Analyzing Articulated Motion Using Expectation-Maximization

Henry A. Rowley
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
har@cs.cmu.edu

James M. Rehg
Digital Equipment Corporation
One Kendall Square, Bldg 700
Cambridge, MA 02139, USA
rehg@crl.dec.com

Abstract

We present a novel application of the Expectation-Maximization algorithm to the global analysis of articulated motion. The approach utilizes a kinematic model to constrain the motion estimates, producing a segmentation of the flow field into parts with different articulated motions. Experiments with synthetic and real images are described.

1. Introduction

Motion is an important cue for the detection and recognition of humans and their actions in video. Few objects in the world move the way people do. An algorithm capable of detecting the presence of human motion in a sequence of images would complement existing detection schemes based on faces [17] and color and form cues [5], and make robust person-detection possible. Such a system would have applications in user-interfaces [13], surveillance, and video indexing. Previous work on human motion detection and classification has relied on special properties of fronto-parallel walking motion [9] or on the reliable estimation of global translation prior to articulated motion analysis [10]. In contrast, our goal is to exploit the known kinematics of the body to detect moving figures under all viewing conditions and in the absence of any additional segmentation information.

Our approach is based on the Expectation-Maximization (EM) algorithm, which provides a means for simultaneously segmenting measurements into a set of models and estimating the model parameters. Recently, the EM algorithm has been successfully used for segmenting optical flow into independent rigid body motion models. We extend this result to the case of articulated motion analysis for human figures, and show its connection to related work on articulated object tracking. The output of our algorithm is a segmentation of the motion field into parts corresponding to different amounts of articulated motion, along with an estimation of the overall motion of each part. This representation is "tuned" to the structure inherent in human motion, and we expect it to be a useful representation for motion-based detection and classification.

2. Motion Analysis and EM

The first application of Expectation-Maximization (EM) to motion analysis, in [6], addressed the segmentation of an optical flow field into independent rigid motions. The flow field was assumed to consist of multiple regions corresponding to independently moving surfaces in the scene. The flow in each region was described by a parametric model, such as affine flow. This flow model approximates with a small number of parameters the overall flow of a rigidly-moving surface of arbitrary shape [1, 15]. Parametric flow estimates can be computed more reliably than local flow, which is often noisy and under-constrained. This in turn enables a higher quality motion-based segmentation than purely local flow can support. The difficulty is that neither the correct segmentation nor the motion estimates are known prior to analysis.

The EM algorithm makes it possible to overcome the interdependency between segmentation and motion estimation. It is based on a general statistical framework for estimating model parameters from incomplete, or missing, data [4]. In the case of motion analysis the missing data is the segmentation which assigns each image measurement (pixel) to a motion model. Given this segmentation information, the motion model parameters can be estimated using standard least squares techniques [8].

It is convenient to assume that the family of motion models takes the form of a Gaussian mixture density [11]. This means that each image measurement is drawn independently from a Gaussian distribution whose mean is a function of the motion model parameters [6, 19]. We let $R_m(\mathbf{P}, x, y)$ denote the measurement deviation, or residual, at pixel (x, y) with respect to model m whose motion is described by parameters $\mathbf{P}$. The total likelihood for a measurement at (x, y) can be written:

$$L(\mathbf{P}, x, y) = \sum_{m=1}^{M} g_m(x, y) e^{-R_m^2(\mathbf{P}, x, y)/\sigma^2}$$

The variance σ^2 controls the softness of the partition, while the mixture probabilities $g_m(x, y)$ describe the likelihood of assigning a measurement at (x, y) to model m. The EM algorithm for motion estimation using mixture models consists of two steps:

Expectation Step (E-Step): The mixture parameters are updated using the current estimate of the motion parameters and the image measurements. This step computes the conditional likelihood of each pixel originating from each motion model, as measured by the residual error associated with that assignment. It can be written as

$$g_m(x, y) = \frac{p_m e^{-R_m^2(\mathbf{P}, x, y)/\sigma^2}}{\sum_{i=1}^{M} p_i e^{-R_i^2(\mathbf{P}, x, y)/\sigma^2}}, \tag{1}$$

where p_m represents the prior probability of assigning a measurement to model m.

Maximization Step (M-Step): The motion parameters are estimated given the soft assignment of pixels to motion models provided by the mixture parameters. The maximum-likelihood estimator for each flow model selects the parameters that minimize the residual error weighted by the segmentation. It can be written as

$$\mathbf{P} = \operatorname*{argmin}_{\mathbf{P}} \sum_{(x,y)} g_m(x, y) R_m^2(\mathbf{P}, x, y), \tag{2}$$

where $\mathbf{P}$ is the vector of parameters for model m. Note that in this minimization $R_m(\mathbf{P}, x, y)$ is treated as a function of $\mathbf{P}$, while $g_m(x, y)$ (which implicitly depends on $\mathbf{P}$) is held constant. Also note that because each model is independent of the others, their parameters can be optimized separately.

Convergence results for the EM algorithm [4] guarantee that iteration of alternating E- and M-steps will cause the overall likelihood of the measured data given the parameters to increase. EM is therefore a gradient-based algorithm, and is subject to the usual pitfalls, such as convergence to local rather than global maxima. In practice, however, the EM algorithm exhibits good performance on a wide variety of problems from speech recognition to medical image segmentation. In addition, it is straightforward to incorporate robust statistical techniques [6] and a spatial coherence constraint [19] into the basic EM framework.

In order to extend the EM framework for rigid motion segmentation to the articulated case we must specify a motion model which defines the effect of the model parameters on the image measurements, and a segmentation model which describes the mapping between image and model. While the rigid motion algorithm could in principle be applied directly to figure analysis, we will show that incorporating the kinematic constraints into the motion model makes it possible to dramatically reduce the number of parameters and at the same time improve the value of the resulting segmentation.

3. An EM Algorithm for Articulated Motion

We model the human figure as an articulated object which is composed of rigid links connected by joints. Given a point on one of the links, the kinematic model can be used to compute the motion of that point in the image. If the link appearance is modeled by a texture-mapped plane, a simple motion model can be derived which expresses pixel motion in terms of joint angles [12]. The corresponding segmentation model assigns pixels to the link in the kinematic chain where their motion originates. We will demonstrate that this model, which was originally developed for object tracking, can be incorporated directly into the framework of Equations 1 and 2.

3.1. Articulated Motion Models

We can model the motion of pixels between two images by a *deformation function*, $\mathbf{F}(\mathbf{P}, x, y)$, which maps pixel coordinates in the first image into the second as a function of the motion parameter vector $\mathbf{P}$. The motion parameters can be estimated by minimizing the residual intensity difference between corresponding pixels, which can be written as

$$\begin{aligned} E(\mathbf{P}) &= \sum_{(x,y)} R(\mathbf{P}, x, y)^2 \\ &= \sum_{(x,y)} [I_2(\mathbf{F}(\mathbf{P}, x, y)) - I_1(x, y)]^2 \end{aligned} \tag{3}$$

where the summation takes place over the segmented pixels for the motion model. In the previous EM formulation, the mixture model might consist of a set of affine deformations, each with six motion parameters (see [15] for details). Alternative residual measures based on fitting optical flow data are also possible, but our method has the advantage of applying motion constraints directly to the pixel data.

Each link in the kinematic model has a separate deformation function which describes the effect of its motion on the image. An arm model, for example, might consist of an upper and lower link. The motion parameters for the deformation are the degrees of freedom of the link's kinematic chain, which are described using the Denavit-Hartenberg notation commonly employed in robotics [18]. Each link in the chain is attached either to the base or to another link. A link m has a local coordinate frame which is specified by a rigid transformation, T_m, relative to its connecting link in the chain. The three-dimensional position of each link can be obtained by composing the transforms along the kinematic chain between it and the base.

We assume that each link's appearance over a small number of images can be modeled by a texture-mapped plane attached to the link coordinate frame. This plane defines a two dimensional intrinsic coordinate system embedded in the link's local coordinate system. The plane rotates about the link's axis of symmetry, and is oriented towards the camera. Given a point in the link plane for link m with intrinsic coordinates (u, v), its location in the image is given by a position function. This function has two components: the kinematic model which positions the link plane in three dimensions, and the camera model which projects the link plane into the image. The position function for link m can be written as

$$\mathbf{f}_m(\mathbf{q}, u, v) = C \cdot T_{\text{base}} \cdot T_1 \cdot \cdots \cdot T_{m-1} \cdot \hat{T}_m \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad (4)$$

where the transformation $\hat{T}_m$ includes the mapping from the link plane to the link coordinate frame and C is a camera projection model.

This model, first presented in [12], assumes that pixels that are visible in one frame will remain visible throughout the sequence. It is violated by motions which take pixels across the occlusion boundary of the link, as in the case of side-to-side head rotation. The assumption is reasonable for short image sequences of the hand or body.

Equation 4 can be used to construct a deformation function for the motion of pixels from link m due to a change in the joint angles. This is done by composing transforms to and from the intrinsic (link plane) coordinates:

$$\mathbf{F}_m(\mathbf{P}, x, y) = \mathbf{f}_m(\mathbf{q} + \Delta\mathbf{q}, \mathbf{f}_m^{-1}(\mathbf{q}, x, y)), \qquad (5)$$

where $\Delta\mathbf{q}$ describes the motion of the links, and $\mathbf{P}$ is the set of parameters $\{\mathbf{q}, \Delta\mathbf{q}\}$. Here, we first map the pixel coordinates (x, y) in image 1 back to the intrinsic coordinates of the link, and then map them forward into the coordinates for image 2. The mapping for image 2 incorporates the change in the state parameters between the two images.

Note that the articulated model says nothing about the shapes of the links. As each link moves, its associated link plane moves with it; thus each link can describe a motion anywhere in the image (for any point on the plane). Segmentation amounts to cutting these planes into the right shapes to match the actual image motion.

The experiments described in this paper use a planar kinematic model with two links, which simplifies the $\mathbf{f}_m(\cdot)$ functions considerably. However, the motion analysis and segmentation algorithms we describe will apply to the general three-dimensional case with a perspective camera.

3.2. Motion Estimation and Segmentation

Given motion models in the form of Equation 5, motion estimation proceeds by minimizing the sum of the squared pixel residuals from Equation 3. The motion parameters are the change in state between images, $\Delta\mathbf{q}$, along with any unknown initial state values in $\mathbf{q}$. We use the Levenberg-Marquardt procedure to iteratively adjust the vector of motion parameters, $\mathbf{P}$, until the first image has been registered with the second. Letting $\mathbf{R}$ denote the vector of residual intensity differences, we can write the update step as:

$$\Delta\mathbf{P} = -(J^T J + \rho I)^{-1} J^T \mathbf{R},$$

where J is the Jacobian (derivative) matrix of $\mathbf{R}$ with respect to $\mathbf{P}$ and ρI is a regularizer term which stabilizes the update.

The residual vector $\mathbf{R}$ can be calculated directly from the image intensities and the deformation functions supplied by the articulated model. The Jacobian J, which is the derivative of $\mathbf{R}$ with respect to the parameters, is equally straightforward to compute. Each row of J results from differentiating one element of $\mathbf{R}$ as follows:

$$\begin{aligned} \frac{\partial R_m(\mathbf{P}, x, y)}{\partial \mathbf{P}} &= \frac{\partial}{\partial \mathbf{P}} \left(I_2(\mathbf{F}_m(\mathbf{P}, x, y)) - I_1(x, y) \right) \\ &= \left. \frac{\partial I_2(x, y)}{\partial (x, y)} \right|_{\mathbf{F}_m(\mathbf{P}, x, y)} \left. \frac{\partial \mathbf{F}_m(\mathbf{P}, x, y)}{\partial \mathbf{P}} \right|_{(x, y)} \end{aligned}$$

Note that this formula has two parts: one is the gradient of the second image, while the other is the derivative of the deformation function defined by the kinematics. The deformation function and its derivatives can be generated automatically from the DH parameters of the kinematic model.

The residual functions used in motion estimation, $R_m(\mathbf{P}, x, y)$, are also used in segmentation. Given the motion of the links, we can compute the residual at each pixel under each of the motion models, and assign the pixel to the link whose motion model produces the smallest residual error. This approach assumes that each pixel in I_2 has a corresponding pixel in I_1, an assumption which is violated by occlusions. In the present work, we assume that the occluded regions are small and will not effect the results significantly.

3.3. Applying EM

The previous sections have shown that the segmentation is easily computed given the motion parameters, and that motion measurement is straightforward given a correct segmentation. The EM algorithm provides a way to combine these two steps. Some modifications to the computations above are needed to convert them to the probabilistic framework used by EM.

Whereas previously the segmentation assigned each pixel to one particular link of the model, in a probabilistic framework, the segmentation now determines the likelihood of each pixel belonging to a link m. Segmentation happens in the E-step, and consists of a modified version of Equation 1:

$$g_m(x,y) = \frac{p_m e^{-(G_w * R_m^2(\mathbf{P}))(x,y)/\sigma^2}}{\sum_i p_i e^{-(G_w * R_i^2(\mathbf{P}))(x,y)/\sigma^2}}$$

where $*$ denotes image convolution, and G_w denotes a Gaussian smoothing kernel of width w. The difference between this equation and Equation 1 is this convolution, which smooths the segmentation maps; other techniques have been explored in [19]. The p_m terms denote prior probabilities for each link m. These can be set in proportion to the expected sizes of each of the links in the images.

Motion analysis, which happens in the M-step, consists of finding a set of parameters to minimize an error function. Previously, the error function was simply the sum of squared residuals. Now it must take into account the account the likelihood of a pixel belonging to a link. This is expressed in a modified version of Equation 2:

$$\mathbf{P} = \underset{\mathbf{P}}{\operatorname{argmin}} \sum_m \sum_{(x,y)} g_m(x,y) R_m^2(\mathbf{P}, x, y)$$

In Equation 2, each pixel contributed to the total energy function once, via the R_m function associated with the link m that the pixel belonged to. Now, each pixel will contribute several times, once for each of the links of the model. This is because the models are no longer independent—they share parameters and must be optimized simultaneously.

Finally, we describe the overall algorithm. We first initialize the segmentation estimates to give each pixel the same probability of belonging to any of the models. The parameters $\mathbf{P}$ are initialized to zero. Then the M-step is applied, to update the parameter values $\mathbf{P}$. Recall that computing the parameters is actually an iterative process; we one perform one iteration per M-step. Then the E-step is applied to update the segmentation, and E- and M-steps are alternated from then on until the motion estimates converge.

4. Experimental Results

We first present the articulated model used in our experiments, followed by applications of the articulated EM algorithm to a synthetic and a real test sequence.

4.1. Specific Articulated Model

For our initial experiments, we used a simple two dimensional model of a person's arm, shown in Figure 1a. In this model, the torso can translate in any direction, the upper arm rotates about the shoulder, and the lower arm rotates about the elbow. The attachment points of the two links are specified. For the upper arm, this requires the (x_0, y_0) parameters, while for the lower arm, the length L and joint angle θ_1 of the upper arm are needed.

The motions of the links in this model are straightforward to compute. Below are the equations for the deformation functions $\mathbf{F}_m(\cdot)$, which give the new position of a pixel in the image, after the motions described by parameters $\mathbf{P} = \{\Delta x_0, \Delta y_0, \theta_1, \Delta\theta_1, \Delta\theta_2\}$. The motions will also depend on the other model parameters, x_0, y_0, L which are assumed to be fixed and known. $\mathrm{Rot}(\alpha, \mathbf{c}, \mathbf{x})$ denotes a rotation by angle α of point $\mathbf{x}$ about a center of rotation $\mathbf{c}$.

$$\mathbf{F}_{\text{background}}(\mathbf{P}, \mathbf{x}) = \mathbf{x}$$

$$\mathbf{F}_{\text{torso}}(\mathbf{P}, \mathbf{x}) = \mathbf{F}_{\text{background}}(\mathbf{P}, \mathbf{x} + \mathbf{x}_0)$$

$$\mathbf{F}_{\text{upper}}(\mathbf{P}, \mathbf{x}) = \mathbf{F}_{\text{torso}}(\mathbf{P}, \mathrm{Rot}(\Delta\theta_1, \mathbf{x}_0, \mathbf{x}))$$

$$\mathbf{F}_{\text{lower}}(\mathbf{P}, \mathbf{x}) = \mathbf{F}_{\text{upper}}(\mathbf{P}, \mathrm{Rot}(\Delta\theta_2, \mathbf{x}_0 + L \begin{pmatrix} \cos\theta_1 \\ \sin\theta_1 \end{pmatrix}, \mathbf{x}))$$

These equations, and their derivatives with respect to the parameters Δx_0, Δy_0, θ_1, $\Delta\theta_1$, and $\Delta\theta_2$, are used in the articulated EM algorithm.

Finally, we must set the prior probabilities associated with each link of the model:

$$p_{\text{background}} = 1 \quad p_{\text{torso}} = 1/2$$
$$p_{\text{upper}} = 1/3 \quad p_{\text{lower}} = 1/4$$

This causes a small bias towards explaining the pixels by links higher in the kinematic model hierarchy.

4.2. Experiments

We performed two experiments using motion sequences which fit the two link model described above. The first used synthetic data, to measure the accuracy of the algorithm, while the second used two frames from a real image sequence, to verify the algorithm's performance on real data. More details on these results are presented in [14].

The first sequence, depicted in Figures 1b and c, consisted of synthetic data. Each part of the model was textured with Gaussian white noise. The motion and segmentation maps derived for the torso and upper and lower arm are shown in Figures 2 and 3. The background motion was fixed at zero, so it is correct by definition. The torso's estimated motion was $(0.004, 0.015)$ pixels, which is close to the correct value of zero. The upper arm rotated an estimated $14.6°$ clockwise, which is close to the correct value of $15.0°$. The motion field for the lower arm required the estimation of two parameters: the angular position of the upper arm $(35.2°)$ and the amount of rotation of the lower

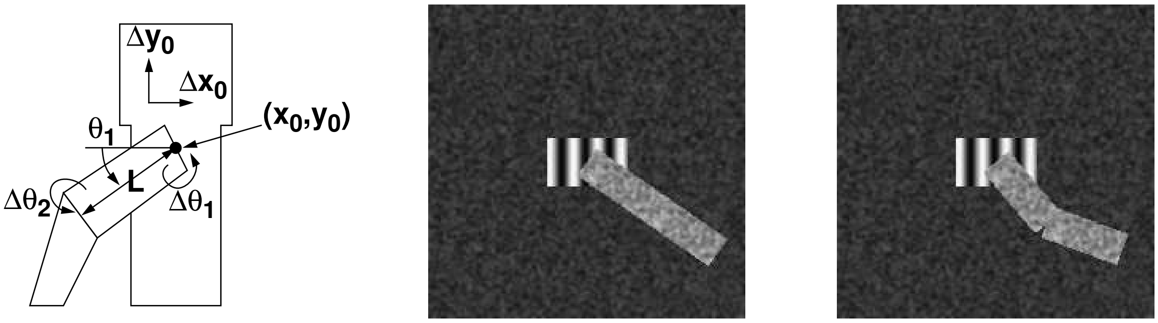

Figure 1. (a) is the two dimensional articulated model used for all of the experiments in this paper. (b) and (c) show a two frames in a synthetic motion sequence.

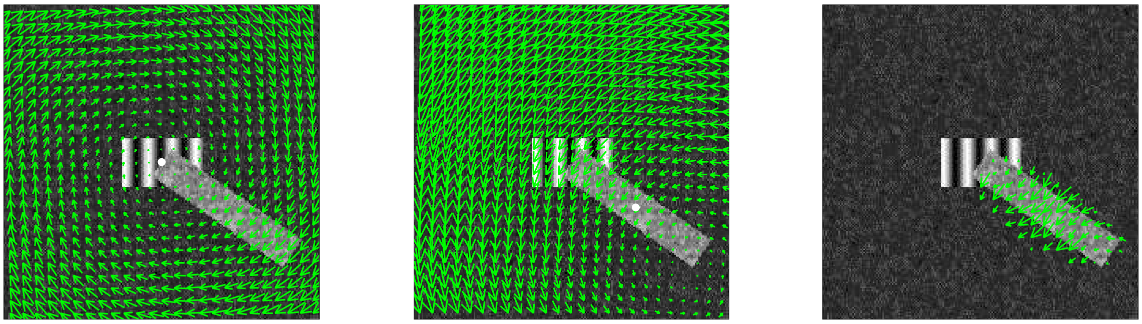

Figure 2. Estimated flow fields for the first link (a) and second link (b) models, with a white dot indicating the center of rotation of each link. (c) shows the composite optical flow estimate.

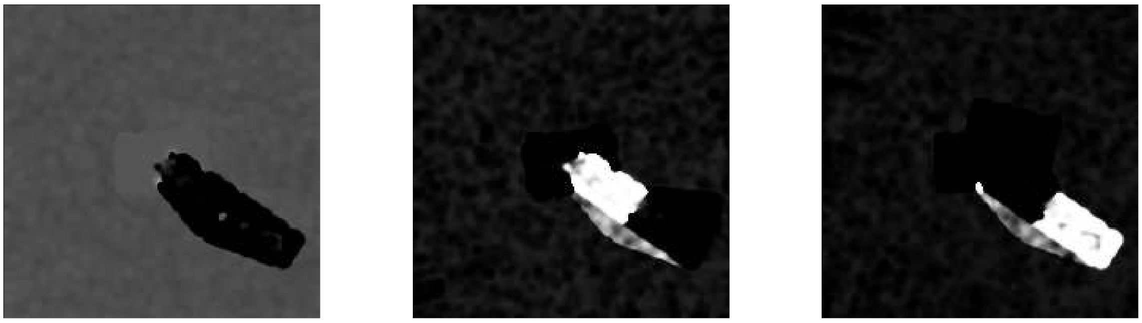

Figure 3. Segmentation maps for the base, first link, and second link.

arm ($29.6°$ counterclockwise), both of which are close to the correct values of $35.0°$ and $30.0°$, respectively.

The segmentation maps (Figure 3) are coded so that white means the pixel belongs to that link with probability one, while black means probability zero. The background and torso were both assigned to the background part of the model, because both the background and torso have zero motion, and the system has a bias towards using the background model. The upper and lower arms were segmented well, although the segmentation maps these parts also include the region that is occluded by the arm's motion.

The second experiment used two frames from a sequence in which the first author moved his arm in a clockwise direction in the image. As can be seen from the images in Figure 4, the upper arm rotated less than the lower arm. The results of applying the articulated EM algorithm to this data are depicted in Figures 5 and 6. As with the previous example, the torso and background were both stationary, so they are assigned to the background part of the model. The upper arm was segmented fairly well, although some pixels from below the elbow and the occluded region were also assigned to this model. The lower arm was segmented quite accurately. The motion estimates seem reasonably accurate, although ground truth measurements are not available. The easiest parameter to evaluate is θ_1, whose accuracy can be seen by the predicted location of the elbow.

5. Previous Work

The present work is motivated by the desire to find a motion representation which is "tuned" to the properties of articulated objects, and which can be extracted from a wide

Figure 4. A pair of adjacent frames from a real motion sequence.

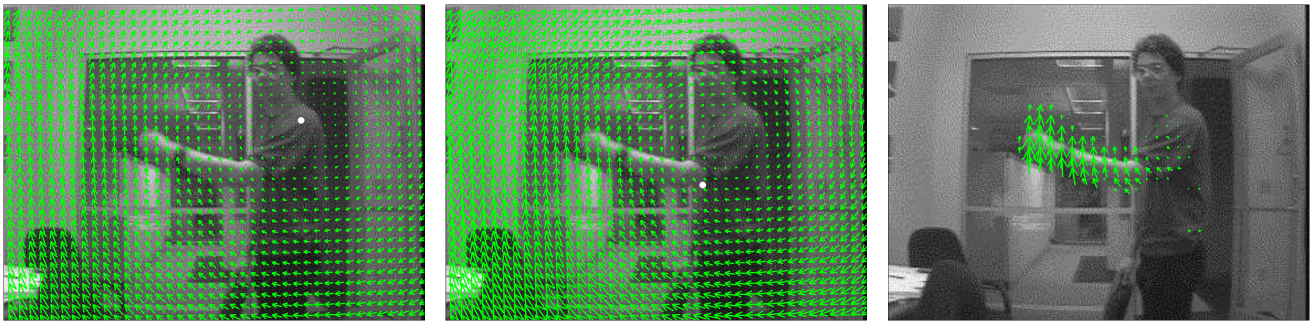

Figure 5. Estimated optical flow fields for the upper arm (a), and lower arm (b) models, with a white dot indicating the center of rotation of each link. (c) shows the composite optical flow estimate.

Figure 6. Segmentation maps for the torso, upper arm, and lower arm.

range of image sequences. Such a representation could provide a basis for the classification and recognition of human motion. We will describe a representative set of previous work on human motion analysis; please refer to [14] for a more comprehensive discussion.

Several authors have addressed the segmentation of repetitive human motion from video. In [10], time-frequency analysis on translation-stabilized imagery is used to detect repetitive patterns due to gait and arm swinging. Periodicity and a known ground plane orientation are exploited in [9] to obtain an initial segmentation of fronto-parallel walking motion, which is then refined by fitting spatiotemporal surface models. In [16], a simple articulated model is applied to previously segmented motion data, while in [7] a deformable articulated model is segmented from carefully staged figure motion. A nice property of these approaches is that they can exploit many frames of video in making a decision. However, their dependency on specific types of motion or viewing directions is an obstacle to general video analysis.

Previous work on gesture recognition employed temporal Markov chain models to segment and detect image feature motion that is consistent with a vocabulary of human action [3, 20]. Related work in [2] introduces the concept of motion-history images for specific actions. These efforts make use of domain-specific, high-level models in an attempt to bypass the need for accurate low-level image analysis. However, they may be difficult to apply in a broad domain such as video indexing, where the number of possible actions is quite large.

Our algorithm for articulated motion segmentation builds heavily on previous work on segmenting multiple rigid motions using EM. Following [6], a number of authors have applied mixture models to motion analysis. For example, [19] presents a general framework for EM-based motion analysis that includes spatial coherence. We extend these approaches by demonstrating how to incorporate kinematic constraints into the EM analysis.

6. Conclusions and Future Work

We have presented an EM algorithm which uses an articulated kinematic model to segment and estimate human motion. Use of kinematic constraints makes it possible to describe complex human motions with a much smaller number of parameters than local flow or rigid motion models would require. This should improve the accuracy and noise robustness of the result. Furthermore, the set of motion models provide a parts decomposition of the motion field. Experimental results with synthetic data have shown that the algorithm is accurate, and results on real images demonstrate the algorithm's robustness.

There are a number of extensions to our algorithm that could improve its accuracy and robustness. A current limitation is the use of only two frames from a motion sequence, which may result in ambiguities in assigning pixels to motion models. It would be interesting to extend the approach to multiple frames, potentially improving the accuracy. Another difficulty is that occluded pixels can adversely affect the motion estimation. This problem will become more severe when many frames are used. It would be interesting to add the segmentation of occluded pixels to the model. Finally, we are interested in extending the complexity of our current kinematic model to encompass more of the degrees of freedom in human motion. This would make it possible to apply the results of articulated motion analysis to detecting human motion and recognizing actions.

References

[1] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *2nd Europ. Conf. on Computer Vision*, pages 237–252, Italy, 1992.

[2] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. In *3rd Workshop on Appl. of Computer Vision*, pages 39–42, Sarasota, FL, December 1996.

[3] C. Bregler, S. Omohundro, et al. Probabilistic models of verbal and body gesture. In S. Pentland and R. Cipolla, editors, *Computer Vision in Man-Machine Interfaces*. Cambridge University Press, Cambridge, UK, 1997.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

[5] M. Fleck, D. Forsyth, and C. Bregler. Finding naked people. In *4th Europ. Conf. on Computer Vision*, pages 593–602, Cambridge, UK, April 1996.

[6] A. Jepson and M. Black. Mixture models for optical flow computation. In *Computer Vision and Pattern Recognition*, pages 760–761, New York City, NY, June 1993.

[7] I. Kakadiaris and D. Metaxas. 3d human body model acquisition from multiple views. In *5th Intl. Conf. on Computer Vision*, pages 618–623, Cambridge, MA, June 1995.

[8] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo. In *7th Int. Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver, B.C., 1981.

[9] S. Niyogi and E. Adelson. Analyzing and recognizing walking figures in XYT. In *Computer Vision and Pattern Recognition*, pages 469–474, Seattle, WA, June 1994.

[10] R. Polana and R. Nelson. Nonparametric recognition of nonrigid motion. Technical Report TR 575, Univ. of Rochester Dept. of Computer Science, March 1995.

[11] R. Redner and H. Walker. Mixture densities, maximum likelihood, and the EM algorithm. *SIAM Revew*, 26:195–239, 1994.

[12] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *5th Intl. Conf. on Computer Vision*, pages 612–617, Boston, MA, 1995.

[13] J. Rehg, M. Loughlin, and K. Waters. Vision for a smart kiosk. In *Computer Vision and Pattern Recognition*, San Juan, PR, June 1997. In this proceedings.

[14] J. Rehg and H. Rowley. An EM algorithm for articulated motion analysis. Technical Report CRL 96/3, Digital Equipment Corp. Cambridge Research Lab, 1996.

[15] J. Rehg and A. Witkin. Visual tracking with deformation models. In *Proc. of Intl. Conf. on Robotics and Automation*, pages 844–850, Sacramento, CA, April 1991.

[16] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1994.

[17] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *Computer Vision and Pattern Recognition*, pages 203–208, San Francisco, CA, June 1996.

[18] M. Spong. *Robot Dynamics and Control*. John Wiley and Sons, 1989.

[19] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *Computer Vision and Pattern Recognition*, pages 321–326, San Fransisco, CA, June 1996.

[20] A. Wilson and A. Bobick. Learning visual behavior for gesture analysis. In *Int. Symposium on Computer Vision*, pages 229–234, Coral Gables, FL, November 1995.