D motion.

5.1 Our Approach for Non-planar Motion

When the scene contains objects undergoing significant 3D motion or deformation, the optical flow cannot be described by any single low dimensional image plane transformation (e.g., affine or homography). However, to keep the problem tractable we need a compact representation of these transformations; we propose the use of thin plate splines for this purpose. A single spline is not sufficient for representing multiple independent motions, especially when the motion vectors intersect [47]. Therefore we represent the optical flow between two frames as a set of disjoint splines. By disjoint we mean that the support of the splines are disjoint subsets of the image plane. The task of fitting a mixture of splines naturally decomposes into two subtasks: motion segmentation and spline fitting.

Ideally we would like to do both of these tasks simultaneously, however these tasks have conflicting goals. The task of motion segmentation requires us to identify groups of pixels whose motion can be described by a smooth transformation. Smoothness implies that each motion segment has the the same gross motion, however, except for the rare case in which the entire layer has exactly the same motion everywhere, there will be local variations. Hence the motion segmentation algorithm should be sensitive to inter-layer motion and insensitive to intra-layer variations. On the other hand, fitting a spline to each motion field requires attention to all the local variations. This is an example of different tradeoffs between bias and variance in the two stages of the algorithm. In the first stage we would like to exert a high bias and use models with a high amount of stiffness and insensitivity to local variations, whereas in the second stage

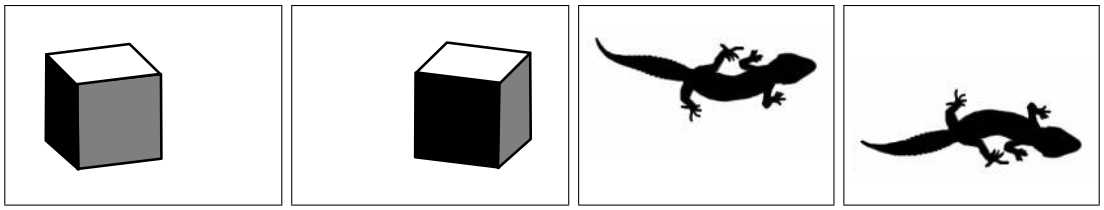

Figure 10: Non-planarity vs. non-rigidity: The left image pair shows a non-planar object undergoing 3D rigid motion; the right pair shows an approximately planar object undergoing non-rigid motion. Both examples result in residual with respect to a 2D planar fit.

we would like to use a more flexible model with a low bias.

We begin with the motion segmentation procedure of Section 3. The output of this stage, while sufficient to achieve a good segmentation, is not sufficient to recover the optical flow accurately. However, it serves two important purposes: firstly it provides an approximate segmentation of the sparse correspondences that allows for coherent groups to be processed separately. This is crucial for the second stage of the algorithm as a flexible model will likely find an unwieldy compromise between distinct moving groups as well as outliers. Secondly, since the assignment is dense, it is possible to find matches for points that were initially mismatched by limiting the correspondence search space to points in the same motion layer. The second stage then bootstraps off of these estimates of motion and layer support to iteratively fit a thin plate spline to account for non-planarity or non-rigidity in the motion. Figure 11 illustrates this process.

However, since splines form a family of universal approximators over $\mathbb{R}^2$ and can represent any 2D transformation to any desired degree of accuracy, it raises the question as to why one needs to use two different motion models in the two stages of the algorithm. If one were to use the affine transform as the dominant motion model, splines with an infinite or very large degree of regularization can indeed be used in its place. However, in the case where the dominant planar motion is not captured by an affine transform and we need to use a homography, it is not practical to use a spline. This is so because the set of homographies over any connected region of $\mathbb{R}^2$ are unbounded, and can in principal require a spline with an unbounded number of knots to represent an arbitrary homography. So while a homography can be estimated using a set of four correspondences, the corresponding spline approximation can, in principle, require an arbitrarily large number of control points. This poses a serious problem for robust estimation procedures like RANSAC since the probability of hitting the correct model decreases exponentially with increasing degrees of freedom.

Many previous approaches for capturing long range motion are based on the fundamental matrix. However, since the fundamental matrix maps points to lines, translations in a single direction with varying velocity and sign are completely indistinguishable, as pointed out, e.g. by [40]. This type of motion is observed frequently in motion sequences. The trifocal tensor does not have this problem; however, like the fundamental matrix, it is only applicable for scenes with rigid motion and there is not yet a published solution for dense stereo correspondence in the presence of multiple motions.

We now describe the refinement stage of the algorithm in detail.

5.1.1 Refining the Fit with a Flexible Model

The flexible fit is an iterative process using regularized radial basis functions, in this case Thin Plate Spline (TPS). The spline interpolates the correspondences to result in a dense optical flow field. This process is run on a per-motion layer basis.

Feature extraction and matching During the planar motion estimation stage, only a gross estimate of the motion is required so a sparse set of feature points will suffice. In the final fit however, we would like to use as many correspondences as possible to ensure a good fit. In addition, since the correspondence search space is reduced (i.e. matches are only considered between pixels assigned to corresponding motion layers), matching becomes somewhat simpler. For this reason, we use the Canny edge detector to find the set of edge points in each of the frames and

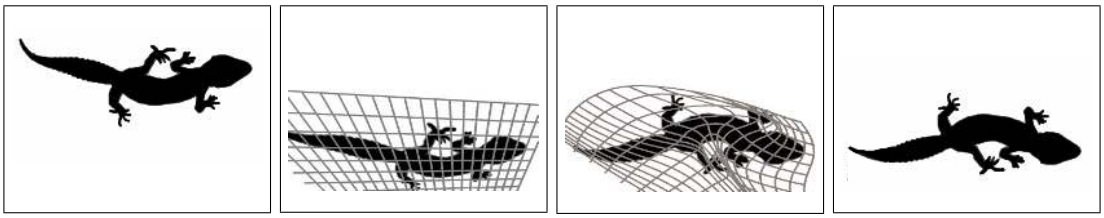

Figure 11: Determining Long Range Optical Flow. The goal is to provide dense optical flow from the first frame (1), to the second (4). This is done via a planar fit (2) followed by a flexible fit (3).

estimate correspondences in the same manner as in Section 3.

Iterated TPS fitting Given the approximate planar homography and the set of correspondences between edge pixels, we would like to find the dense set of correspondences. If all of the correspondences were correct, we could jump straight to a smoothed spline fit to obtain dense (interpolated) correspondences for the whole region. However, we must account for the fact that many of the correspondences are incorrect. As such, the purpose of the iterative matching is essentially to distinguish inliers from outliers, that is, we would like to identify sets of points that exhibit coherence in their correspondences.

One of the assumptions that we make about the scenes we wish to consider is that the motion of the scene can be approximated by a set of planar layers. Therefore a good initial set of inliers are those correspondences that are roughly approximated by the estimated homography. From this set, we use TPS regression with increasingly tighter inlier thresholds to identify the final set of inliers, for which a final fit is used to interpolate the dense optical flow. We now briefly describe this process.

Thin Plate splines are a family of approximating splines defined over $\mathbb{R}^d$. The theory for Thin Plate Splines was first developed by Duchon [14, 15] and subsequently by Meinguet [27]. Our presentation here follows follows Wahba [45].

Our task here is to construct smoothly varying functions that map pixel positions in one image to pixel positions in another. We use two splines, one each for the x and y mappings. Let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the positions of the points in the first image. Let the target for the first spline be given by v_i and let f denote the transformation that we are trying to estimate. In two dimensions the smoothness penalty for Thin Plate Splines is given by

$$J_2 = \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

J_2 is also known as the bending energy. The minimization is performed over the space of functions χ whose partial derivatives of total order 2 are in $\mathcal{L}(R^2)$, i.e., the integral of square of every partial derivative of order 2 over $\mathbb{R}^2$ is bounded. Meinguet [27] provides a detailed description of this space. The functional $J_2(f)$ defines a semi-norm over χ.

The smoothing thin-plate spline is then defined to be the solution to the following variational problem:

$$\arg\min_{f \in \chi} \left[\frac{1}{n} \sum_i^n (v_i - f(x_i, y_i))^2 + \mu \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy \right] \quad (2)$$

where the scalar μ is the tradeoff between fitting the target values v_i and the smoothness of the function f.

The null space of the penalty functional is a three dimensional space consisting of polynomials of degree less than or equal to one, i.e., the space of all functions spanned by the basis functions

$$\phi_1(x, y) = 1, \qquad \phi_2(x, y) = x, \qquad \phi_3(x, y) = y$$

$$ax + by + c, \qquad a, b, c \in \mathbb{R}$$

Duchon [14] showed that if the points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ are such that the least squares regression on ϕ_1, ϕ_2, ϕ_3 is unique then the variational problem above has a unique solution f_μ and is given by

$$f_\mu(x, y) = \sum_i^n w_i G_i(x, y) + ax + by + c \quad (3)$$

Here $G(x, y)$ is the Green's function to the twice iterated Laplacian. It is also known as the fundamental solution to the bi-harmonic equation

$$\Delta^2 G = 0$$

where,

$$G_i(r) = r^2 \log r, \qquad r^2 = (x - x_i)^2 + (y - y_i)^2$$

Thus the calculation of the spline fit requires the estimation of the parameters w_i and a, b, c.

Now let K be an $n \times n$ matrix with entries given by

$$K_{ij} = G_i(x_j, y_j)$$

and let T be a $n \times 3$ matrix with rows given by

$$T_i = [\ x_i \quad y_i \quad 1\ ].$$

Also let $\mathbf{d}$ be the 3-vector

$$\mathbf{d} = [\ a \quad b \quad c\ ]^\top.$$

Then it can be shown that the optimal value for the coefficient vector $\mathbf{w} = [w_i]$ is given by the solution to the matrix equations [45, 30, 5]

$$(K + n\mu I)\,\mathbf{w} + T\mathbf{d} = 0 \tag{4}$$

$$T^\top \mathbf{w} = 0 \tag{5}$$

This is a simple linear system that can be solved using matrix inversion. For the case of $\mu = 0$ we obtain the interpolating Thin Plate Spine.

The complexity of matrix inversion scales as $O(n^3)$ in the number of rows. Thus as the number of points that we are fitting to goes up it is not practical to use these methods on the full dataset. In our experiments we take a naive subsampling based approach. Out of the 1200 points that we are required to fit to, we randomly subsampled 500 points and used them as the landmarks for the spline fitting procedure. We observe that this simple approach works well in practice. A number of researchers have explored more sophisticated and computationally attractive approaches to the problem [19, 33, 47, 13]. Any of these can be used as a replacement for our spline estimation procedure.

We estimate the TPS mapping from the points in the first frame to those in the second where μ_t is the regularization factor for iteration t. The fit is estimated using the set of correspondences that are deemed inliers for the current transformation, where τ_t is the threshold for the t^{th} iteration. After the transformation is estimated, it is applied to the entire edge set and the set of correspondences is again processed for inliers, using the new locations of the points for error computation. This means that some correspondences that were outliers before may be pulled into the set of inliers and vice versa. The iteration continues on this new set of inliers where $\tau_{t+1} \leq \tau_t$ and $\mu_{t+1} \leq \mu_t$. We have found that three iterations of this TPS regression with incrementally decreasing regularization and corresponding outlier thresholds suffices for a large set of real world examples. Additional iterations produced no change in the estimated set of inlier correspondences.

This simultaneous tightening of the pruning threshold and annealing of the regularization factor aid in differentiating between residual due to localization error or mismatching and residual due to the non-planarity of the object in motion. When the pruning threshold is loose, it is likely that there will be some incorrect correspondences that will pass the threshold. This means that the spline should be stiff enough to avoid the adverse effect of these mismatches. However, as the mapping converges we place higher confidence in the set of correspondences passing the tighter thresholds. This process is similar in spirit to iterative deformable shape matching methods [2, 10].

I. **Estimate planar motion**
1. Find correspondences between I and I'
2. Robustly estimate the motion fields
3. Densely assign pixels to motion layers

II. **Refine the fit with a flexible model**
4. Match edge pixels between I and I'
5. For t=1:3
6. Fit all correspondences within τ_t using TPS regularized by μ_t
7. Apply TPS to set of correspondences

Note: $(\tau_{t+1} \leq \tau_t, \mu_{t+1} \leq \mu_t)$

Figure 12: Algorithm Summary

5.2 Experimental Results

We now illustrate our algorithm, which is summarized in Figure 12, on several pairs of images containing objects undergoing significant, non-planar motion. Since the motion is large, displaying the optical flow as a vector field will result in a very confusing figure. Because of this, we show the quality of the op-

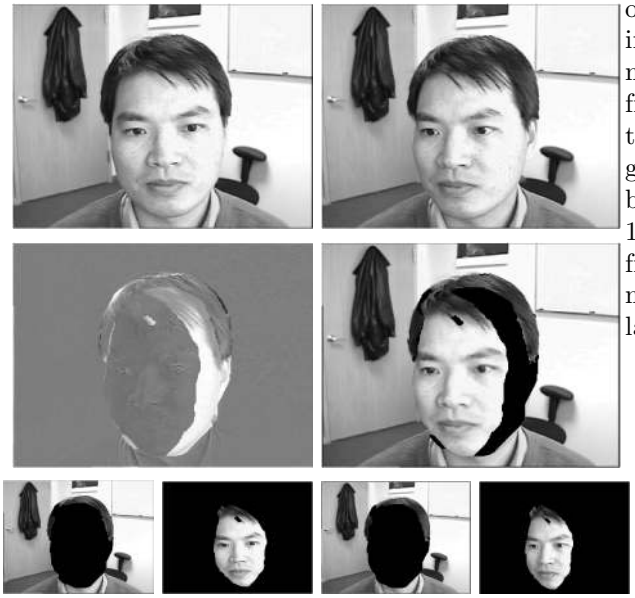

Figure 13: Face Sequence. (Row 1) The two input images, I and I' of size 240×320. (Row 2) The difference image is show first where grey regions indicate zero error regions and the reconstruction, $\mathcal{T}(I)$ is second. (Row 3) The initial segmentation found via planar motion.

tical flow in other ways, including (1) examining the image and corresponding reconstruction error that result from the application of the estimated transform to the original image (we refer to this transformed image as $\mathcal{T}(I)$), (2) showing intermediate views (as in [32]), or by (3) showing the 3D reconstruction induced by the set of dense correspondences. Examples are presented that exhibit either non-planarity, non-rigidity or a combination of the two. We show that our algorithm is capable of providing optical flow for pairs of images that are beyond the scope of existing algorithms. We performed all of the experiments on grayscale images using the same parameters[2].

Face Sequence The first example is shown in Figures 13 and 14. The top row of Figure 13 shows the two input frames, I and I', in which a man moves his head to the left in front of a static scene (the nose moves more than 10% of the image width). The second row shows first the difference image between $\mathcal{T}(I)$ and I' where error values are on the interval [-1,1] and gray regions indicate areas of zero error. This image is followed by $\mathcal{T}(I)$; this image has two estimated transformations, one for the face and another for the background. Notice that error in the overlap of the faces is very small, which means that according to reconstruction error, the estimated transformation successfully fits the relation between the two frames. This transformation is non-trivial as seen in the change in the nose and lips as well as a shift in gaze seen in the eyes, however all of this is captured by the estimated optical flow. The final row in Figure 13 shows the segmentation and planar approximation from Section 3, where the planar transformation is made explicit as the regions' pixel supports are related exactly by a planar homography. Dense correspondences allow for the estimation of intermediate views via interpolation as in [32]. Figure 14 shows the two original views of the segment associated with the face as well as a synthesized intermediate view that is realistic in appearance. The second row of this figure shows an estimation of relative depth that comes from the disparity along the rectified horizontal axis. Notice the shape of the nose and lips as well as the relation of the eyes to the nose and forehead. It is important to remember that no information specific to human faces was provided to the algorithm for this optical flow estimation.

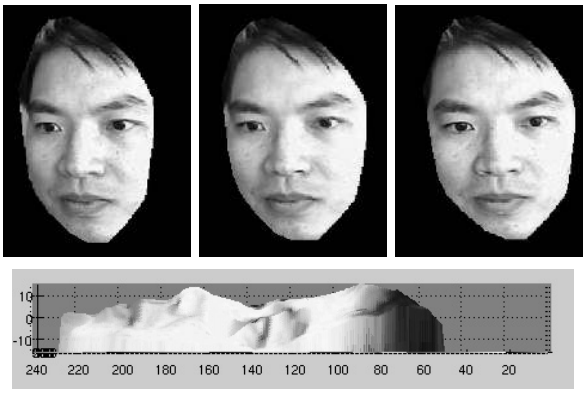

Figure 14: Face Sequence – Interpolated views. (Row 1) Original frame I', synthesized intermediate frame, original frame I, (Row 2) A surface approximation from computed dense correspondences.

Notting Hill Sequence The next example shows how the spline can also refine what is already a close approximation via planar models. Figure 15 shows a close up of the planar error image, the reconstruction error and finally the warped grid for the scene that was shown in Figure 6. The planar approximation was not able to capture the 3D nature of the clothing and the non-rigid motion of the head with respect to the torso, however the spline fit captures these things accurately.

[2] $k = 2, \lambda = .285, \tau_p = 15, \mu_1 = 50, \mu_2 = 20, \mu_3 = 1, \tau_1 = 15, \tau_2 = 10, \tau_3 = 5$. Here, k, λ, and τ_p refer to parameters in Section 3.

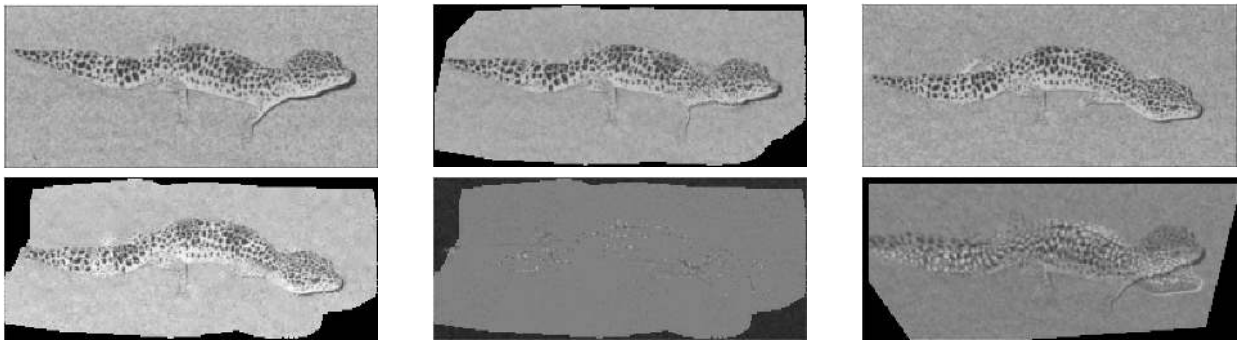

Figure 16: Gecko Sequence. (Row 1) Original frame I of size 102×236, synthesized intermediate view, original frame I'. (Row 2) $\mathcal{T}(I)$, Difference image between the above image and I' (gray is zero error), Difference image for the planar fit.

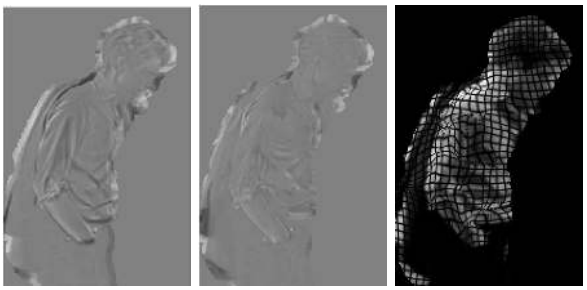

Figure 15: Notting Hill. Detail of the spline fit for a layer from Figure 6, difference image for the planar fit, difference image for the spline fit, grid transformation.

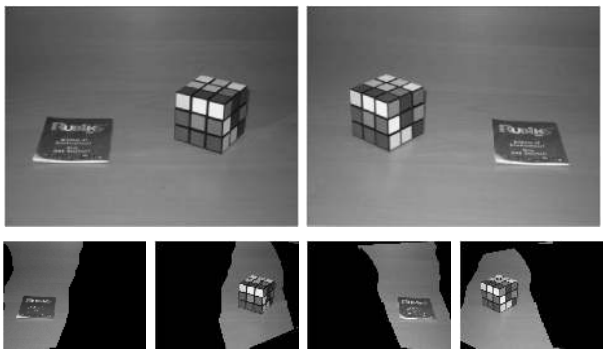

Figure 17: Rubik's Cube. (Row 1) Original image pair of size 300×400, (Row 2) assignments of each image to layers 1 and 2.

Gecko Sequence The second example, shown in Figure 16, displays a combination of a non-planar object (a gecko lizard), undergoing non-rigid motion. While this is a single object sequence, it shows the flexibility of our method to handle complicated motions. In Figure 16(1), the two original frames are shown as well as a synthesized intermediate view (here, intermediate refers to time rather than viewing direction since we are dealing with non-rigid motion) . The synthesized image is a reasonable guess at what the scene would look like midway between the two input frames. Figure 16(2) shows $\mathcal{T}(I)$ as well as the reconstruction error for the spline fit $(\mathcal{T}(I) - I')$, and the error incurred with the planar fit. We see in the second row of Figure 16(2) that the tail, back and head of the gecko are aligned very well and those areas have negligible error. When we compare the reconstruction error to the error induced by a planar fit, we see that the motion of the gecko is not well approximated by a rigid plane. Here, there is also some 3D motion present in that the head of the lizard changes in both direction and elevation. This is captured by the estimated optical flow.

Rubik's Cube The next example shows a scene with rigid motion of a non-planar object. Figure 17 displays a Rubik's cube and user's manual switching places as the cube rotates in 3D. Below the frames, we see the segmentation that is a result of the planar approximation. As can be seen the segmentation contains large chunks of the background along with the Rubik's Cube. While it is indeed desirable that the only pixels that we segment are those belonging to the Rubik's Cube, we must note that the background lacks any distinguishing features making its motion truly ambiguous. Hence without additional knowledge about the objects in the scene, any prior that we place on the scene while segmenting it will be the cause of some mistakes. In our MRF-based segmentation scheme we make the assumption that the layers are spatially contiguous, this coupled with the motion ambiguity mentioned earlier results in some portion of the background being interpreted as belonging to the same layer as the Rubik's cube. Figure 18 shows $\mathcal{T}(I)$, the result of the spline fit applied to

this same scene. The first row shows a detail of the two original views of the Rubik's cube as well as a synthesized intermediate view. Notice that the rotation in 3D is accurately captured and demonstrated in this intermediate view. The second row shows the reconstruction errors, first for the planar fit and then for the spline fit, followed by $\mathcal{T}(I)$. Notice how accurate the correspondence is since the spline applied to the first image is almost identical to the second frame.

Correspondences between portions of two frames that are assumed to be projections of rigid objects in motion allow for the recovery of the structure of the object, at least up to a projective transformation. In [37], the authors show a sparse point-set from a novel viewpoint and compare it to a real image from the same viewpoint to show the accuracy of the structure. Figure 18 shows a similar result, however since our correspondences are dense, we can actually render the novel view that validates our structure estimation. The novel viewpoint is well above the observed viewpoints, yet the rendering as well as the displayed structure is fairly accurate. Note that only the set of points that were identified as edges in I are shown; this is not the result of simple edge detection on the rendered view. We use this display convention because the entire point-set is too dense to allow the perception of structure from a printed image. However, the rendered image shows that our estimated structure was very dense. It is important to note that the only assumption that we made about the object is that it is a rigid, piecewise smooth object. To achieve similar results from sparse correspondences would require additional object knowledge, namely that the object in question is a cube and has planar faces. It is also important to point out that this is not a standard stereo pair since the scene contains multiple objects undergoing independent motion.

6 Applications

6.1 Automatic Object Removal

We demonstrate an application of our algorithm to the problem of video object deletion in the spirit of [22, 46]; see Figure 19. The idea of using motion segmentation information to fill in occluded regions is not new, however previous approaches require a high frame rate to ensure that inter-frame disparities clearpage
are small enough for differential optical flow to work properly. Here the interframe disparities are as much as a third of the image width.

6.2 Structure From Periodic Motion

The periodicity of moving objects such as pedestrians has been widely recognized as a cue for salient object detection in the context of tracking and surveillance, see for example [11, 26]. In addition, it can be used for 3D reconstruction. The key idea is very simple. Given a monocular video sequence of a periodic moving object, any set of period-separated frames represents a collection of snapshots of a particular pose of the moving object from a variety of viewpoints. This is illustrated in Figure 20. Thus each complete period in time yields one view of each pose assumed by the moving object, and by finding correspondences in frames across neighboring periods in time, one can apply standard techniques of multiview geometry, with the caveat that in practice such periodicity is only approximate.

One consequence of this matching between phase-separated frames is that the motion of the object between the two frames can be quite large. This is where we can apply our motion segmentation technique to segment out the object in question before attempting the reconstruction. In Figure 21, we show the results from the segmentation for two frames.

Figure 22(c) shows the sparse 3D structure recovered for the T_o-separated frames of a walking person shown in Figure 22(a,b). To get this reconstruction, we estimated the epipolar geometry between the two frames, performed bundle adjustment to improve the estimation and computed depth as described in [20]. A detail of the head and left shoulder region is shown in Figure 22(d) from a viewpoint behind the person and slightly to the left. Here we can see that the qualitative shape of the head relative to the sleeve region is reasonable.

The set of points used here consists of (i) the Förstner interest points used to estimate the fundamental matrix and (ii) the neighboring Canny edges with correspondences consistent with the epipolar geometry. Many points appear around the creases in the clothing, but this leaves several blank patches around the lower shirt and the arm.

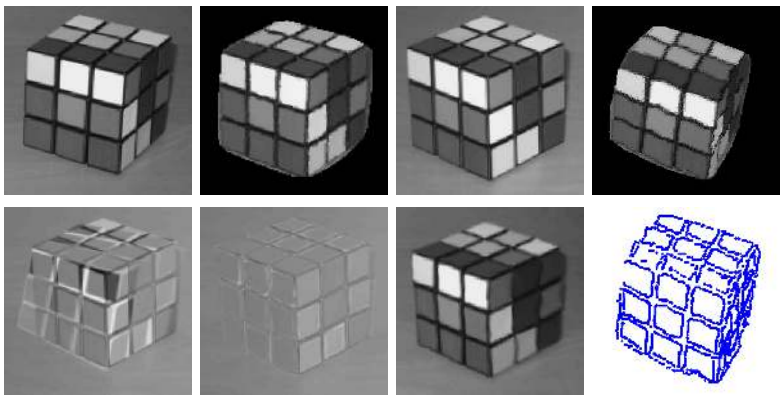

Figure 18: Rubik's Cube – Detail. (Row 1) Original frame I, synthesized intermediate frame, original frame I', A synthesized novel view, (Row 2) difference image for the planar fit, difference image for the spline fit, $\mathcal{T}(I)$, the estimated structure shown for the edge points of I. We used dense 3D structure to produce the novel view.

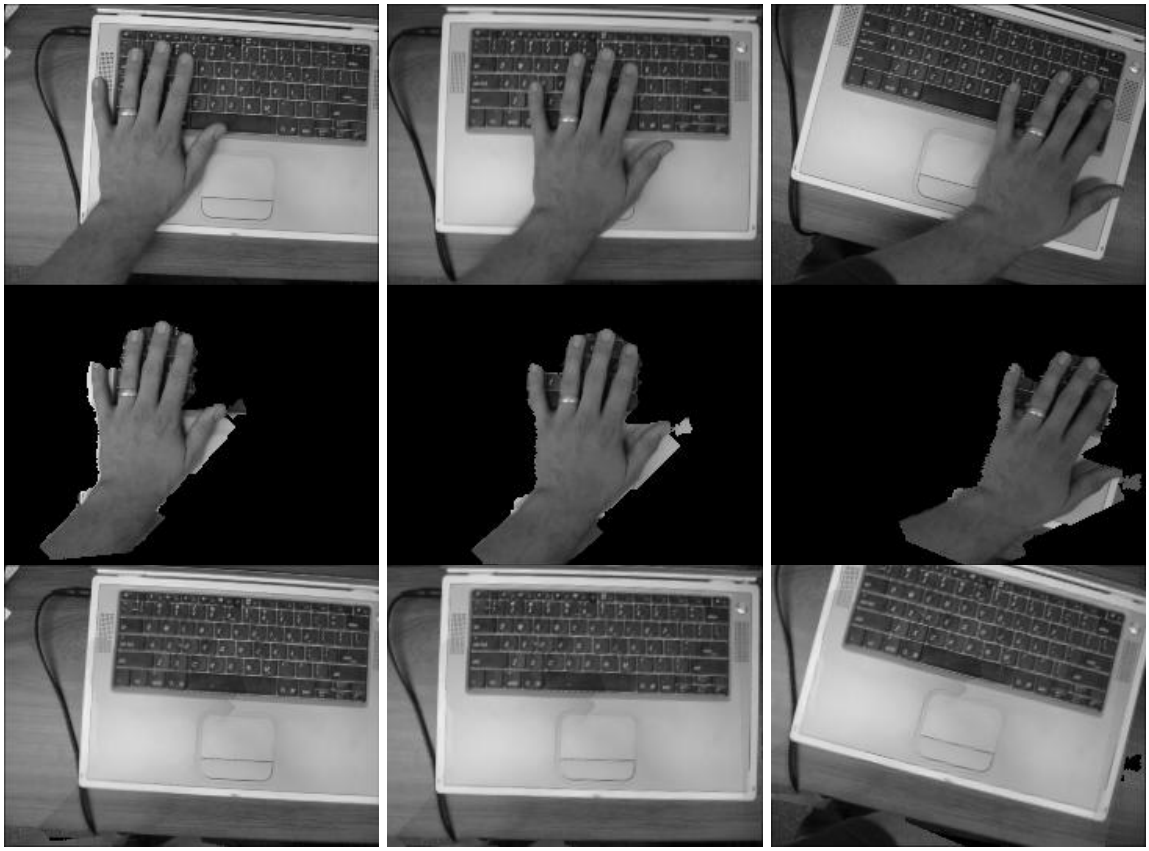

Figure 19: Illustration of video object deletion. (1) Original frames of size 180×240. (2) Segmented layer corresponding to the motion of the hand. (3) Reconstruction without the hand layer using the recovered motion of the keyboard. Note that no additional frames beyond the three shown were used as input.

7 Discussion

In this paper we have presented a new method for performing dense motion segmentation and estimation in the presence of large inter-frame motion.

Like any system, our system is limited by the assump-

Figure 20: Illustration of periodic motion for a walking person. Equally spaced frames from one second of footage are shown. The pose of the person is approximately the same in the first and last frames, but the position relative to the camera is different. Thus this pair of frames can be treated approximately as a stereo pair for purposes of 3D structure estimation. Note that while the folds in the clothing change over time, their temporal periodicity makes them rich features for correspondence recovery across periods.

(a) (b)

Figure 21: (a,b) Segmentation of Region of Interest for T_o-separated input frames.

tions it makes. We make three assumptions about the scenes:

1. Identifiability
2. Constant appearance
3. Dominant planar motion.

A system is identifiable if its internal parameters can be estimated given the data. In the case of motion segmentation it implies that given a pair of images it is possible to recover the underlying motion. The minimal requirement under our chosen motion model is that each object present in the two scenes should be uniquely identifiable. Consider Figure 23; in this display, several motions can relate the two frames, and unless we make additional assumptions about the underlying problem, it is ill posed. Similarly in some of the examples we can see that while the segments closely match the individual objects in the scene, some of the background bleeds into each layer. Motion is just one of several cues used by the human vision system in perceptual grouping and we cannot expect a system based purely on the cues of motion

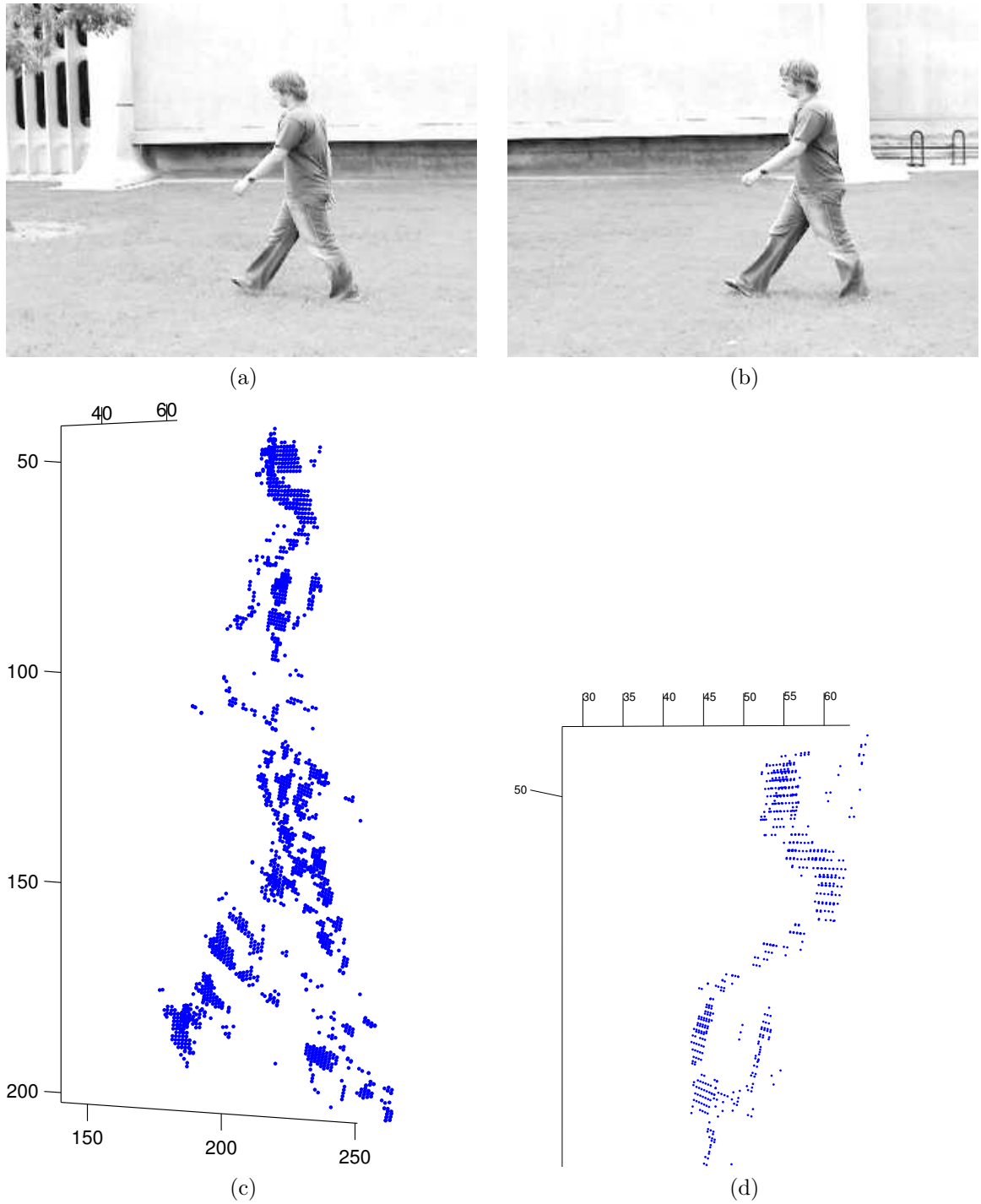

Figure 22: (a,b) T_o-separated input frames. (c) Estimated 3D structure for interest points. Here the viewpoint is diagonally behind the person and we can see the relative depths of the legs and arm as well as the head. (d) Detailed view of head and shoulder region viewed from behind the person which shows the outline of the hair, neck and shirt-sleave.

and brightness to be able to do the job. Incorporation of the various Gestalt cues and priors on object appearance will be the subject of future research.

Figure 23: Ternus Display. The motion of the dots is ambiguous; additional assumptions are needed to recover their true motion.

Our second assumption is that the appearance of an object across the two frames remains the same. While we do not believe that this assumption can be done away with completely, it can be relaxed. Our feature extraction, description, and matching is based on a fixed set of filters. This gives us a limited degree of rotation and scale invariance. We believe that the matching stage of our algorithm can benefit from the work on affine invariant feature point description [28] and feature matching algorithms based on spatial propagation of good matches [25].

Our third assumption is that the individual motion fields are predominantly planar. This is not a strict requirement and is only needed insofar as we are able to obtain the initial planar fits. The actual motion estimate and segmentation is based on the more flexible spline based model.

8. Conclusion

In this paper, we have presented a solution to the problem of motion segmentation for the case of large disparity motion and given experimental validation of our method. We have also presented an extension to handle non-planar/non-rigid motion as well as applications to automatic object deletion and structure from periodic motion. Our approach combines the strengths of the feature-based approaches (i.e., no limits on the disparity between frames) and the the direct, optical flow-based methods (i.e., provides a dense segmentation and correspondences).

Acknowledgements

Our approach for motion segmentation has appeared in [48], the extension to non-planar motion has appeared in [49], and the approach for structure from periodic motion has appeared in [3].

We would like to thank Charless Fowlkes and Ben Ochoa for helpful discussions. The images in Figures 13 and 14 are used courtesy of Dr. Philip Torr. This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48, by an NSF IGERT Grant (Vision and Learning in Humans and Machines, #DGE-0333451), by an NSF CAREER Grant (Algorithms for Nonrigid Structure from Motion, #0448615) and by The Alfred P. Sloan Research Fellowship.

References

[1] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV 95*, pages 777–784, 1995.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[3] S. Belongie and J. Wills. Structure from periodic motion. In *Spatial Coherence for Visual Motion Analysis*, Prague, Czech Republic, May 2004.

[4] M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *T-PAMI*, 18:972–986, 1996.

[5] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

[6] Y. Boykov, O. Veksler, and R. Zabih. Approximate energy minimization with discontinuities. In *IEEE International Workshop on Energy Minimization Methods in Computer Vision*, pages 205–220, 1999.

[7] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, 2001.

[8] M. Brand. Morphable 3d models from video. In *CVPR01*, pages II:456–463, 2001.

[9] M. Brand and R. Bhotika. Flexible flow for 3d nonrigid tracking and shape recovery. In *CVPR01*, pages I:315–322, 2001.

[10] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.

[11] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[12] T. Darrell and A. Pentland. Robust estimation of a multi-layer motion representation. In *Proc. IEEE Workshop on Visual Motion*, Princeton, NJ, 1991.

[13] G. Donato and S. Belongie. Approximate thin plate spline mappings. In *Proc. 7th Europ. Conf. Comput. Vision*, volume 2, pages 531–542, 2002.

[14] J. Duchon. Fonction-spline et esperances conditionnelles de champs gaussiens. *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14:19–27, 1976.

[15] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables*, pages 85 –100. Berlin: Springer-Verlag, 1977.

[16] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, vol. 24:381–95, 1981.

[17] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, 1987.

[18] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[19] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[21] M. Irani and P. Anandan. All about direct methods. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*. Springer-Verlag, 1999.

[22] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, December 1993.

[23] D. Jones and J. Malik. Computational framework to determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708, Dec. 1992.

[24] J. Kleinberg and E. Tardos. Approximate algorithms for classification problems with pairwise relationships: Metric labelling and markov random fields. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 1999.

[25] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146, 2002.

[26] Y. Liu, R. Collins, and Y. Tsin. Gait sequence analysis using Frieze patterns. In *Proc. 7th Europ. Conf. Comput. Vision*, 2002.

[27] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys. (ZAMP)*, 5:439–468, 1979.

[28] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002. Copenhagen.

[29] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.

[30] M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.

[31] H. S. Sawhney and A. R. Hanson. Trackability as a cue for potential obstacle identification and 3D description. *International Journal of Computer Vision*, 11(3):237–265, 1993.

[32] S. M. Seitz and C. R. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, 1996.

[33] A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *ICML*, 2000.

[34] S. Soatto and A. J. Yezzi. DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In *European Conference on Computer Vision*, pages 32–47. Springer, 2002. Copenhagen.

[35] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Conference on Computer Vision Pattern Recognition*, pages 194–201, Seattle, Washington, 1994.

[36] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210, 1996.

[37] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proc. IEEE Workshop on Visual Motion*. IEEE, 1991.

[38] P. H. S. Torr. Geometric motion segmentation and model selection. In J. Lasenby, A. Zisserman, R. Cipolla, and H. Longuet-Higgins, editors, *Philosophical Transactions of the Royal Society A*, pages 1321–1340. Roy Soc, 1998.

[39] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *Seventh International Conference on Computer Vision*, volume 2, pages 983–991, 1999.

[40] P. H. S. Torr, A. Zisserman, and D. W. Murray. Motion clustering using the trilinear constraint over three views. In R. Mohr and C. Wu, editors, *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 118–125. Springer–Verlag, 1995.

[41] L. Torresani, C. Bregler, and A. Hertzmann. Learning non-rigid 3d shape from 2d motion. In *NIPS 2003*, 2003.

[42] L. Torresani and A. Hertzmann. Automatic non-rigid 3d modeling from video. In *ECCV04*, pages Vol II: 299–312, 2004.

[43] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–500, Kauai, Hawaii, 2001.

[44] R. Vidal and Y. Ma. A unified algebraic approah to 2-d and 3-d motion segmentation. In *Proc. European Conf. Comput. Vision*, Prague, Czech Republic, May 2004.

[45] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[46] J. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[47] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 520–526, 1997.

[48] J. Wills, S. Agarwal, and S. Belongie. What went where. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, volume 1, pages 37–44, Madison, WI, June 2003.

[49] J. Wills and S. Belongie. A feature-based approach for determining long range correspondences. In *Proc. European Conf. Comput. Vision*, volume 3, pages 170–182, Prague, Czech Republic, May 2004.

[50] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proc. European Conf. Comput. Vision*, Prague, Czech Republic, May 2004.

[51] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. In *CVPR04*, Washington, D. C. 2004.