

Elastically Adaptive Deformable Models

Dimitri Metaxas and Ioannis A. Kakadiaris

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6228
{dnm, ioannisk}@grip.cis.upenn.edu

Abstract. We present a novel technique for the automatic adaptation of a deformable model's elastic parameters within a Kalman filter framework for shape estimation applications. The novelty of the technique is that the model's elastic parameters are not constant, but time varying. The model for the elastic parameter variation depends on the local error of fit and the rate of change of the error of fit. By augmenting the state equations of an extended Kalman filter to incorporate these additional variables and take into account the noise in the data, we are able to significantly improve the quality of the shape estimation. Therefore, the model's elastic parameters are initialized always to the same value and they subsequently modified depending on the data and the noise distribution. In addition, we demonstrate how this technique can be parallelized in order to increase its efficiency. We present several experiments to demonstrate the effectiveness of our method.

1 Introduction

Physics-based modeling provides a powerful mechanism for quantitatively modeling an object's shape, structure and motion [3, 5, 9, 11, 13, 15, 16, 17, 12, 18, 19, 1]. Deformable models offer a data-driven recovery process, in which forces derived from the image deform the model until it fits the data. In previous work [16], we developed a physics-based framework for recovering shape and nonrigid motion from both 2-D and 3-D data using deformable part models with local and global deformations. Global deformation parameters represent the salient shape features of natural parts, and local deformation parameters capture shape details.

Most formulations of physics-based models assume that the user correctly initializes the model's parameters which determine the goodness of fit of the model to the given data. Recently, there have been several attempts to overcome the above problem. Blake et al. [1] developed a new technique based on ideas from adaptive control theory and maximum likelihood estimation, to learn the model dynamics for tracking in real time 2D curve motions similar to those in the training set. Samadani [19] provides rules to estimate and adjust the parameters of a snake to avoid instabilities and improve the accuracy of shape estimation. Larsen [13] proposes guidelines to determine the bounds of the optimal elastic parameters.

In this paper, we propose a new formal methodology to automatically determine a deformable model's elastic parameter values [16] which affect the accuracy of the shape estimation. The technique is useful in applications where accurate shape estimation is desired and the shape characteristics of the data may vary significantly in space and/or time. Such applications include accurate contour estimation from biomedical data, and extraction of 3D shape from range data for reverse engineering. In all these applications, minimal human intervention in terms of defining the model's elastic parameters and accuracy is desired.

Our technique is based on the use of a model for the modification of the model's elastic parameters and the stiffness matrix. The characteristic of this model is that each of the elastic parameters are modified based on the local error of fit and the local rate of change of the error of fit. If the model's initial elastic parameters are sufficient to fit the given data within a user specified tolerance, then their change during the fitting process will be minimal. Otherwise, they will gradually change based on the above criteria. In particular, the elastic parameters decrease when the model has not fit the data, and increase when the model is close to the data. The increase of the elastic parameters in the latter case, has the effect of anchoring the model to the portion of the data it has fit and also improves the continuity of the solution.

Kalman filtering has also been used by many researchers in computer vision to account for noise in the data [1, 2, 5, 14, 15, 18]. Based on our previous experience with incorporating a dynamic deformable model to an extended Kalman filter framework, we develop a *modified* extended Kalman filter. This filter allows the simultaneous modification of the model's degrees of freedom and its elastic parameters. In particular, we extend the state vector of the dynamical system which corresponds to our deformable model, to include the model's elastic parameters. This modification is based on the theory of dynamic system parameter identification [8].

We present a series of experiments with two and three dimensional data to demonstrate the effectiveness of our technique in accurate shape estimation, where the elastic parameters are always initialized to the same value.

2 Deformable Models

Geometrically, deformable models are defined on a domain Ω . The positions of points on the model relative to an inertial frame of reference Φ in space are given by a vector-valued, time varying function. We set up a noninertial, model-centered reference frame ϕ and express the position function as

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p} = \boldsymbol{\xi}(\mathbf{q}), \quad (1)$$

where $\mathbf{c}(t)$ is the origin of ϕ at the center of the model and the rotation matrix $\mathbf{R}(t)$ gives the orientation of ϕ relative to Φ . We further express \mathbf{p} as the sum of a reference shape $\mathbf{s}(\mathbf{u}, t)$ and a displacement $\mathbf{d}(\mathbf{u}, t)$. Finally, we incorporate into the vector \mathbf{q} the degrees of freedom of our model which consist of the parameters

necessary to define the translation, rotation, global and local deformations of the model [16].

Our goal when fitting the model to visual data is to recover the vector of degrees of freedom \mathbf{q} . The velocity of a point on the model is computed based on $\dot{\mathbf{x}} = \mathbf{L}\dot{\mathbf{q}}$, where \mathbf{L} is the model's Jacobian matrix. This is achieved based on forces that the data exert to the surface of the model [16]. We make our model dynamic in \mathbf{q} by introducing mass, damping, and a deformation strain energy. Based on Lagrangian dynamics, the simplified equations of motion [16] take the general form

$$\mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q, \quad (2)$$

where \mathbf{D} , and \mathbf{K} are the damping, and stiffness matrices, respectively. The generalized forces \mathbf{f}_q are computed from $\mathbf{f}_q(\mathbf{u}, t) = \int \mathbf{L}^\top \mathbf{f}$, where \mathbf{f} are the external three-dimensional forces that are exerted on the model. The stiffness matrix \mathbf{K} determines the elastic properties of the model and is calculated from a deformation energy which is a linear combination of a membrane and a thin-plate. [16].

3 Elastically Adaptive Deformable Models

According to the theory of elasticity, the relationship between the stresses ($\boldsymbol{\sigma}$) and strains ($\boldsymbol{\epsilon}$) of an elastic material is expressed as $\boldsymbol{\sigma} = dW/d\boldsymbol{\epsilon} = \mathbf{C}\boldsymbol{\epsilon}$ for a linear material, and as $d\boldsymbol{\sigma} = \mathbf{C}(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}$ for a non-linear material. Furthermore, by assuming a small stress-strain displacement¹ and the use of finite elements, we can further write $\boldsymbol{\epsilon} = \mathbf{D}\mathbf{d} = \mathbf{D}\mathbf{S}\mathbf{q}_d$, where \mathbf{d} is the material displacement, \mathbf{S} is the finite element shape matrix, and \mathbf{q}_d are the FEM nodal displacements. Based on the above definitions and the theory of elasticity we can express the (linear or nonlinear) elastic deformation energy W as

$$W = \int \boldsymbol{\epsilon}^\top \boldsymbol{\sigma} du = \mathbf{q}_d^\top \left[\int (\mathbf{D}\mathbf{S})^\top \mathbf{C}(\boldsymbol{\epsilon})(\mathbf{D}\mathbf{S}) du \right] \mathbf{q}_d, \quad (3)$$

where the stiffness matrix \mathbf{K} is defined as

$$\mathbf{K} = \int (\mathbf{D}\mathbf{S})^\top \mathbf{C}(\boldsymbol{\epsilon})(\mathbf{D}\mathbf{S}) du. \quad (4)$$

In the past we have used a combined membrane and a thin-plate energy deformation energy which can be written in the general form:

$$W = \frac{1}{2} \int (a_1 e_{11}^2 + a_2 e_{22}^2 + a_3 e_{33}^2 + a_4 e_{12}^2 + a_5 e_{23}^2 + a_6 e_{13}^2) du, \quad (5)$$

where e_{ij} are the components of the strain vector $\boldsymbol{\epsilon}$. In almost all implementations of finite elements to computer vision, the elastic parameters a_i are assumed constant across the deformable model and during model fitting, and are initialized manually at the beginning of the shape estimation process. This approach

¹ The theory easily generalizes to large stress-strains[20]

requires that the user may have to experiment for a long time before the correct initial values are identified. Second, since these parameters are assumed constant across the model, accurate shape estimation may never be achieved in case of complex data. Clearly a technique for automatically adjusting a deformable model's elastic parameters in a local fashion is necessary.

The first contribution of this paper is the development of a new method for automatically modifying the elastic parameters a_i for each of the model's finite elements. The model for the modification of each of the model's elastic parameters is based on ideas from the theory of PD (Proportional-Derivative) control. In particular, we modify during the fitting process the model's elastic parameters based on the local error of fit and the rate of change of the error of fit. In all of our experiments, we always start with the same initial value $a_i = a_0 = 0.005$ for all the model's elastic parameters. In our implementation, each finite element $j, j = 1 \dots k$, has its own elastic parameters a_{i_j} . We then fit the deformable model to the given data based on (2). For each finite element j , we define the error of fit $\|\mathbf{e}_j\|$ to be the average of the error of fit of its nodes $l, l = 1 \dots n$. In particular

$$\|\mathbf{e}_j\| = \frac{\sum_{l=1}^n \|\mathbf{d}p_{l_j} - \mathbf{x}_{l_j}(\mathbf{q})\|}{n}, \quad (6)$$

where $\mathbf{x}_{l_j}(\mathbf{q})$ is the position of node i and $\mathbf{d}p_{l_j}$ is the average position of the datapoints assigned to this node, based on algorithms defined in [16].

During the fitting process, the values of each of the elastic parameters a_{i_j} for each finite element j are modified based on

$$a_{i_j} = (a_0 - a_{min}) e^{-sgn(\mathbf{e}_j \cdot \dot{\mathbf{e}}_j)(\|\mathbf{e}_j\| + \|\dot{\mathbf{e}}_j\|)} + a_{min}, \quad (7)$$

where a_{min} is the minimum value for all the elastic parameters which for a membrane and/or thin-plate deformable model is 10^{-5} , and sgn is the *sign* function. Notice that $sgn(\mathbf{e}_j \cdot \dot{\mathbf{e}}_j)$ is positive or zero when the model is converging towards the data and negative otherwise. The whole process of fitting and elastic parameter adaptation terminates when the error of fit for each finite element is below a tolerance ϵ_{tol} specified by the user.

This way of modifying the elastic parameters has the following desired properties. In (7), the change in each of the a_{i_j} 's is always w.r.t. to their initial value a_0 . Initially, since the error of fit is large, while the rate of change of the error of fit is small or zero, the value of the a_{i_j} 's decrease exponentially to quickly improve the fitting. In the intermediate steps of the fitting, the values of the a_{i_j} 's stabilize and are not modified significantly, since the sum of the error of fit and the rate of change of the error of fit does not change significantly. When the model is very close to the desired data, the sum of the error of fit and the rate of change of the error of fit decrease (the forces assigned to each node are now small) which result in an increase of the a_{i_j} 's towards a_0 . This results in a model that achieves a smooth solution to the data where necessary, and also better "holds" the model to the desired data.

In summary, the behavior of this model is such that initially all its elastic parameters are equal. During the intermediate steps of the fitting the parameters exponentially decrease to improve the shape estimation, while when the model has almost fit the data they start to exponentially increase again towards their initial value. Therefore, the elastic parameters oscillate mostly between a_0 and a_{min} . Due to the introduction of $sgn(\mathbf{e}_j \cdot \dot{\mathbf{e}}_j)$ in (7), the model's elastic parameters are automatically increased beyond a_0 , if the model has fit the data and tries to deviate from them. Therefore, the model resists deviation from the data once it has fit them. This is an additional desired property in cases where the model has partially fit the data. It will allow the portion of the model that has not fit the data to become more elastic and fit the data, while the portion that has fit them will not be modified or become more stiff in case there is any deviation from the data.

4 Dynamic Shape Estimation

The above model for the modification of the model's elastic parameters does not take into account the noise in the data. In the past we have shown how the dynamics of a deformable model can be incorporated into an extended Kalman filtering framework to formally account for noise in the data [15, 16]. However, an extension to this formulation is necessary in order to incorporate the the model for the modification of the model's elastic parameters into a Kalman filter. The reason is that the elastic parameters are not degrees of freedom, since they do not appear in \mathbf{q} , but are the unknown parameters that determine the value of the stiffness matrix \mathbf{K} . Therefore, the problem we have is that of parameter identification in a dynamic system. In [7, 8], the theory of parameter identification in dynamic systems is presented. Based on this approach, with the addition that we have a model for the modification of the deformable model's elastic parameters² we augment the state vector of our system to include the model's elastic parameters. Therefore, the new state vector is of the form

$$\mathbf{u} = \begin{bmatrix} \mathbf{a} \\ \mathbf{q} \end{bmatrix}, \quad (8)$$

where \mathbf{a} is the vector of the model's elasticity parameters with components a_{i_j} .

Based on the above modification, we make the following definitions necessary to define the equations of a modified extended Kalman filter. Let the observation vector $\mathbf{z}(t)$ denote time-varying input data. We can relate $\mathbf{z}(t)$ to the model's state vector $\mathbf{u}(t)$ through the nonlinear observation equation

$$\mathbf{z} = \mathbf{h}(\mathbf{u}) + \mathbf{v}, \quad (9)$$

where $\mathbf{v}(t)$ represents uncorrelated measurement errors as a zero mean white noise process with known covariance $\mathbf{V}(t)$, i.e., $\mathbf{v}(t) \sim \mathbf{N}(\mathbf{0}, \mathbf{V}(t))$. If \mathbf{z} consists

² According to that theory the time derivative of the elastic parameters should have been zero, as opposed to the one we use.

of observations of time varying positions of model points at material coordinates \mathbf{u}_k on the model's surface, the components of \mathbf{h} are computed using (1) evaluated at \mathbf{u}_k ³. In case of computing an image potential, then what is being measured at every node of the model is the difference $\mathbf{z} - \mathbf{h}(\mathbf{u})$, which is what is needed for an extended Kalman filter formulation. In addition let's also assume that $\mathbf{w}(t)$ represents uncorrelated modeling errors as a zero mean white noise process with known covariance i.e., $\mathbf{w}(t) \sim \mathbf{N}(\mathbf{0}, \mathbf{Q}(t))$.

Based on the above definitions and (2), the *modified* extended Kalman filter equations for our dynamic system take the form

$$\dot{\mathbf{u}} = \mathbf{f}(\mathbf{u}) + \mathbf{w}, \quad \mathbf{z} = \mathbf{h}(\mathbf{u}) + \mathbf{v}, \quad (10)$$

where

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} \dot{\mathbf{a}} \\ -\mathbf{D}^{-1}\mathbf{K}\mathbf{q} \end{bmatrix}, \quad \dot{\mathbf{a}} = (\dot{a}_1, \dots, \dot{a}_j, \dots, \dot{a}_{6k})^\top, \quad (11)$$

$$\dot{a}_{i_j} = -(a_0 - a_{min}) e^{-sgn(\mathbf{e}_j \cdot \dot{\mathbf{e}}_j)(\|\mathbf{e}_j\| + \|\dot{\mathbf{e}}_j\|)} sgn(\mathbf{e}_j \cdot \dot{\mathbf{e}}_j) (\|\dot{\mathbf{e}}_j\| + \|\mathbf{e}_j\|). \quad (12)$$

Notice that due to the modification in the state vector we now have a fully nonlinear extended Kalman filter as opposed to our previous formulations [16]. However, the filter converges to the right solution since we impose a correct behavior to the model for the adaptation of the model's elastic parameters and the model dynamics are appropriate for our applications.

The state estimation equation for uncorrelated system and measurement noises (i.e., $E[\mathbf{w}(t)\mathbf{v}^\top(\tau)] = 0$) is

$$\dot{\hat{\mathbf{u}}} = \mathbf{f}(\hat{\mathbf{u}}) + \mathbf{P}\mathbf{H}^\top\mathbf{V}^{-1}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{u}})), \quad (13)$$

where \mathbf{H} is computed from

$$\mathbf{H} = \left. \frac{\partial \mathbf{h}(\mathbf{u})}{\partial \mathbf{u}} \right|_{\mathbf{u}=\hat{\mathbf{u}}}. \quad (14)$$

The expression $\mathbf{G}(t) = \mathbf{P}\mathbf{H}^\top\mathbf{V}^{-1}$ is known as the Kalman gain matrix. The symmetric error covariance matrix $\mathbf{P}(t)$ is the solution of the matrix Riccati equation

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^\top + \mathbf{Q} - \mathbf{P}\mathbf{H}^\top\mathbf{V}^{-1}\mathbf{H}\mathbf{P}, \quad (15)$$

where

$$\mathbf{F}(\mathbf{u}) = \frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}. \quad (16)$$

The improvement that is offered from the Kalman filter formulation is that we can formally introduce data noise statistics into the model.

³ Note that the definition of function ξ in (1) does not depend on \mathbf{a} .

5 Implementation

Since the model's equations of motion of the model are numerically well-conditioned, we partition, for computational efficiency, the full Kalman filter formulated above into two separate filters. One that includes the translation, rotation and global deformations, and another that includes only the local deformations. While the computation to the solution of the Riccati equation for the first filter is fast since the associated degrees of freedom are few, this is not the case for the second filter whose state vector includes the model's elastic parameters and the local degrees of freedom.

To solve the matrix Riccati equations at interactive rates in the latter case, we take advantage of the decomposition of the model's surface into finite elements. We use a similar approach to the one we use for the computation of the stiffness matrix \mathbf{K} . Based on the covariance of each component of \mathbf{u} that corresponds to the variable of the second Kalman filter at a given step, we compute the contribution of each element to (15), by computing the right hand side of this equation for each element. The right hand side of (15) written for each finite element results in matrices of very small dimensions compared to the size of the respective matrices for the whole system. This per element computation can be done in parallel and once we loop through all the elements and we place the result for each element to the appropriate location of \mathbf{P} (in an identical way to the computation of \mathbf{K}), we have computed the right hand side of (15) for the second filter. This significantly improves the speed of the calculations and is justified since we have partitioned the model's surface into finite elements.

While the observation covariance matrix \mathbf{V} can be computed based on the noise in the data, the system covariance matrix \mathbf{Q} serves the purpose of expressing our "belief" in the appropriateness of the dynamic model we use for our applications. This covariance matrix serves as a way of tuning the filter and achieve fast convergence [7]. In our applications however, the dynamic system we use is appropriate and we did not have to tune the filter by experimenting with various values for the elements of \mathbf{Q} . In particular we used a value of $\mathbf{Q} = 0.01\mathbf{I}$ which assumes a small system noise.

6 Results

Based on the above implementation, all our experiments run at interactive rates on a Silicon Graphics R4400 Indigo workstation. Furthermore, we always started from a unit covariance matrix \mathbf{P} . Even though that was the initial value, the subsequent structure of \mathbf{P} is not diagonal and has a form similar to \mathbf{K} . Notice that any other reasonable initial condition will work if our dynamic model is appropriate for our applications. For the global deformations we used a superellipsoid, the elastic parameters were always initialized to $a_0 = 0.005$, and we used an adaptive Euler integration method for increased stability. In addition, the noise in our data is small and we defined \mathbf{V} as $\mathbf{V} = 0.1\mathbf{I}$.

In the first experiment, we applied our technique to the semi-automated identification of the myocardial borders from breath-hold MRI. The data were

obtained from the Department of Radiology at the University of Pennsylvania. The dataset included sixteen slice locations, from the Left Ventricle apex to the level of the aortic valve. In order to determine the location of the borders with higher accuracy we magnify each image four times, and then we convolve it with an 8x8 Gaussian mask. An initial superellipsoid model of the model was placed manually at the vicinity of the border of the first slice. In this study, we concentrated at the identification of the LV endocardial contour for locations 4 to 11 in which the contour is visible. During the fitting process, the results from fitting one slice, were used as the initial model for the next slice, as if we had an evolving curve over time. Therefore the user only initializes the model in the first slice. In the second experiment our goal is to fit the shape of the

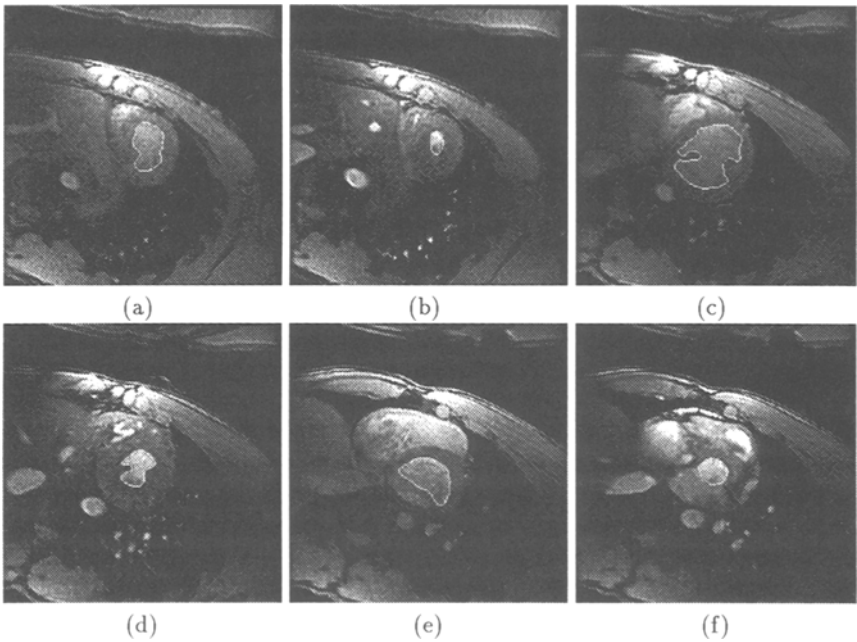


Fig. 1. Semi-automated Identification of the myocardial borders. (a), (c), (e) Locations 4, 7, 11 - End diastole; (b), (d), (f) Locations 4, 7, 11 - End systole.

outline of a head. Figure 2(a) shows the original image. The model recovered using a superellipsoid with only global deformations can be seen overlaid to the original data in Figure 2(b), while Figure 2(c) shows the contours of the data and the model. In Figure 2(d) the model has reached an equilibrium state and the error does not change over time. By varying the elastic parameters according to our technique we obtain the result in Figure 2(e) that can be compared to the original image in Figure 2(f).

In the third experiment 3D data from a human head were fitted using a deformable superquadric. Figure 3(a) shows the 1269 3D datapoints of a Viewpoint model of a human head. Figure 3(b) shows an intermediate result in fitting the

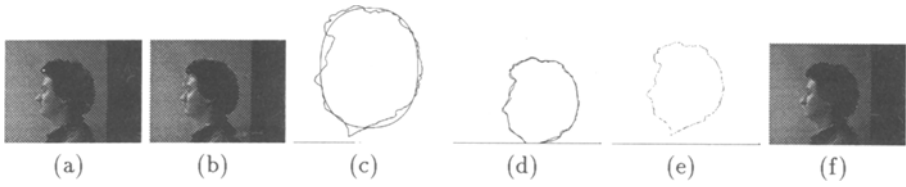


Fig. 2. Fitting 2D head contours.

data using global deformations only. Figures 3(c-d) show two views of the result obtained using the initial elastic parameters ($a_i = 0.005$), and Figures 3(e-f) show the same views of the final fitting. One can notice the difference in the accuracy of the fitting result, especially in the area around the eyes. The reason, despite the automatic modification of the model's elastic parameters is that the node distribution was insufficient. Our above technique has to be coupled with our earlier developed technique [17] for automatically adapting the model's nodes in order to further improve the fitting results. But this is beyond the scope of this paper.

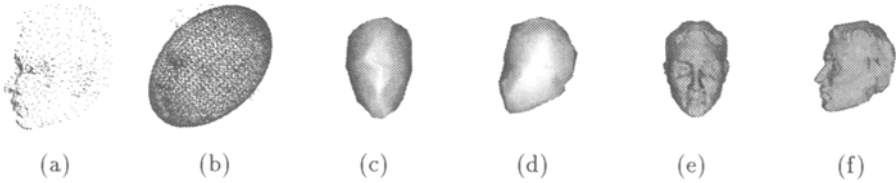


Fig. 3. Fitting 3D head data.

7 Conclusion

We have presented a new method for the automatic adjustment of the elastic parameters of a deformable model. This method coupled with our method [17] for automatically adapting a model's nodes to better fit a given dataset, is very promising towards automating the process of object shape estimation based on deformable models.

Acknowledgments

This work has been supported in part by the following grants: ARO Grant DAAL03-89-C-0031PRI, ARPA Grants N00014-92-J-1647 and DAAH-049510067, NSF Grants CISE/CDA-88-22719, MIP94-20397, IRI93-09917 and MIP94-20393.

References

1. A. Blake and M. Isard, "3D position, attitude and shape input using video tracking of hands and lips", Proc. Siggraph'94, pp. 185-192, 1994.
2. T. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images", IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(1):90-98, January 1986.
3. L. D. Cohen. "On Active Contour Models and Balloons". CVGIP: IU, 53(2), 1991.
4. C. W. Chen and T. S. Huang, "Epicardial Motion and Deformation Estimation from Coronary Artery Bifurcation Points", IEEE Proc. Third International Conference on Computer Vision (ICCV'90), pp. 456-459, Osaka, Japan, Dec. 1990.
5. E. D. Dickmanns and Volker Graefe, "Dynamic Monocular Machine Vision", Machine Vision and Applications, 1:223-240, 1988.
6. J. S. Duncan and R. L. Owen and P. Anandan, "Measurement of Nonrigid Motion Using Contour Shape Descriptors", IEEE Computer Vision and Pattern Recognition Conference (CVPR'91), pp. 318-324, Hawaii, 1991.
7. A. Gelb. "Applied Optimal Estimation", MIT Press, Cambridge, MA, 1974.
8. M. S. Grewal and A. P. Andrews. Kalman Filtering: Theory and Applications. Prentice Hall, 1993.
9. D. B. Goldgof and H. Lee and T. S. Huang, "Motion Analysis of Nonrigid Structures", IEEE Computer Vision and Pattern Recognition Conference (CVPR'88), pp. 375-380, 1988.
10. T. S. Huang, "Modeling, Analysis and Visualization of Nonrigid Object Motion", IEEE 10th International Conference on Pattern Recognition, Atlantic City, NJ, pp. 361-364, 1990.
11. M. Kass and A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models", International Journal of Computer Vision, 1(4), pp. 321-331, 1988.
12. I. A. Kakadiaris and D. Metaxas, 3D Human Body Model acquisition from Multiple views, Proc. ICCV'95, pp. 618-623, June 20 -23, Boston, MA, 1995.
13. O. V. Larsen, P. Radeva, and E. Marti, Guidelines for choosing optimal parameters of elasticity for snakes, 6th International Conference on Computer Analysis of Images and Patterns, pp. 106-113, Praga, 1995.
14. L. Matthies, T. Kanade, and R. Szeliski, "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences", International Journal of Computer Vision, 3:209-236, 1989.
15. D. Metaxas and D. Terzopoulos. Recursive Estimation of Nonrigid Shape and Motion. *Proc. IEEE Motion Workshop, Princeton, NJ*, pp. 306-311, October 1991.
16. D. Metaxas and D. Terzopoulos. Shape and Nonrigid Motion Estimation Through Physics-Based Synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, June, 1993.
17. D. Metaxas and E. Koh. "Flexible Multibody Dynamics and Adaptive Finite Element Techniques for Model Synthesis and Estimation" *Computer Methods in Applied Mechanics and Engineering*, in press.
18. A. Pentland and B. Horowitz, "Recovery of Non-rigid Motion and Structure", IEEE Trans. Pattern Analysis and Machine Intelligence, 13(7):730-742, 1991.
19. R. Samadani. Adaptive Snakes: Control of damping and material parameters. *SPIE Geometric Methods in Computer Vision*, Vol 1570, 1991.
20. O. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, 1977.