Reliable and Efficient Image Registration

B. C. Vemuri[1] S. Huang[1] S. Sahni[1]
C. M. Leonard[2] and J. Fitzsimmons[3]
[1]Department of Computer & Information Sciences & Engineering
[2]Department of Neuroscience
[3]Department of Radiology
University of Florida, Gainesville, Fl 32611

Abstract

Image registration is a very important problem in computer vision with applications to several fields e.g., medical imaging, computer graphics etc. Estimating the registration between two image data sets is formulated as a motion estimation problem. We use an optical flow motion model that allows for both global as well as local motion between the data sets. In this hierarchical motion model, we represent the flow field with a B-spline basis which implicitly incorporates smoothness constraints on the field. In computing the motion, we minimize the expectation of the squared differences (SD) energy function numerically via a modified Newton iteration scheme. The main idea in the modified Newton method is that we precompute the Hessian of the energy function at the optimum without explicitly knowing the optimum. This idea is used for both global and local motion estimation in the hierarchical motion model. We present examples of motion estimation on public domain synthetic data sequences, real medical image data and compare the performance of our algorithm with competing ones.

1 Introduction

Image registration is a very common and important problem in diverse fields such as medical imaging, computer vision, art, entertainment etc. The problem of registering two images, be they 2D or 3D, is equivalent to estimating the motion between them. There are numerous motion estimation algorithms in the computer vision literature [1, 3, 8, 10, 12] that could potentially be applied to the problem of registration of volume images specifically, MR brain scans which is the topic of focus in this paper. We draw upon this large body of literature of motion estimation techniques for problem formulation but *develop a new numerical algorithm for robustly and efficiently computing the motion parameters.* Among the motion estimation schemes, the most general of them all is the optical flow formulation. We therefore treat the problem of registering two images as equivalent to computing the flow between the data sets. There are numerous techniques for computing the optical flow from a pair of images [3, 8, 10, 12]. We chose the motion model introduced in Anandan et. al., [4] because of its ability to cope with global as well as local motions between the data sets. Thus, in this image framework, registration between two image data sets can be modeled either by a global or a local motion. In the former case, the assumption is that the entire image undergoes the same transformation while in the later case, the two data sets can differ by complex motions/transformations that are nonparametric.

A large body of literature exists on registration methods for medical image data. Most of these need to detect features/surface/contours in the images and hence the accuracy of registration is dictated by the accuracy of the feature detector. Also, additional computational time is needed in detecting these features prior to application of the actual registration scheme. *In this paper, we propose a registration method that is reliable and fast and is applicable directly to the raw data.*

Some approaches in the recent past have used the direct approach i.e., determining the registration transformation directly from the image data. One such approach is based on the concept of maximizing mutual information reported in Wells [6]. Mutual information between the model and the image that are to be registered is maximized using a stochastic analog of the gradient descent method. Reported registration experiments are quite impressive for the case of rigid motion. This formulation dealt with globally parameterized motion. There was no provision for handling local (non-parametric) and multiple motions.

In our approach, we use the motion model – presented in [12] – to compute the registration. This model consists of globally parameterized motion flow models at one end of the "spectrum"and a local motion flow model at the other end. The global motion model is defined by associating a single global motion model with each patch of a recursively subdivided input image. The flow field corresponding to the displacement at each pixel/voxel is represented by a B-spline basis. Using this spline based representation of the flow field (u_i, v_i) in the popular sum of squared differences error term i.e., $E_{ssd}(u_i, v_i) = \sum_i [I_2(x_i + u_i, y_i + v_i) - I_1(x_i, y_i)]^2$, the unknown flow field may be estimated at each pixel/voxel via numerical iterative minimization of E_{ssd}. I_2 and I_1 denote the target and initial reference images respectively. In the local flow model, the flow field is not parameterized.

In this paper, our main contributions are: (a) A new formulation of the global/local flow field computation model based on the idea of precomputing the Hessian at the optimum prior to knowing the optimum, (b) Development of a new and robust numerical solution technique based on precomputation of the Hessian matrix in a modified Newton iteration wherein the computation of the Hessian and gradients use a spline-basis representation of the (u, v) field. (c) A novel application namely, a 3D flow-based MRI data registration.

The rest of the paper is organized as follows, in section 2, we describe the global/local flow field motion model briefly. Section 3 contains the new formulation of the flow field model describing the use of precomputed Hessian and the numerical algorithm used for computing the motion/registration. In section 4, we present several examples of implementation results on standard synthetic data available on the computer vision homepage (http://www.cs.cmu.edu/čil/vision.html). We also present results of estimated motion on synthesized motion applied to real brain MRI data as well as results of registering two 3D MR brain scans of the same individual. Section 5 contains the conclusions.

2 local/global Motion Model

Optical flow computation has been a very active area of research in the field of computer vision for the last fifteen plus years. This model of motion computation is very general especially when set in a hierarchical framework. In this framework, at one extreme, each pixel/voxel is assumed to undergo an independent displacement. This is considered as a local motion model. At the other extreme, we have global motion wherein the flow field model is expressed parametrically by a small set of parameters e.g., rigid motion, affine motion etc.

A general formulation of the image registration can be posed as follows: Given a pair of images (possibly from a sequence) I_1 and I_2, we assume that I_2 was formed by locally displacing the reference image I_1 as given by $I_2(x + u, y + v) = I_1(x, y)$ The problem is to recover the displacement field (u, v) for which the maximum likelihood solution is obtained by minimizing the error given by $E_{ssd}(u_i, v_i) = \sum_i I_2(x_i + u_i, y_i + v_i) - I_1(x_i, y_i)^2$. This formula is popularly known as the *sum of squared differences* formula. In this paper, we use a slightly different error measure from the one described above. Given two gray level images $I_1(x, y)$ and $I_2(x, y)$, where I_1 is the model, I_2 is the target, to compute an estimate $\hat{T} = (\hat{u_1}, \hat{v_1}, ..., \hat{u_n}, \hat{v_n})^T$ of the true flow field $T = (u_1, v_1, ..., u_n, v_n)^T$, the motion is modeled by:

$$I_m(X_i, \hat{T}) = I_1(x_i + \hat{u}_i, y_i + \hat{v}_i) \tag{1}$$

and the expectation E of the squared difference E_{sd} is chosen to be the error criterion $J\{E_{sd}(\hat{T})\}$:

$$J\{E_{sd}(\hat{T})\} = E(I_m(X_i, \hat{T}) - I_2(X_i))^2. \tag{2}$$

We represent the displacement fields $u(x, y)$ and $v(x, y)$ by B-splines with a small number of control points $\hat{u}_k$ and $\hat{v}_k$ as in Szeliski [12]. The displacement at a pixel location i is given by

$$\begin{aligned} u(x_i, y_i) &= \sum_k \hat{u}_k w_{ik} = \sum_k \hat{u}_k B_k(x_i, y_i) \\ v(x_i, y_i) &= \sum_k \hat{v}_k w_{ik} = \sum_k \hat{v}_k B_k(x_i, y_i) \end{aligned} \tag{3}$$

where the $B_k(x_i, y_i)$ are the *basis functions* with finite support. In our implementation, we have used bilinear bases and assumed that the spline control grid is a sub-sampled version of the image pixel grid.

In the following, we describe the 2D/3D global flow models of motion in the above framework. We will then discuss the numerical optimization of the error term to determine the motion parameters in each of the motion models discussed.

2.1 2D Global Flow

When a global motion model is used to model the motion between I_1 and I_2, it is possible to parameterize the flow by a small set of parameters describing the motion for e.g., rigid, affine, quadratic and other types of transformations. The affine flow model is defined in the following manner,

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} t_0 & t_1 \\ t_3 & t_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_2 \\ t_5 \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \tag{4}$$

where, the parameters $T = (t_0, ..., t_5)^T$ are called *global motion parameters.* To compute an estimate of the global motion, we first define the spline control vertices $\mathbf{\hat{u}_j}= (\hat{u_j}, \hat{v_j})^T$ in terms of the global motion parameters:

$$\begin{aligned} \mathbf{\hat{u}_j} &= \left[\begin{array}{cccccc} \hat{x_j} & \hat{y_j} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{x_j} & \hat{y_j} & 1 \end{array} \right] \mathbf{T} - \left[\begin{array}{c} \hat{x_j} \\ \hat{y_j} \end{array} \right] \\ &= \mathbf{T_j T} - \mathbf{\hat{X}_j}. \end{aligned} \quad (5)$$

and then define the flow at each pixel by interpolation using our spline representation. The error criterion $J\{E_{sd}(\hat{T})\}$ becomes:

$$J\{E_{sd}(\hat{T})\} = E\{(I_1(X_i + \sum_j w_{ij}(\mathbf{T_j T} - \mathbf{\hat{X}_j})) - I_2(X_i))^2\}$$

Extending from 2D affine flow to 3D affine flow is quite straightforward. There are twelve motion parameters collected into a vector $T = (t_0, ..., t_{11})^T$, and the control points in terms of the motion parameters are defined as:$\mathbf{\hat{u}_j} = \mathbf{T_j T} - \mathbf{\hat{X}_j}$.

3 Numerical Solution

We now describe a novel adaptation of an elegant numerical method by Burkardt and Diehl [7] which is a modification of the standard Newton method for solving a system of nonlinear equations. The modification involves precomputation of the Hessian matrix at the optimum without starting the iterative minimization process. Our adaptation of this idea to the framework of optical flow computation with spline-based flow field representations is new and involves nontrivial derivations of the gradient vectors and Hessian matrices.

We present the modified Newton method based on the work of [7] to minimize the error term $J\{E_{sd}(\hat{T})\}$. In the following, we will essentially adopt the notation from [7] to derive the modified Newton iteration and develop new notation as necessary. The primary structure of the algorithm is given in the following iteration formula

$$\hat{T}^{k+1} = \hat{T}^k - H^{-1}(\hat{T} = \hat{T}^* = T)g(\hat{T}^k). \quad (6)$$

Where, H is the Hessian matrix and g is the gradient vector. Unlike in a typical Newton iteration, in the above equation, the *Hessian* is always computed at the optimum $\hat{T} = \hat{T}^* = T$ instead of the iteration point $\hat{T}^k$. So, one of the key problems is, how to calculate the Hessian at the optimum prior to beginning the iteration i.e., without actually knowing the optimum.

Let the vector X denote the coordinates in any image and $h : X \rightarrow X'$ a transformation from X to another set of coordinates X', characterized by a set of parameters collected into a vector T i.e., $X' = h(X, T)$. The parameter vector T can represent any of rigid, affine, shearing, projective etc. transformations. Normally the Hessian at the optimum will explicitly depend on the optimum motion vector and hence can not be computed directly. However, a clever technique was introduced in [7], involving a moving coordinate system $\{X^k\}$ and an intermediate motion vector $\tilde{T}$ to develop the formulas for precomputing the Hessian. This intermediate motion vector gives the relationship between $\{X^k\}$ of iteration step k and $\{X^{k+1}\}$ of iteration step $k+1$:

$$X^k = h(X, \hat{T}^k) \quad (7)$$

$$\begin{aligned} X^{k+1} = h(X^k, \tilde{T}^{k+1}) &= h(h(X, \hat{T}^k), \tilde{T}^{k+1}) \\ &= h(X, \hat{T}^{k+1}) \end{aligned} \quad (8)$$

The Hessian at the optimum is given by:

$$\tilde{H} = 2E\left\{ \left(\frac{\partial h(X,T)}{\partial T}\right)^T \frac{\partial I_1(X)}{\partial X} \left(\frac{\partial I_1(X)}{\partial X}\right)^T \frac{\partial h(X,T)}{\partial T} \right\}\Bigg|_{T=0} \quad (9)$$

Whereas, the gradient vector with respect to $\tilde{T}$ is given by:

$$\tilde{g}(\hat{T}^k) = 2E\left\{ e\left(\left(\frac{\partial I_2(X)}{\partial X}\right)^T \left(\frac{\partial X^k}{\partial X}\right)^{-1} \left(\frac{\partial X^{k+1}}{\partial \tilde{T}^{k+1}}\right)\Bigg|_{\tilde{T}^{k+1}=0} \right)^T \right\} \quad (10)$$

where $e = I_m(X, \hat{T}^k) - I_2(X)$.

Thus the modified Newton algorithm consists of the following iteration:

$$\tilde{T}^{k+1} = -\tilde{H}^{-1}\tilde{g}(\hat{T}^k) \quad (11)$$

and the estimate at step k+1 is given by:

$$\hat{T}^{k+1} = f(\hat{T}^k, \tilde{T}^{k+1}) \quad (12)$$

Where, f is a function that depends on the type of motion model used. One of the advantages of the modified Newton method is an increase in the size of the region of convergence. Note that normally, the Newton method requires that the initial guess for starting the iteration be reasonably close to the optimum. However, in all our experiments – described subsequently –

with the modified Newton scheme described here, we always used the zero vector as the initial guess for the motion vector to start the iterations. For more details on the convergence behavior of this method, we refer the reader to [7].

In the following sections, we describe a reliable way of pre-computing the Hessian matrix and the gradient vector at the optimum for the local motion model and refer the reader to [5] for further details.

3.1 H & g for 2D Local Flow

Let $\hat{X}_j, (j = 1, 2, ..., n)$ be the control points. Then the flow vector T is $(\hat{u}_1, \hat{v}_1, ..., \hat{u}_n, \hat{v}_n)^T$. Actually, local flow is equivalent to pure translation at each pixel and hence, the Hessian at the optimum is only related to the derivatives with respect to the original coordinates and does not depend on the flow vector. Therefore, it can be calculated without introducing $\tilde{T}$ as shown below.

$$Let\ \partial = (w_1 \frac{\partial}{\partial x}, w_1 \frac{\partial}{\partial y}, ...w_n \frac{\partial}{\partial x}, w_n \frac{\partial}{\partial y}) \tag{13}$$

The Hessian at the optimum is then given by:

$$H^*_{ij} = \tilde{H}_{ij} = 2E\{\partial_i I_1(X) \partial_j I_1(X)\} \tag{14}$$

Whereas the gradient vector is,

$$\tilde{g}(\hat{T}^k) = 2E\left\{(I_m - I_2)N^T M^{-T} \frac{\partial I_2(X)}{\partial X}\right\} \tag{15}$$

Where, the matrices M and N are given by

$$\begin{aligned} M &= \tfrac{\partial X^k}{\partial X} \\ &= \left[\begin{array}{cc} 1 + \sum_j (w_j)_x{}' \hat{u}^k_j & \sum_j (w_j)_y{}' \hat{u}^k_j \\ \sum_j (w_j)_x{}' \hat{v}^k_j & 1 + \sum_j (w_j)_y{}' \hat{v}^k_j \end{array} \right] \end{aligned} \tag{16}$$

and

$$\begin{aligned} N &= \left. \left(\frac{\partial X^{k+1}}{\partial \tilde{T}^{k+1}} \right) \right|_{\tilde{T}^{k+1}=0} \\ &= \left[\begin{array}{ccccc} w_1 & 0 & ... & w_n & 0 \\ 0 & w_1 & ... & 0 & w_n \end{array} \right] \end{aligned} \tag{17}$$

respectively. We can now substitute equations 14 and 15 into equation 11 yielding $\tilde{T}^{k+1}$ which upon substitution into equation 12 results in $\hat{T}^{k+1}$. Hence, the numerical iterative formula 12 used in computing the local motion becomes:

$$\hat{T}^{k+1} = \hat{T}^k - \tilde{H}^{-1} \tilde{g}(\hat{T}^k) \tag{18}$$

The size of $\tilde{H}$ is determined by how many control points are used in representing the flow field (u, v). For 3D problems, $\tilde{H}$ is $(3n \times 3n)$ where n is the number of control points. For such large problems, numerical iterative solvers are quite attractive and we use a preconditioned conjugate gradient (PCG) algorithm [10, 12] to solve the linear system $\tilde{H}^{-1}\tilde{g}(\hat{T}^k)$. The specific preconditioning we use in our implementation of the local flow is a simple diagonal Hessian preconditioning. More sophisticated Preconditioners can be used in place of this simple preconditioner [10].

4 Implementation Results

We tested our algorithm with the various types of motions discussed in earlier sections. Our test data consisted of publicly available synthetic/real image sequences used by most in the optical flow community. In addition, we also present applications of our algorithm in registration of medical image data. The medical data are a slice of an MR brain scan for the 2D case and the entire scan for the 3D case. To measure the accuracy of the registration, we used the relative error between the optimum motion vector and computed motion vector defined using the vector 2-norm. Let the optimum motion vector be denoted by T_{opt}, and the computed motion vector be $\hat{T}$, then the error is defined as $error = \frac{\|\hat{T} - T_{opt}\|_2}{\|T_{opt}\|_2}$. This measure is used in our motion experiments – on medical images – to quantify the accuracy of the registration. This measure is more meaningful than the average angle error between computed and true flow vectors used in Barron et al., [3]. In addition, it can be used for global as well as local flow models of motion. For non-medical data, we used standard synthetic data made publicly available by Barron et. al [3]. When testing with non-medical data, to facilitate comparison with competing methods, we used the error measure proposed in [3].

4.1 Experiments on Non-medical Data

In this subsection we compare – using the angle error measure in [3] – the performance of our algorithm with competing optical flow estimators in literature [2, 4, 9, 10, 12]. The comparison was done on methods that have reported 100% flow density estimates. In the comparison, we have included some of the robust statistics based schemes that have reported results on flow estimates for the chosen images. We perform the comparison via experiments on standard 2D image data made available on the web by Barron et al. [3]. The following tables summarize our results of (100%) flow field computation on the translating tree, diverging tree and the Yosemite image sequences. The tables depict the average/mean and standard deviation (S.D.) of the error in computed flow. In all the examples, as is evident, our scheme outperforms the

Technique	Translating Tree		Diverging Tree	
	mean	S.D.	mean	S.D.
B&A [4]	0.481°	0.481°	3.484°	3.891°
Sim [11]	0.365°	0.233°	3.621°	3.075°
Szeliski [12]	0.35°	0.34°	0.78° ∗∗	0.47°
Lai [10]	0.30°	0.27°	1.32°	1.04°
Ours	0.20°	0.10°	0.77°	0.35°

Table 1: 2D Local motion tests on translating & diverging tree images. The double star indicates a time sampling rate of four.

Technique	mean	S.D.
B&A [4]	4.46°	4.21°
A&S [2]	3.17°	6.46°
Szeliski [12]	2.45°	3.05°
Ju [9]	2.16°	2.00°
Lai [10]	1.97°	1.38°
Ours	1.83°	1.85°

Table 2: 2D Local motion test on the Yosemite sequence.

competing methods.

The above errors were computed for two consecutive image frames and 100% flow density. Results for this Yosemite image sequence were obtained by computing the flow on a cropped image consisting of the lower 176 rows of image.

4.2 Experiments on Medical 2D/3D Data

The first experiment involved testing for the global 2D affine motion model. An arbitrary slice of a 3D MR brain scan was selected and a known affine motion was applied to it in order to generate the second data set. These two images were used in testing the affine motion model. The table 3 summarizes the true and computed motion for different affine motions. The computed motion results are shown for three different methods namely, the Lavenberg-Marquardt (LM) method used in Szeliski and Coughlan [12] for minimization of the SSD error and the modified Newton method of Burkhardt [7] for minimizing the expectation of the squared difference error and our method. This experimental study involved comparing the performance of our algorithm with the performance of our implementation of the published algorithms in Szeliski et al., [12] and Burkhardt [7]. Our implementation of the algorithm in [12] was able to reproduce their published results of estimated motion for public domain image data thereby lending credence to our implementation. The motion used in the testing included large expansive and contractive scaling as well as rotation and translations. Although Burkhardt's algorithm depicts better accuracy in some of the tests, we observe that it lacks consistent performance in accuracy. When an algorithm yielded over 100% relative error between the optimal motion vector and the computed motion vector after reaching a maximum iteration count, it was deemed to have diverged. Note that our method is quite reliable compared to the other two methods and is able handle large scaling fairly accurately. Our algorithm takes 13.0 seconds of CPU time on an Ultrasparc-1 to achieve the registration for the affine motion shown in the last row of the table 3. The number of control points and the number of patches used in our algorithm for this experiment are 25 and 16 respectively.

True Motion	Error from		
	SZ [12]	Burk. [7]	Ours
(1.1,0.0,0.0,0.0,1.1,0.0)	1.02%	0.60%	0.68%
(1.5,0.0,0.0,0.0,1.5,0.0)	3.11%	19.54%	1.16%
(2.5,0.0,0.0,0.0,2.5,0.0)	diverge	1.45%	1.34%
(0.8,0.0,0.0,0.0,0.8,0.0)	26.32 %	0.75%	0.88%
(0.6,0.0,0.0,0.0,0.6,0.0)	diverge	diverge	4.25%
(1.13,-0.41,2,0.41,1.13,4)	0.69%	0.25%	0.27%
(1.50,-0.39,2,0.39,1.45,4)	1.11%	16.58%	0.54%

Table 3: Comparison of computed motion using the LM, Burkhardt's and our method for the affine motion model

Finally, we applied our algorithm in 3D to register two MR brain scans of the same individual. These two data sets were of size $(256, 256, 122)$ and $(256, 256, 124)$, respectively. We applied our algorithm to the raw 3D data, computed the motion using 125 control points in the (u, v, w) representation,and then applied the inverse of the computed transformation to the second data set. For visualization purposes, we extracted the heads from the first and transformed second data sets and superimposed them as shown in the figure 1. The registration shown involves an application of the computed global affine flow followed by a refinement with local affine flow and appears to be visually quite accurate. The evident misregistration could be attributed to the error in the head extrac-

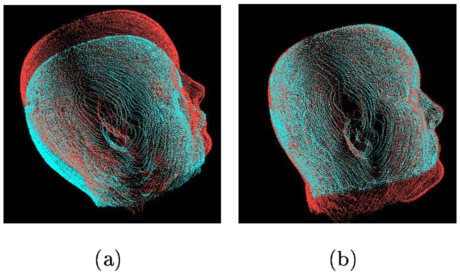

Figure 1: Experimental results for 3D real data, (a) superposed images before registration (b) after registration.

tion process. Our global flow computation algorithm takes 10 *mins*. CPU time – on a SUN Enterprise (using only one 250Mhz processor) to register two MR brain scans of size $(256, 256, 122)$ each.

5 Conclusion

In this paper, we presented a new image registration algorithm that is reliable and efficient. The algorithm is well suited for registering data from the same modality but is not applicable in its current form for data from disparate modalities. The algorithm has several strengths namely, it can be applied directly to raw image data, it is very stable, can cope with large motions including scaling and is computationally efficient. The algorithm has the capability to handle global as well as local motions and proceeds to first globally register the data sets and then refines the registration locally if necessary. The effectiveness of the registration was measured quantitatively and the percentage error in registration achieved by using our algorithm was compared with competing methods. Flow estimates obtained using our algorithm on several public domain synthetic (2D) image sequences was compared with competing methods and found to be more accurate. Promising registration performance was also demonstrated on 3D volumetric image data specifically, MR brain scans. Our future efforts will be focussed on developing computationally efficient methods for registering data sets from disparate sensors e.g., MR and PET etc.

Acknowledgements

This research was supported in part by the grant NIH-R01-LM05944.

References

[1] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from a sequences of images – a review,. *Proc. of the IEEE,*, 76((8)):917–935, 1988.

[2] A. Bab-Hadiashar and D. Suter. Optical flow calculation using robust statistics. In *Proc. of the IEEE CVPR*, pages 988–993, 1997.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques,. *Intern. J. Comput. Vision*, 12(1):43–77, 1994.

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

[5] B. C. Vemuri et. al. Reliable and efficient image registration,. Technical Report UF-CISE-TR97-019, Dept. of CISE, University of Florida, Gainesville, Fl. 32611, 1997.

[6] W. M. Wells III et. al. Multi-modal volume registration by maximization of mutual information,. *Medical Image Analysis,*, 1(1):35–51, March 1996.

[7] H.Burkhardt and N.Diehl. Simultaneous estimation of rotation and translation in image sequences,. *Proc. of the European Signal Processing Congerence,*, pages 821–824, 1986.

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

[9] S. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency,. In *Proc. of the IEEE CVPR'96,*, pages 307–314, San Fransico, CA, 1996.

[10] S. H. Lai and B. C. Vemuri. Robust and efficient computation of optical flow,. In *IEEE symposium on Computer Vision and Pattern Recognition (extended version to appear in IJCV Vol. 29, 1998)*, pages 455–460, Miami, Fl., 1995.

[11] D. Sim and R. Park. Robust reweighted map motion estimation. *IEEE Trans. on PAMI*, 20(4):353–365, 1998.

[12] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration,. In *IEEE Conf. Comput. Vision Patt. Recog.,*, pages 194–201, Seattle, 1994.