CVPR 97, pp. 520-527, Puerto-Rico

Smoothness in Layers: Motion segmentation using nonparametric mixture estimation

Yair Weiss
Dept. of Brain and Cognitive Sciences, MIT
E10-120, Cambridge, MA 02139, USA yweiss@psyche.mit.edu

Abstract

Grouping based on common motion, or "common fate" provides a powerful cue for segmenting image sequences. Recently a number of algorithms have been developed that successfully perform motion segmentation by assuming that the motion of each group can be described by a low dimensional parametric model (e.g. affine). Typically the assumption is that motion segments correspond to planar patches in 3D undergoing rigid motion. Here we develop an alternative approach, where the motion of each group is described by a smooth dense flow field and the stability of the estimation is ensured by means of a prior distribution on the class of flow fields. We present a variant of the EM algorithm that can segment image sequences by fitting multiple smooth flow fields to the spatiotemporal data. Using the method of Green's functions, we show how the estimation of a single smooth flow field can be performed in closed form, thus making the multiple model estimation computationally feasible. Furthermore, the number of models is estimated automatically using similar methods to those used in the parametric approach. We illustrate the algorithm's performance on synthetic and real image sequences.

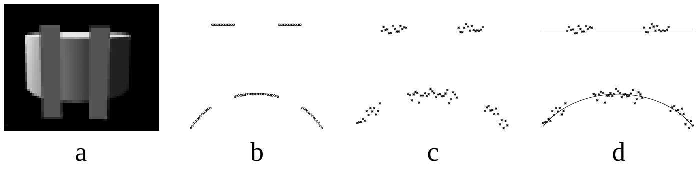

Figure 1. a. A simple three dimensional scene that can cause problems for existing motion segmentation algorithms. A cylinder is partially occluded by two bars. **b.** A cross section through the theoretical horizontal image velocity field caused by a moving camera. **c.** The same data as in (b) but with added Gaussian noise. In practice, the image velocity will be noisy. **d.** The desired description of the data.

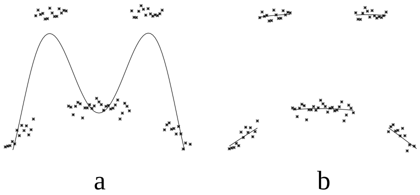

Figure 2. a. The fit of a single smooth curve to the data shown in figure 1c. Regularization causes heavy oversmoothing. **b.** Regularization with line processes. Fitting a smooth curve with discontinuities, or "line processes", causes two problems. First, there is no indication that the three occluded parts are part of a single object. Second, since no information is propagated between the occluded parts, the curvature of the cylinder is lost by the fit.

1 Introduction

Considerable progress in motion analysis has been achieved by systems that fit multiple global motion models to the image data [3, 10, 9, 8, 1, 19]. While differing in implementation, these algorithms share the goal of deriving from the image data a representation consisting of (1) a small number of global motion models and (2) a segmentation map that indicates which pixels are assigned to which model.

The advantages of these approaches over previous ones are twofold. First, by combining information over large region of the image, the local ambiguity of the image data is overcome and a reliable motion estimate can be found. Second, the derived segmentation map, in which individual pixels are grouped into perceptually salient parts, is useful for higher level processing (e.g. video database indexing, object recognition).

In order to segment images based on common motion, most existing algorithms assume that the motion of each model is described by a low dimensional parameterization. The two most popular choices are a six parameter affine model [19, 20] or an eight parameter projective model [1, 9]. Both of these parameterizations correspond to the rigid motion of a plane: the affine model assumes orthographic projection while the projective model assumes a perspective projection.

Despite the success of existing algorithms in segmenting image sequences, the assumption that motion segments correspond to rigid planar patches is obviously restrictive. Non-planar surfaces, or objects undergoing non-rigid motion cannot be grouped. In order for the motion segmentation map to be useful for higher level processing, these methods need to be extended so they can deal with non-planar surfaces and non-rigid motions.

Figure 1a shows a simple 3D scene that can cause problems for existing motion segmentation algorithms. A cylinder is partially occluded by two bars. Figure 1b shows a cross section of the horizontal component of optical flow when a camera, viewing the scene head on, is rotated horizontally about a distant point. The bars that are closer to the camera move fastest, and the velocity of points on the cylinder trace out a smooth curve. In practice, of course the velocity field will not be so perfect and Figure 1c shows the same data with added Gaussian noise.

One way to overcome the noisiness of the flow field is to use regularization to approximate the data with a single smooth function. Figure 2a shows the result of applying a typical regularization algorithm to the data shown in figure 1b. Although the noise is smoothed out, the fit suffers from heavy over-smoothing: the motions of the cylinder and the bars are averaged together. Figure 2b shows the output of a "regularization with discontinuities algorithm" [18] on the same data. Although this fixes the problem of over-smoothing, discontinuities are a bad model of occlusion (cf. [13, 3, 12]): data in a scene containing multiple occluding objects is not generated by a single discontinuous function but rather multiple smooth functions interacting nonlinearly. The results of fitting a single discontinuous function causes two problems, which can be seen in the fit. First, there is no indication that the three parts of the cylinder are part of a single object. Second, because no information is propagated between the different fragments, the cylinder is fit with three nearly straight lines, rather than curved segments. The cylindrical shape is lost in the fit.

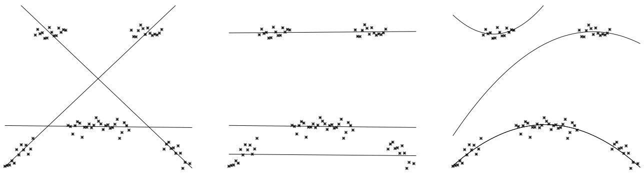

Figure 3. **a,b.** Two outputs of a multiple line fitting algorithm. Three lines are needed to achieve a reasonable fit, and the cylinder is broken apart. Various different solutions are found, and only two are shown. **c.** A result of extending the order of the models to quadratic. Although the model class is now rich enough to capture the data, estimation becomes unstable.

These limitations of regularization motivated much of the recent work in motion segmentation and led to the development of approaches that fit multiple global motion models to the data. How would parametric segmentation work on the data in figure 1? Figure 3 shows the output of a multiple parametric curve fitting algorithm to this data. The number of models was estimated automatically as in [20] (cf. [19, 1]). When the curves are restricted to be lines, different outcomes are obtained depending on initial conditions, two of which are shown in figures 3a-b. Three lines are needed to achieve a reasonable fit, and the cylinder is fragmented.

What about using a quadratic model? In this case, the model class is rich enough to capture the data, but the estimation becomes unstable. Figure 3c shows a typical output. The instability of fitting higher order models causes each of the two bars to be fit with a parabola, an example of over-fitting. Although the correct fit is sometimes obtained, it is in no way favored over other erroneous interpretations.

The instability problems associated with increasing the dimensionality of parameterization are, of course, not limited to motion analysis or even to computer vision. It is generally accepted that one should avoid fitting high order polynomials to data. Multidimensional splines and regularization theory present an elegant alternative - the functions used in this approach are flexible enough to model the data yet avoid the instability associated with high order polynomials. Regularization theory has a long history of use in computer vision [14] and has enjoyed considerable success, yet its disadvantages are well known. First, smoothness is simply a bad thing to assume over the whole image. Typically the image will contain multiple occluding objects, and assuming smoothness will lead to terrible estimates particularly in the regions of discontinuities. Second, calculating the regularized solution has typically involved highly iterative algorithms (e.g. [5, 18, 13, 12]) whose convergence may be excruciatingly slow.

Here we develop an approach to segmentation that is based on the assumption of *smoothness in layers*. Rather than assuming that the motion of the whole image varies smoothly, we assume that the motion of a given motion group or layer varies smoothly. We show how this leads to the notion of *nonparametric mixture estimation*, where the stability of the estimation process is ensured by means of a prior distribution on the class of flow fields. We present a variant of the EM algorithm that can perform the segmentation in a computationally feasible manner, and show how the algorithm is able to segment higher order flow fields while avoiding over-fitting.

2 Algorithm Description

2.1 Generative Model

The model assumes that the image data (the spatial and temporal derivatives of the sequence) were generated by K smooth motion groups. The velocity field of each group is drawn from a distribution where smooth velocities are more probable:

$$P(V) = \frac{1}{Z_1} e^{-\sum_{x,y} ||Dv(x,y)||^2}/\sigma_R^2 \tag{1}$$

Here Dv is a differential operator that penalizes fields that have strong derivatives:

$$||Dv||^2 = \sum_{n=0}^{\infty} a_n || \frac{\partial^n}{\partial x^n} v ||^2 \tag{2}$$

Thus a_0 multiplies the penalty for the magnitude of the velocity field, a_1 the penalty for the magnitude of the gradient of the velocity field, a_2 the penalty for the magnitude of the Laplacian and so forth. We follow [21] in using $a_n = \sigma^{2n}/(n!2^n)$, although similar results are obtained with other choices.

The next stage is to generate a labeling of the image, i.e. a vector $L(x,y)$ at every location such that $L_k(x,y) = 1$ if and only if position x, y will be assigned to group k. The labelings are drawn from a Markov Random Field distribution:

$$P(L) = \frac{1}{Z_2} exp \left(\sum_{x,y,x',y'} w_{x,y,x',y'} L^t(x,y) L(x',y') \right) \tag{3}$$

The link weights $w_{x,y,x',y'}$ determine the distribution of labelings. For example setting $w_{x,y,x',y'} = 1$ for neighboring sites and zero otherwise makes labelings in which neighboring locations have similar labels more probable.

Now given the labeling and the velocity field of each group, the probability of observing $I_x = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}), I_t = \frac{\partial I}{\partial t}$ at location (x,y) is given by:

$$P(I_x, I_t | L, V) = \text{exp}(-\sum_k L_k (I_x^t v_k + I_t)^2 / \sigma_N^2) \tag{4}$$

where, for clarity's sake, we have omitted the dependence of $L_k, I_x, I_y, I_t, v_k, L_k$ on (x,y) and σ_N is the expected level of noise in the sequence. Similar likelihood functions have been used for the single motion case in [15, 11]. Note that here the likelihood depends on multiple velocities, but if $L_k(x,y)$ is known then the likelihood depends only on the velocity model to which a pixel is assigned.

2.2 Nonparametric mixture estimation

To estimate the parameters of this model we use the Expectation-Maximization (EM) algorithm [4]. The algorithm iterates two steps: (1) the Expectation (or E) step in which the hidden labels are replaced with their conditional expectation and (2) the Maximization (or M) step in which the velocity fields are found by maximizing their posterior probability.

Previous implementations of the EM algorithm for motion segmentation are described in [20, 10, 1]. Two aspects of the algorithm used here are similar to the implementation described in [20] and will only be described briefly:

- The number of models is estimated automatically, by initializing the algorithm with more models than will be needed. The algorithm merges redundant models and the final number of models found depends on the parameter σ_N.
- The MRF priors on the labelings make an exact E step computationally expensive and hence a consistent approximation to the MRF distribution is used for which an exact E step can be computed efficiently.

2.3 Estimating smooth flow fields using Green's functions

The distinguishing feature of our algorithm in comparison to previous EM based approaches is in the M step. It requires finding, for each model, the dense flow field that maximizes the conditional posterior probability, or equivalently minimizes:

$$\begin{aligned} J_k(V) &= \sum_{x,y} \hat{L}_k(x,y) \left(I_x^t(x,y) v(x,y) + I_t(x,y) \right)^2 \\ &\quad + \lambda \sum_{x,y} ||Dv(x,y)||^2 \end{aligned} \tag{5}$$

where the parameter λ is determined by the ratios of σ_N and σ_R in the generative model. $\hat{L}_k(x,y)$ is the "filled in" estimate for the labeling at location (x,y). It is these weights in equation 5 that cause the estimated dense flow to differ from model to model.

Since the nonparametric EM algorithm calls for minimizing equation 5 at every iteration for all models, this approach can only be computationally feasible if the minimization can be performed efficiently. We now show how this can be done.

Using the method of Green's Functions (cf. [21, 6]) it can be shown that the optimal velocity field V_k^* is a linear combination of basis flow fields,$B_i(x,y)$:

$$V_k^*(x,y) = \sum_i \alpha_i B_i(x,y) \tag{6}$$

There is a basis flow field centered at every pixel where the image gradient is nonzero:

$$B_i(x,y) = G(x-x_i, y-y_i)\left[I_x(x_i,y_i), I_y(x_i,y_i)\right]^t \quad (7)$$

The scalar valued function $G(x,y)$ is the Green's function corresponding to the differential operator D in equation 5 (cf. [16]). It is a solution to:

$$D^* D G = \delta(x,y) \quad (8)$$

For the differential operator used here, the Green's function is a two dimensional Gaussian [21]. The coefficients α are the solution to the linear system:

$$(WM + \lambda I)\alpha = WY \quad (9)$$

With M_{ji} is given by the scalar product of the basis field centered on pixel i and the gradient at pixel j, Y_j is simply the temporal derivative at pixel j and W is a diagonal matrix whose diagonal elements determine the weight of a pixel in estimating model parameters $W_{ii} = \hat{L}_k(x_i,y_i)$.

Although equation 9 gives a closed form solution for the optimal velocity field for each model, its solution is still computationally prohibitive as it requires solving a linear system whose rank is equal to the number of nonzero gradients in the image. Are we then back to square one? No, because a remarkably good suboptimal solution can be found using this method in a computationally feasible way.

The suboptimal solution is obtained by using only a subset of the basis fields in equation 6. Denote by N the number of basis fields in the reduced expansion, then the N coefficients are a solution to:

$$(M^t WM + \lambda R)\alpha = M^t WY \quad (10)$$

where M_{ji} is again given by the scalar product of the basis field centered on pixel i and the gradient at pixel j, R is a NxN submatrix of M in which only the pixels which have basis functions centered on them are used, and W and Y are as before. Note that equation 10 is of rank N independent of the number of pixels. Note also the term λR in the left hand side of equation 10. It is this term that imposes the prior distribution and makes the estimation well posed regardless of the dimensionality of the parameter vector α. In general, the solution obtained by solving equation 10 will be different from one obtained by simply assuming the flow field is parameterized by a spline basis set (e.g. [17]). Finally, note that the reduced rank of the system is obtained by using only a subset of the basis fields, *not* by using a subset of the gradient constraints. The solution of the system gives the flow field spanned by the reduced basis set that best satisfies the gradient constraints at all pixels.

The difference between the optimal and the suboptimal solution depends on the image data, the differential operator D and the choice of subsets. In practice, we have found the difference to be negligible when 50 basis fields are used, chosen so that they are equally spaced on the image. To get an intuition regarding the optimal and suboptimal solutions, we generated a synthetic sequence by warping the image shown in figure 4a according to the superimposed flow. Figure 4b shows cross sections from the estimated velocity fields. The suboptimal solution is plotted with crosses, and the optimal one is plotted with circles. The solutions are indistinguishable. On a R4400 Silicon Graphics workstation, solving equation 10 to calculate the suboptimal solution took less than $1/100$ of a second, while solving equation 9 to calculate the optimal solution took over an hour.

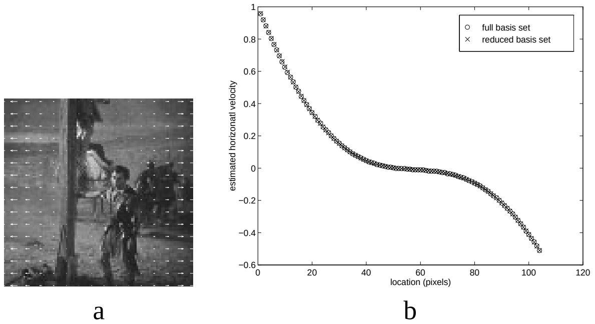

Figure 4. Using the method of Green's functions, a closed form solution can be found for fitting a smooth dense flow field to the image data. A suboptimal solution, which is computationally efficient can also be found. We have found the difference between the optimal and suboptimal solutions to be negligible. **a.** A frame from a test sequence. A second frame was generated by warping this frame with the superimposed flow field. **b.** Cross sections from the estimated velocity field using the full basis function set (circles) and using only 50 basis functions (crosses). The two solutions are indistinguishable.

Although we have used here the differential operator suggested in [21] the exact same method can be used with other differential operators. For example, we have been able to solve the Horn and Schunck [7] equations in closed form using this method.

2.4 Algorithm summary

To summarize, the statistical assumptions about the generative model are characterized by four numbers: σ, λ which embody the smoothness assumption, σ_N the assumed level of noise in the sequence and $w_{xyx'y'}$ which specifies the probability that a pixel will belong to a different model than its four neighbors.

Given these assumptions and spatiotemporal derivatives computed over the image, we use a computationally efficient EM algorithm to calculate number of models, the segmen-

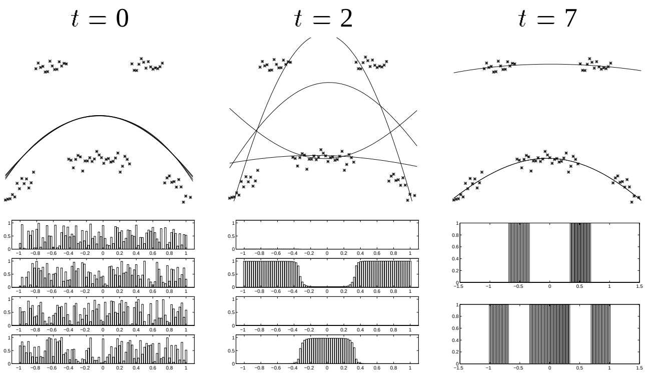

Figure 5. The performance of our algorithm on the data shown in 1c. The model is randomly initialized with four curves and automatically decides that two curves are sufficient in this case. The algorithm converges in seven iterations. Note that the curvature of the cylinder is correctly estimated.

tation of the image and a smooth dense flow field for every model.

3 Results

Before showing the results of our motion segmentation algorithm, we show the performance of a similar 1D non-parametric mixture estimation algorithm on the data discussed in the introduction and shown in figure 1c. Although some of the problems characteristic of motion segmentation are not present in 1D (e.g. the aperture problem), we choose to first illustrate the performance on a 1D problem because it enables us to display the evolution of the model's estimates. Figure 5 shows the line fits and estimated labelings, $\hat{L}_k(x)$, as a function of iteration. Note that although the label $L_k(x)$ are assumed to be binary, their "filled in" estimates $\hat{L}_k(x)$ are continuous valued and lie between zero and one. The algorithm is initialized with four curves each of which has is initially assigned a random subset of the data. Hence the initial fits are nearly identical. After six iterations, when the algorithm converges, two of the models are merged and only two unique models are needed to explain the data.

Compare the fit obtained by our algorithm to those discussed in the introduction. Unlike the regularization with discontinuities fit in figure 2b, our algorithm combines information across the different portions of the occluded cylinder and the curvature of the cylinder is apparent in the fit. Since each of the models is flexible enough, our algorithm can achieve a good fit with just two curves, unlike the line fit shown in figure 3. Since our algorithm uses a prior favoring smooth fits, it does not over-fit as does the quadratic fit shown in figure 3c.

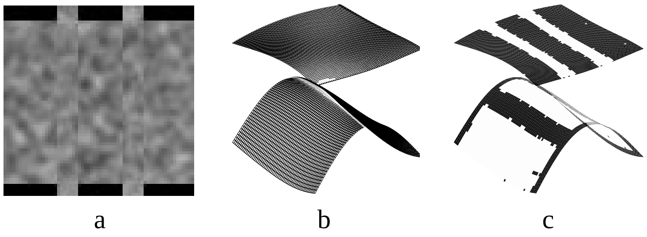

Figure 6. **a.** A single frame from the cylinder sequence. A textured cylinder is partially occluded by two textured bars, and the camera is rotating about a distant center. **b.** Reconstructed three dimensional surfaces obtained from the horizontal dense velocity fields estimated by our algorithm. **c.** The segmentation maps displayed on top of the surfaces, indicating the opacity of each layer.

We now show an example of the full 2D motion segmentation algorithm. We generated a synthetic image sequence modeled after the scene in figure 1a. Figure 6a shows a single frame from the sequence. A textured cylinder is partially occluded by two textured bars, and the camera is rotating about a distant center and translating. The camera was assumed to be orthographic and the translation was such that the mean horizontal velocity of the image was zero. Similar to the 1D case discussed earlier, this sequence is hard to segment using parametric approaches. Figures 6b-c show the output of our algorithm – it correctly estimates the number of models and the segmentation. The high quality velocity field obtained using our method enables us to reconstruct a three dimensional surface for each segment output (assuming orthography, this is simply the horizontal component of the derived dense flow fields). Figure 6b shows the surfaces obtained in this way, and Figure 6c shows the segmentation maps displayed on top of the surfaces, indicating the opacity of each layer.

We have found that sequences which are easily segmented using parametric motion models are also segmented using our approach. This is not surprising – the low dimensional motion models are often smooth and hence favored as segmentations by our model. Figure 7a shows a single frame from a sequence that was segmented using translational models in [10, 3]. A person is moving behind a plant. Figure 7b shows the segmentation derived by our algorithm. The parameter settings are identical to those used in the cylinder sequence. The number of models is correctly estimated and the different fragments corresponding to the person are grouped together. Figure 7c shows the estimated velocities.

Figure 8a shows a single frame from the MPEG flower-garden sequence that was segmented using planar models in [19, 1, 20]. Since this sequence contains large motions, we replaced the temporal derivative in equation 4 with a cal-

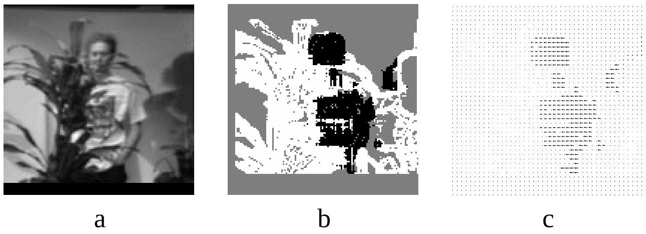

Figure 7. a. The plant sequence. A person is moving behind a plant. **b.** The segmentation found by our algorithm. Pixels belonging to the person are grouped together. **c.** The velocity estimate obtained by plotting at each pixel the velocity of the model to which that pixel is assigned.

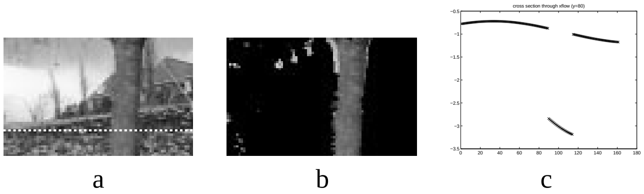

Figure 8. a. The flower garden sequence. The camera is translating approximately horizontally. **b.** The segmentation found by our algorithm. Two segments are found - one corresponding to the tree (shown) and another corresponding to the rest of the image. **c.** A cross section through the horizontal flow field taken at the dotted line in a. Note that the algorithm correctly finds the nonplanar motions of the flower bed and the tree.

culated normal velocity at each pixel. The normal velocity was calculated using a coarse-to-fine method (cf. [2]). The other aspects of the algorithm were identical to those used in previous sequences.

For the value of σ_N used two segments are found, one corresponding to the tree (shown in figure 8b) and the other corresponding to the rest of the image. An advantage of using nonparametric models for the segmentation, is that the nonplanarity of the scene can be captured in the output. Figure 8c shows a cross section through the horizontal flow recovered by our algorithm (since the camera motion is roughly horizontal, this flow is approximately related to distance from the camera). The cross section is taken at the position indicated by the dotted line in figure 8a. Note that the motions of the flower bed and the tree are smooth, curved functions. This type of structure can be easily captured in a nonparametric technique, but is lost when segments are assumed to be 3D planes. As a result, when we use the estimated motions to align the two frames, we obtain a noticeably better alignment with the nonparametric segmentation technique as compared to affine segmentation.

4 Discussion

Motion segmentation algorithms are often categorized as “direct” or “indirect” based on whether they fit models directly to the image data (e.g. [1]) or to local optical flow measurements (e.g. [19]). The particular implementation presented here would be classified as direct, since the models are fit to the spatiotemporal derivatives (see equation 4). However, the framework we have developed here is in no way restricted to spatiotemporal derivatives and can also be applied to local optical flow measurements, in cases when an indirect method is judged to be advantageous.

Our generative model assumes that for every pixel, there exists a motion model that generated the spatiotemporal derivatives at that pixel. This formulation ignores the accretion and deletion of pixels at occlusion boundaries that give rise to spatiotemporal data that is not well explained by any of the motion models. In current work, we are exploring the use of outlier models to deal with those pixels (cf. [10]).

The preceding discussions highlight the relationship between our approach and existing segmentation algorithms. Our approach fits a dense smooth flow field for every segment, and this allows us to segment non-planar surfaces or objects undergoing non-rigid motions. However, our approach shares the basic structure of existing parametric segmentation algorithms, and thus when dealing with questions of model selection, large motions and outlier rejection, we can build on the progress made by existing algorithms.

The distinction between parametric and nonparametric estimation may seem rather arbitrary. Indeed, the dense flow field by which we represent the motion of each group may be thought of as a parametric description with the number of parameters equal to the number of pixels. However, there is a fundamental difference between the two approaches. The difference is not in the number of free parameters but rather lies in what is responsible for making the estimation well posed. In parametric approaches, this is accomplished by assuming a *small number* of unknowns, while in nonparametric approaches the well-posedness is a result of assuming a *prior distribution* over the unknowns. In this work, we assumed a prior distribution where the probability of a flow field is inversely related to its smoothness and showed how to efficiently maximize the posterior probability under this assumption. An advantage of the nonparametric mixture framework developed here, is that other types of prior distributions can be easily incorporated in place of the smoothness assumption. Thus this framework can be used to investigate what assumptions are necessary to achieve stable segmentation of arbitrary image sequences.

5 Conclusion

Existing motion segmentation algorithms are able to segment image sequences by restricting the motion of each segment to lie in a low dimensional subspace. This approach has inherent limitations. If the subspace is small then it is too restrictive and cannot group together pixels undergoing more complex motions. If the subspace is rich enough to capture complex motions, the dimensionality is large and the estimation becomes unstable.

Existing regularization approaches avoid some of the shortcomings of parametric models but introduce new problems. The assumption of smoothness over the whole image leads to erroneous estimates in any scene containing multiple objects, and the solution involves slow, iterative calculations. The addition of "line processes" to the regularization framework only partially addresses these problems: line processes are a bad model for occlusion, thus disabling the propagation of information between occluded fragments, and the computational cost associated with these algorithms is even more prohibitive.

Here we have developed a new approach that builds on the recent progress made in statistically based segmentation. We have presented a generative model that embodies a prior towards smoothness, but smoothness in a layer and not smoothness over the whole image. We have shown how this leads to nonparametric mixture estimation and developed a variant of the EM algorithm that can efficiently perform segmentation under this assumption. By deriving a closed form solution to the smooth motion problem, we are able to avoid the slow iterative calculations of traditional approaches. Based on the successful performance of our algorithm on synthetic and real image sequences, we are optimistic that this framework will also be useful for other segmentation tasks in computational vision.

Acknowledgments

I thank my advisor E. Adelson for numerous discussions that influenced this work; M. Black, D. Fleet, J. Tenenbaum and X. Feng for comments on previous versions of this manuscript. Supported by a training grant from NIGMS.

References

[1] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding. In *Proc. Int'l Conf. Comput. Vision*, pages 777–784, 1995.

[2] J. Bergen, P. Anandan, K. Hana, and R. H. al. Hierarchial model-based motion estimation. In *Proc. Second European Conf. on Comput. Vision*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.

[3] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proc. IEEE Workshop on Visual Motion*, pages 173–178, Princeton, New Jersey, October 1991.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[5] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, November 1984.

[6] F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. AI Memo No: 1430, MIT AI Lab, 1993.

[7] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.*, 17(1–3):185–203, August 1981.

[8] S. Hsu, P. Anandan, and S. Peleg. Accurate computation of optical flow by using layered motion representation. In *Proc. 12th Int'l Conf. Pattern Recog.*, 1994.

[9] M. Irani and S. Peleg. Image sequence enhancement using multiple motions analysis. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 216–221, Champaign, Illinois, June 1992.

[10] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 760–761, New York, June 1993.

[11] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with application to the computation of optical flow. *IEEE Transactions on image processing*, 3(1):41–64, 1994.

[12] S. Madrasmi, D. Kersten, and T. Pong. Multi-layer surface segmentation using energy minimzation. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 774–775, 1993.

[13] J. L. Marroquin. Random measure fields and the integration of visual information. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):705–716, 1992.

[14] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[15] E. Simoncelli, E. Adelson, and D. Heeger. Probability distributions of optical flow. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 310–315, 1991.

[16] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge, 1986.

[17] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. In *Proc. Int'l Conf. Comput. Vision*, pages 757–762, 1995.

[18] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. PAMI*, 8:413–424, 1986.

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

[20] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 321–326, 1996.

[21] A. L. Yuille and N. M. Grzywacz. a mathematical analysis of the motion coherence theory. *Int'l J. Comput. Vision*, 3:155–175, 1989.