Control Point Assessment for Image Registration

L.M.G. Fonseca[1] and C.S. Kenney[2]

[1] INPE, São José dos Campos, SP, Brazil
[2] ECE Department, UCSB, Santa Barbara, CA 93106

Abstract. This paper presents several extensions of the basic CPA algorithm. First we compare CPA to standard corner detection algorithms and then turn to the question of selecting control points with adequate dispersion since this is crucial for accurate registration. Two selection methods are proposed. The first consists of clustering the control points via the Lloyd algorithm followed by selecting the dominant control point in each cluster. This 'gold standard' approach produces excellent dispersion but is costly in terms of computational effort. The second selection method consists of subdividing the image and then selecting dominant control points in each subdivision. This is extremely fast and produces results comparable to the Lloyd selection method. The paper concludes with a discussion of how LS operator norm information can be coupled with anisotropic diffusion to produce smoothed images without corner degradation.

1 INTRODUCTION

This paper considers the problem of registering a reference image g and a related image g_1. See Brown [1] for a survey of registration techniques. We assume that g_1 is a transformed version of g:

$$g_1 = F(g, p) + \eta$$

where $F(\cdot, p)$ is an image transformation operator, p is a parameter vector and η is noise. For example, we might have $g_1(x, y) = g(x', y')$ for any of the following transformations

- Simple translation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

- Translation, rotation and scaling:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

- General affine:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix}$$

Recently a new method of automatically determining control points has been described by Fonseca et al [2]. This method is summarized briefly in the next section. The main problem with this procedure for registration is that the control points may not be adequately dispersed. Two methods of dealing with this problem are considered. The first consists of clustering the control points via the Lloyd algorithm followed by selecting the dominant control point in each cluster. This 'gold standard' approach produces excellent dispersion but is costly in terms of computational effort. The second selection method consists of subdividing the image and then selecting the dominant control point in each subdivision. This is extremely fast and produces comparable results to the clustering selection method. Various examples are given to illustrate CPA and compare the dispersive algorithms.

2 Control Point Assessment

It is helpful to look at some standard optical flow methods for estimating the parameter vector p from g and g_1. Suppose we ignore the effect of noise for the moment and asssume that $g_1 = F(g, p)$ where F is the identity at $p = 0$. Expand $F(g, p)$ as a power series in p and drop any higher order terms in p:

$$\begin{aligned} g_1 &= F(g, 0) + \nabla F\, p \\ &= g + \nabla F\, p \end{aligned}$$

where ∇F is the gradient of F with respect to p.

At each point (x, y) in the image we have one equation

$$g_1 = g + \nabla F\, p \tag{1}$$

but several unknowns (the entries of p). As we have described the problem so far we assume that the parameter vector p is constant for the entire image. In most optical flow problems however p is allowed to vary with x and y. To circumvent the underdetermined nature of (1) for the case of varying parameters, there are several possible

strategies. The most common is to assume that the parameter vector varies smoothly over the entire image. This leads to the variational approach of Horn and Shunck [3]; the resulting elliptic linear system of equations for $p(x, y)$ is globally connected. This limits their usefulness and doesn't work well for discontinunous parameter fields associated with occlusions.

Both of these problems can be ameliorated by assuming that in a window about (x, y) the parameter vector is constant, see Lucas and Kanade [4] and Bergen [5]. This gives an overdetermined set of equations for p at (x, y). Solving for p using least squares gives

$$p = \left(\nabla F^T \nabla F\right)^{-1} \nabla F^T (g_1 - g)$$

As an example consider simple translation with $p = (dx, dy)^T$ and $F(g, p) = g(x + dx, y + dy)$. Expanding F about $p = 0$ and dropping higher order terms as in (1) gives

$$g_1(x, y) = g(x, y) + g_x dx + g_y dy$$

Let $g^1, \ldots, g^n$ be the window values of g about (x, y). If (dx, dy) is constant in the window then we can solve for (dx, dy) using least squares:

$$\left[\begin{array}{c} dx \\ dy \end{array}\right] = M \left[\begin{array}{c} g_1^1 - g^1 \\ \vdots \\ g_1^n - g^n \end{array}\right]$$

where M is the least squares solution matrix

$$M = \left[\begin{array}{cc} \sum g_x^i g_x^i & \sum g_x^i g_y^i \\ \sum g_x^i g_y^i & \sum g_y^i g_y^i \end{array}\right]^{-1} \left[\begin{array}{ccc} g_x^1 & \cdots & g_x^n \\ g_y^1 & \cdots & g_y^n \end{array}\right]$$

The assumption that the parameter vector is constant over a window is needed to overcome the underdetermined nature of (1). Unfortunately this approach may fail due to aperture affects. For example if the window image consists of a linear edge moving from left to right in g and g_1 then the least squares system of equations is rank deficient and thus does not have a unique solution. This problem is well known and leads to a loss of accuracy in the parameter vector estimate as the least squares system becomes ill-conditioned. This has prompted some investigators such as Irani [6] et al to assign a reliability measure to the flow estimates based on the condition number of the least squares system. Conditioning of the of the least squares system has also been shown to be directly related to the curvature of the level line running through the point (x, y).

In spite of these connections we feel that conditioning of the least squares system is not the best measure of reliability for the computed flow parameters. The reason for this is simple. Consider a flat background point in g, say a point with intensity value c at (x, y) and also c in surrounding

points. The aperture effect is maximal at such points because the least squares system is doubly rank dificient; its rank is 0! Now add a slight amount of noise to the image. Since the condition number is the ratio of the largest singular value to the smallest singular value of the least squares system, the condition number can be close to 1 under the addition of even arbitrarily small noise.

Example 1 We generated gaussian random noise with mean 0 and standard deviation $\sigma = 10^{-8}$ at each point in a 3x3 window and formed the least squares matrix

$$L = \left[\begin{array}{cc} \sum g_x^i g_x^i & \sum g_x^i g_y^i \\ \sum g_x^i g_y^i & \sum g_y^i g_y^i \end{array}\right]$$

using MATLAB with the derivatives g_x and g_y estimated by central differences. Running this test 100 times we found that the condition number varyied between a low of 1.13 which is very well conditioned to a high of 7.36. Thus none of the tests returned an ill-conditioned estimate even though the images were within 10^{-8} of an extremely ill-conditioned problem.

How then can we estimate the quality of the computed flow field? More generally how can we assess the quality of the computed estimate for the parameter vector p?

The kind of quality measure that we want is one that will identify points in the image where the difference $\|p - p_{est}\|$ is small as well as points where the difference is likely to be large. Here we are using the symbol p for the true parameter vector and p_{est} for the estimated parameter vector. If we knew p we could simply form the difference $\|p - p_{est}\|$.

This has limited usefulness since if we knew p we wouldn't be trying to estimate it. However we can make some headway by considering the case $p = 0$. If g_1 is given by $g_1 = F(g, p) + \eta$ where η is noise and $p = 0$ then since $F(\cdot, 0)$ is the identity we find that $g_1 = g + \eta$. That is the difference between g_1 and g is noise and the least squares solution p_{est} is given by $p_{est} = M(g_1 - g) = M\eta$ where

$$M = \left(\nabla F^T \nabla F\right)^{-1} \nabla F^T$$

In this case we find that

$$\|p - p_{est}\| = \|M\eta\|$$

which shows that the estimation error is determined by the norm of the least squares operator. Although we have no control over the noise we can select points at which the norm of M is small in order to minimize the estimation error.

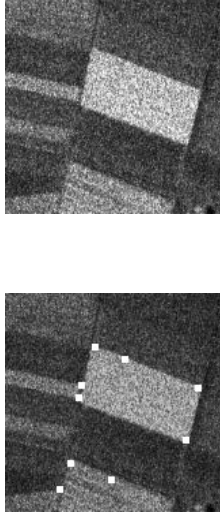**Definition:** The control points of the image g with respect

Figure 1: Original image (top); Control points determined by CPA (bottom)

to the transformation F at $p = 0$ are the points in the image where

$$r(x, y) = \max_{\eta} \frac{\|M\tilde{\eta}\|}{\|\tilde{\eta}\|}$$

attains its minimum values.

Lemma: The control points occur where the LS operator has minimum norm.

Proof: Obvious.

For related work on determining accuracy in optical flow computations see Simoncelli et al. [24] which takes a probabilistic approach and the work by Shi and Tomasi [25] which discusses the selection of good tracking features.

2.1 Comparison with Least Squares Corner Detection

The appearance of the work of Fonseca [2] et al raised the question of how CPA is related to corner detection algorithms. The standard approach to corner detection is to use the least squares procedure discussed in Haralick and Shapiro [23]. We can summarize this approach as follows. An idealized corner consists of the intersection of lines associated with regions of constant intensity. Let $X_c = (x_0, y_0)^T$ be the corner location. At any point $X = (x, y)^T$ on an edge, the intensity gradient points perpendicular to the edge. At the same time, both X and X_c lie along the edge (by assumption for both points). This means that the line connecting X to X_c is perpendicular to the gradient at X:

$$\nabla g(X - X_c) = 0$$

This equation is also true away from the edges if we assume that the image intensity is constant within regions (i.e., $\nabla g = 0$ inside regions). Rewrite this equation as $\nabla g X_c = \nabla g X$. Letting X vary over all the points X_i in the window gives a system of over determined equations for the unknown corner location X_c:

$$W X_c = b$$

where W is a 2 column matrix with row i equal to the gradient of g at the point X_i. The ith entry of the vector b is equal to $\nabla g(X_i) X_i$.

This overdetermined system has the least squares solution

$$X_c = (W^T W)^{-1} W^T b$$

where we assume that the normal matrix $W^T W$ is invertible.

Note that there are two limiting assumptions used in deriving the least squares system of equations. First the corner is the intersection of straight edges. This limits the size of the window that can be used since most edges are not straight over long distances. Second, the regions are assumed to have constant intensity, so that $\nabla g = 0$ inside regions. This assumption can be violated in images subject to speckle. The least squares form of the corner equations has some resemblance to the form of the LS operator for optical flow but with one significant difference: the term b in the corner equations has entries $b_i = \nabla g_i X_i$. This means that the equations of the corner are homogeneous of order zero and hence invariant with respect to rescaling. That is if g is replaced by sg for any scale factor s then we have the same set of equations. In many situations such invariance is desirable but in this circumstance it is not since it means that tiny amounts of noise in an otherwise uniform region can produce the effect of a strong corner signal. This is similar to the problem of measuring the reliability of optical flow by the condition number of the least squares system as discussed above in Example 1.

2.2 Multiscale Control Point Evaluation

The reference image g may be heavily contaminated with noise. Example 1 above shows that noise destroys the effectiveness of the condition number as a reliability measure for the computed optical flow vectors and the same effect limits the usefulness of standard conrner detectors as described above. Although this effect is greatly reduced for control points defined via the minima of the least squares operator, it is still possible for especially strong speckle noise to produce image artifacts that mimic good control points. Fonseca [2] et al give three strategies for dealing with this problem: 1) prefiltering 2) multiscale control point evaluation and 3) multispectral consistency.

For brevity we will discuss and use multiscale control point evaluation only since this is the most effective of the three strategies and by itself produces acceptable results.

In finding the control points for g we look for the minima of the surface defined by the norm of the least squares operator $M = M(x, y)$. In practice we found it more practical to look for the maxima of the reciprocal $\rho = 1/\|M\|$ where for convenience we have selected the Frobenius matrix norm. Since our main goal is to find the dominant motion in the image we employed a multiresolution framework using a Laplacian pyramid of Gaussian smoothing followed by subsampling. At the kth level of resolution we evaluate ρ_k. This is then interpolated back to the original image and summed to provide an overall potential surface

$$\eta = \sum \rho_k$$

As a rule of thumb in determining the number k of levels of resolution to use we have found that for images of size 2^m by 2^m setting $k = \text{Floor}(m/2)$ provides reasonable results.

Once we have formed the surface η we still need to identify the control points corresponding to the surface maxima. This can be a problem for surfaces with discontinuous variations. Fortunately the averaging over the levels of resolution reduces this problem because it provides a smoothing effect. In order to distinguish between maxima we adopted a two part test with control point maxima required to satisfy both tests.

1. Any maxima must be larger than the surrounding pixels as measured by a distance scale d. That is $\eta(x, y) > \eta(x', y')$ for $1 \le |(x - x', y - y')| \le d$. Generally we found that $d = 3$ gave good results.

2. Since we are mostly interested in the strongest values of η we introduced a tolerance parameter τ. We required that the control point maxima satisfy $\eta(x, y) - \bar{\eta} > \tau(\eta_{\max} - \bar{\eta})$ where $\bar{\eta}$ be the mean of η and $\eta_{\max}$ is the maximum value of η. By its form we want τ to lie between 0 and 1. We found that $0.25 \le \tau \le 0.75$ gave good results with the number of maxima detected decreasing as τ increases.

Example 2 Figure 1 (top) shows a detail of a satellite image of an agricultural area. The heavy speckle in the image provides a good test of the multiscale CPA method. Figure 1 (bottom) shows the detected control points for a value of $\tau = 0.5$. Note that two of the control points are the result of stairstepping effects that are reinforced by speckle.

3 Control Point Dispersion

Dispersion is vital when comparing control points between images for the purposes of determining transformation parameters such as angle of rotation, scaling etc. As a simple

Figure 2: Original image (top); Control points (bottom)

example consider the problem of determining the rotation angle between two images. Let x_1 and x_2 be two points in the reference image and suppose that X_1 and X_2 are the corresponding points in the transformed image. If the transformation consists solely of rotation through an angle θ followed by translation then we have

$$\cos \theta = v * V$$

where $v = (x_2 - x_1)/\|x_2 - x_1\|$ and $V = (X_2 - X_1)/\|X_2 - X_1\|$ are unit vectors formed by normalizing the differences $x_2 - x_1$ and $X_2 - X_1$ respectively. Here we are using the symbol $*$ to denote the inner product of the two normalized vectors.

Now suppose that the control points are computed with errors dx and dX: i.e., $\tilde{x}_2 - \tilde{x}_1 = x_2 + x_1 + dx$ and i.e., $\tilde{X}_2 - \tilde{X}_1 = X_2 + X_1 + dX$ where the tildes denote computed values.

How is the the computed value of $\cos \theta$ affected? If we let dc denote the perturbation in $\cos \theta$ we find that to first order in dx and dX we have

$$|dc| \le 2 \left(\frac{\|dx\|}{\|x_2 - x_1\|} + \frac{\|dX\|}{\|X_2 - X_1\|} \right)$$

Thus the angular error decreases as the reciprocal of the distance between the control points.

Similar results are obtained for the problem of computing the scaling parameter between two images: the error decreases as the control points spread out. In order to discuss incorporating this need for dispersion in the control points we need some notation. Denote the minima of the least squares operator norm by x_i for $i = 1, 2, \ldots N$ where N is the number of minima. Depending on the image, the

value of N can be quite large. For example the Goldhill image in Figure 2 (top) generated over 500 potential control points.

Previously we pruned these potential control points by looking only at those points with the smallest LS operator norms, i.e., the strongest control points. These points can however be rather clustered together as seen in the Figure 2 (bottom) where setting $\tau = 0.5$ reduced the number of control points from over 500 to 23 as per the multiscale pruning method described above (Section 2.2).

3.1 Control Point Dispersion Based on Clustering

The first remedy for the problem of undispersed control points is based on clustering the control points into k clusters and then selecting the strongest control point in each cluster. To find the clusters we use a simple scheme based on the Lloyd algorithm.

In this approach we seek cluster centers $y_1, \ldots, y_k$ to represent the control point data. These cluster centers are selected to minimize a distortion measure of the form

$$f(x, y) = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, y_j) \qquad (2)$$

where the jth cluster is denoted by C_j and consists of all the data points x_i that are closer to y_j than to any other cluster center. The distance in this case is measured by $d(x_i, y_j) = < x_i - y_j, x_i - y_j >$. Other distance measures can be used. See [19] for an excellent treatment of vector and scalar quantization.

Finding the cluster centers that minimize the distortion is an unsolved problem at least in a global sense. However there are some standard results for local minimization.

If we know which data values belong to a cluster then the best cluster center is the average of the cluster members. On the other hand if we know the cluster centers then the cluster membership is determined by the nearest cluster center. This suggests the basic Lloyd algorithm [21, 22].

Lloyd Algorithm

Given data values $\{x_i\}$ and initial cluster centers $\{y_j\}$ iterate Steps 1 and 2 until the cluster centers converge.

Step 1 Determine the cluster membership for C_j by finding those points x_i that are closer to y_j than to any other cluster center.

Step 2 Update the cluster centers by using

$$y_j = \frac{1}{N(C_j)} \sum_{x_i \in C_j} x_j \qquad (3)$$

where $N(C_j)$ is the number of data values in C_j.

This algorithm requires an initial guess on the cluster center locations. There are many methods of selecting these initial cluster locations. Often the cluster centers are collectively refered to as a codebook.

Some schemes build up a codebook by using the original data values. A data vector is evaluated as a codebook member by looking at its distance from the current members of the codebook. If this distance is too small (so that the data vector is near one of the codebook members) then the data vector is rejected. This is repeated until we have a codebook of the desired size; the Lloyd algorithm is then applied to 'relax' the codebook.

Splitting is another standard approach; see [20]. This method focuses on the clusters rather than the centers. The first cluster is just the entire set of data values. This large cluster is split into two clusters and these clusters are then split until we have the desired number of clusters. Splitting a cluster can be done in several ways. The most common is to find the axis of maximum variance (i.e., find the eigenvector of the covariance matrix for the cluster corresponding to the maximum eigenvalue). This axis defines a hyperplane through the center of the cluster and we use this hyperplane to split the cluster. The next cluster to be split can be selected so as to yield the maximum decrease in the distortion. We can wait until all the clusters have been found by splitting before applying the Lloyd algorithm or we can apply Lloyd after each cluster splitting. The latter relaxes the cluster centers and redefines the clusters prior to the next splitting. This takes more computational effort but usually gives a lower final distortion. This is a greedy algorithm since it seeks the maximum distortion decrease per splitting. Refinements include maximum distortion over two or more possible splittings.

The last standard codebook initialization method that we discuss is merging. Merging is the opposite of splitting. We assume that every data point is a cluster and then merge clusters together until we have the desired number of clusters. As with splitting, this is usually done in a greedy manner with the next cluster merging chosen so that the new distortion is as small as possible. The Lloyd algorithm can be applied after each merging or after the merging process is completed. See [17], [18].

These codebook initialization schemes produce good results relatively quickly. They are used since it is not known how to produce a globally optimal codebook in a reasonable number of steps. However, for control point dispersion, we found that simply subsampling the original data set produced codebooks that were adequate for our purposes. This is then followed by the Llyod algorithm until the decrease in the distortion falls below a certain tolerance (we used 1 percent as the cutoff). Once the clusters are determined we then selected the strongest member of each cluster as measured by the smallest value of the least squares opera-

tor norm. Figure 3 (top) shows the results for the Goldhill image using 23 clusters. We see that the control points are very well distributed over the image. In general we found that this approach produces excellent dispersion.

Unfortunately this 'gold standard' also is computationally expensive because each iteration of the Lloyd algorithm takes many operations and quite often many iterations are required to achieve convergence.

Because of this we turned to a fast and dirty scheme that produces almost the same results at a fraction of the cost. In this procedure we divide the original image into several smaller images and then take the strongest control points from each smaller image using the selection procedure described in Section 2.2. Note that when applied to each subimage this procedure may produce more than one control point for each subimage, depending on how the selection parameters are set. Dividing the Goldhill image in 16 smaller images produced the results seen in Figure 3 (bottom). Note that the control point dispersion lies somewhere between the tight clustering of Figure 2 (bottom) and that of the Lloyd algorithm in Fgure 3a. However this is balanced by the fact that the wonderful dispersion of the Lloyd algorithm is obtained at the cost of selecting control points that may be rather weak in the sense of having relatively large least squares operator norm. This is somewhat mitigated in the subdivision method as seen by the retention of many of the original control point locations vis a vis Figure 2 (bottom).

The significance of this is that there are two competing effects: 1) dispersing the control points leads to a decrease in the inaccuracy of the tranformation parameter computation 2) weakness in the control points leads to lower positional stability between frames, i.e., the control points are more subject to shift errors between the reference and transform images.

Thus the final parameter estimates derived from the subdivision control points were found to be just as accurate as those derived from the clustering control points and in some cases better.

4 Anisotropic Diffusion

Anisotropic diffusion was introduced by Perona and Malik [11] to preserve image edges while smoothing the interior of uniform regions. The guiding principle in this method is to use gradient information to control a diffusion process that starts with the original image as initial data. To mitigate the effects of noise the gradient information is usually computed via a multiscale approach. Thus the image u may be smoothed using a Gaussian filter of variance σ to produce an image S and then the diffusion may take the form
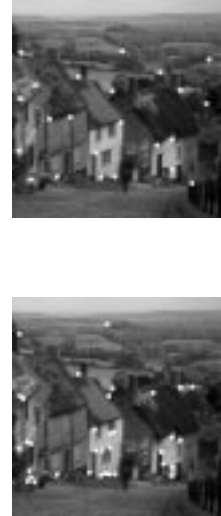
$$u_t = \nabla \cdot (f \nabla u)$$

Figure 3: Dispersed control points using the clustering algorithm (top) and the subdivision algorithm (bottom)

where the weighting function f depends on the norm of the gradient of the smooth image S. By selecting the form of f properly the diffusion increases as the norm of the gradient of S decreases thus yielding more diffusion in uniform regions and less diffusion at the edges. See Morel and Simonelli [9] for details on this and related variational methods.

One problem with this procedure is that the the noise reducing smoothing operation can cause some dislocation in the edges and especially the corners. This is seen in Figure 4 (top) in which the corners have regressed toward the interiors of their respective regions. To overcome this problem we have modified the weight function f to include information from not only the gradient but also the norm of the least squares operator. This can be done in a variety of ways, so long as f increases with respect to the LS operator norm and decreases with the gradient norm. We used multiscale implementations to evaluate both the gradient and LS operator norms. Figure 4 (bottom) shows that the results of applying this to the agricultural image; note that the corner regression is avoided.

We are currently working on extending this procedure to variational image processing in the manner of Hewer [10] et al. in which the objective function is modified to include LS operator norm information.

5 Conclusion

An analysis has been given showing that inadequate separation in selected control points can adversely effect the accuracy of estimates of rotation angle and other transform
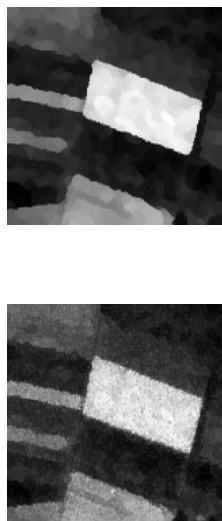
Figure 4: Aniostropic diffusion based on gradient only (top); gradient and least squares operator norm (bottom)

parameters. To deal with the problem of clustered control points for CPA we have described two methods of ensuring adequate control point dispersion. The first of these relies on the Lloyd algorithm to cluster potential control points and then select the strongest member of each cluster. This method preforms well but is costly. A second procedure subdivides the image and then selects the strongest control point in each subimage. This method is fast and produces results comparable to the Lloyd algorithm. Both of these methods presuppose that there are strong control points in the image that are adequately separated. If this is not the case then performance can be compromised. We have also shown how CPA can be applied to filtering methods such as anisotropic diffusion and variational methods to correct problems related to corner loss. This area invites further research.

References

[1] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys* **24**(4), pp. 325–376, 1992.

[2] L. Fonseca, G. Hewer, C. Kenney, and B. Manjunath, "Registration and fusion of multispectral images using a new control point assessment method derived from optical flow ideas," in *Proc. SPIE Conference*, (Orlando FLA), April 1999.

[3] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence* **17**, pp. 185–203, 1981.

[4] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Image Understanding Workshop*, pp. 121–130, 1981.

[5] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *European Conf. Comp. Vis.*, pp. 237–252, 1992.

[6] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *International Journal of Computer Vision* **12**(1), pp. 5–16, 1994.

[7] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *Image Understanding*, S. Ullman and W. Richards, eds., 1988.

[8] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Comm. on Pure and Appl. Math.* **XLii**(4), 1989.

[9] J. Morel and S. Solimini, *Variational Methods in Image Segmentation*, Birkhauser, Boston, 1995.

[10] G. Hewer, C. Kenney, and B. Manjunath, "Variational image segmentation using boundary functions," *IEEE Trans. Image Processing* **7**(9), pp. 1269–1282, 1998.

[11] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Anal. Mach. Intell.* **PAMI-12**, pp. 629–639, 1990.

[12] S. Osher and L. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numer. Anal.* **27**, pp. 919–940, 1990.

[13] S. Osher and L. Rudin, "Shocks and other nonlinear filtering applied to image processing," in *Proc. SPIE Appl. Dig. Image Proc.*, vol. 1567, pp. 414–430, 1991.

[14] G. Hewer, C. Kenney, L. Peterson, and A. V. Nevel, "PDE techniques for variational image processing," in *Proc. ICIP Conference*, (Santa Barbara CA), 1997.

[15] L. Yaroslavsky, "Linear and rank adaptive filters for picture processing," in *Digital Image Processing and Computer Graphics: Theory and Applications*, L. Dimitrov and E. Wenger, eds., 1991.

[16] Y. Deng, C. Kenney, M. Moore, and B. Manjunath, "Peer group filtering and perceptual color image quantization," in *Proc. ISCAS*, (Orlando Fl), 1999.

[17] W.H. Equitz, "Fast Algorithms for Vector Quantization Picture Coding," in *International Conf. on Acoustics, Speech, and Signal Processing*, pp. 725-728, 1987.

[18] W.H. Equitz, "A New Vector Quantization Clustering Algorithm," *IEEE Trans. Acoust. Speech Signal Proc.*, pp. 1568-1575, Oct., 1989.

[19] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

[20] Y. Linde, A Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Comm.*, COM-28, pp. 84-95, Jan. 1980.

[21] S.P. Lloyd, "Least Squares Quantization in PCM," Unpublished Bell Laboratories Technical Note. Portions presented at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, Sept. 1957.

[22] S.P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. on Information Theory*, IT-28, pp. 127-135, March 1982.

[23] Haralick, Robert M. and Linda G. Shapiro, *Computer and Robot Vision* Vol. 2, Addison-Wesley, New York, 1993.

[24] E.P. Simoncelli, E.H. Adelson and D.J. Heeger, "Probability Distributions of Optical Flow," *Proc. CVPR*, pp. 310-315, June, 1991, Lahaina, Maui, Hawaii.

[25] J. Shi and C. Tomasi, "Good Features to Track," *Proc. CVPR*, pp. 593-600, June, 1994, Seattle, Washington.