Multimodal Shape Tracking with Point Distribution Models

J. Giebel and D.M. Gavrila

Machine Perception, DaimlerChrysler Research,
Wilhelm Runge Str. 11, 89089 Ulm, Germany
{jan.giebel, dariu.gavrila}@daimlerchrysler.com

Abstract. This paper addresses the problem of multimodal shape-based object tracking with learned spatio-temporal representations. Multimodality is considered both in terms of shape representation and in terms of state propagation. Shape representation involves a set of distinct linear subspace models or Point Distribution Models (PDMs) which correspond to clusters of similar shapes. This representation is learned fully automatically from training data, without requiring prior feature correspondence. Multimodality at the state propagation level is achieved by particle filtering. The tracker uses a mixed-state: continuous parameters describe rigid transformations and shape variations within a PDM whereas a discrete parameter covers the PDM membership; discontinuous shape changes are modeled as transitions between discrete states of a Markov model. The observation density is derived from a well-behaved matching criterion involving multi-feature distance transforms. We illustrate our approach on pedestrian tracking from a moving vehicle.

1 Introduction

For many real world tracking applications there are no explicit prior models available to account for object appearance and motion. This paper presents a technique to learn spatio-temporal shape models for complex deformable objects from examples. See Figure 1. To capture the shape variation we derive a set of distinct object parameterizations, corresponding to clusters of similar shapes, based on the integrated registration and clustering approach introduced in [7]. For compactness a linear subspace decomposition reduces the dimensionality in each cluster. To constrain the temporal changes in shape, these object parameterizations are treated as discrete states in a Markov model. The transition probabilities for such a model can be derived from training sequences.

Tracking is performed using an adaption [9, 12] of the stochastic framework ("Condensation") proposed by Isard and Blake [13], which can cope with the mixed continuous/discrete state space of a spatio-temporal shape model. Due to the stochastic nature and the ability to approximate multimodal probability density functions, the algorithm is quite robust against cluttered backgrounds and partial occlusions. The states of our tracker are propagated over time by applying random noise assuming constant velocity of the 2D movement of the

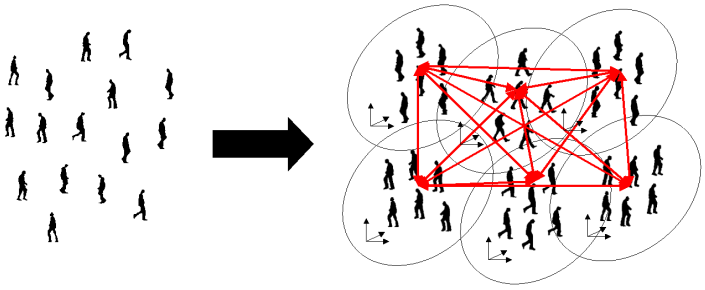

Fig. 1. Acquiring the proposed spatio-temporal shape representation

object. The observation density is computed from a matching criteria based on multi-feature distance transforms, which has previously been successfully applied to object detection [6].

The paper is organized as follows. Section 2 discusses previous work. Section 3 describes how the proposed spatio-temporal model can be learned from examples. Tracking with this representation is discussed in Section 4. Preliminary experiments are described in Section 5. Finally, the paper is concluded in Section 6.

2 Previous Work

Several representations have been proposed to cover the distribution of object appearance and motion.

Compact low-dimensional object parameterizations can be obtained by linear subspace techniques based on shape (PDMs) [4], texture [2, 18], motion, or combinations [1, 5] and are widely used in the computer vision community. However, these methods have some limitations. One concerns the global linearity assumption: nonlinear object deformations have to be approximated by linear combinations of the modes of variation. Therefore linear subspace models are not the most compact representations for objects undergoing complex (nonlinear) deformations. They are also not specific, since implausible shapes can be generated, when invalid combinations of the modes are used.

To treat the problem of linearity, Bregler and Omohundro [3] developed a method to learn nonlinear surfaces from examples. The object manifold is approximated by the union of a set of locally-linear surface patches. Heap and Hogg used this idea to build their "Hierarchical Point Distribution Model" [8], which they extended to deal with discontinuous changes in shape for tracking [9].

Stochastic frameworks for object tracking are also frequently discussed in the computer vision literature. In particular particle filtering ("Condensation") has been popular recently [9, 13, 16]. It is conceptually simple and more general than the classical Kalman filter techniques since it does not rely on Gaussian noise or linearity assumptions and can even be applied when no closed-form

solutions of the posterior are available. The algorithm simultaneously considers multiple hypothesis (without explicitly solving the correspondence problem), which makes it robust to escape from local maxima of the estimated probability density function. Several extensions to the original implementation have been proposed, for example to achieve real-time performance [11], to cope with high dimensional state spaces [16], to deal with mixed discrete/continuous state spaces [12], and multiple targets [14, 17]. A rich variety of stochastic dynamical models were used in combination with particle filtering. In [15] the complex dynamics of an object are decomposed into several motion classes. The motion in each class is modeled by an auto-regressive process, while the class transitions are treated in Markov fashion. Unfortunately, it is not always straightforward to find distinct motion classes for complex objects.

Our approach, discussed in the next sections, builds upon previous work of Heap and Hogg [9] and is closely related to Isard and Blake [12]. We extended this work to deal with our spatio-temporal shape representation, which does not utilize a common object parameterization for all possible shapes. Instead a set of unconnected local parameterizations is used, which correspond to clusters of similar shapes. The learning scheme is more general since it does not assume prior feature correspondence among the training shapes. In contrast to single object parameterizations [4, 12] an improved specificity of the model can be expected. During tracking we explicitly model the cluster membership as a discrete state of our samples. The observation density used in our experiments is based on multi-feature distance transforms. We demonstrate our system on difficult real world data, taken from a moving vehicle.

3 Spatio-temporal Shape Representation

This section describes how the spatio-temporal shape representation is obtained from examples. The algorithm passes a sequence of three successive steps. At first our integrated registration and clustering approach [7] partitions the shapes, establishes point correspondence between similar shapes, and aligns them with respect to similarity transform. For compactness, a separate linear subspace decomposition reduces the dimensionality in each cluster. Finally, the transition probabilities between the different parameterizations are determined for the Markov model. The learning scheme is illustrated in Figure 1. The shape distribution is represented by a set of linear subspace models, corresponding to clusters of similar shapes, with transition probabilities between them.

The next two paragraphs review the integrated registration and clustering approach, which is discussed in detail in [7].

Registration, which brings the points of two shapes into correspondence and aligns them with respect to similarity transform (ie. translation, scale and rotation), proceeds as follows. At first the shapes, represented by the sequence of their n x- and y-coordinates $\boldsymbol{s} = (x_1, y_1, x_2, y_2, ..., x_n, y_n)$, are decomposed into segments locating the extrema of the curvature function. The segments on both shapes are characterized by their transformation invariant Fourier descriptors.

Since the ordering of the segments is known on the shapes, the optimal solution to the correspondence problem between them can be found by dynamic programming, using weighted Euclidean metrics on the low order Fourier coefficients as similarity measure. Dense point correspondence can then be derived by interpolation between the corresponding segments. Finally the shapes are aligned using a least squares fit [4].

The cluster algorithm has a k-means flavor and simultaneously embeds similar shapes into a common feature space. Iteratively a shape is chosen and registered to all existing prototypes. If the alignment error to the best matching prototype is below a user defined threshold, then the shape is assigned to the particular cluster and the corresponding prototype is updated to be the mean of the shape vectors inside this cluster. Otherwise, a new cluster is created with the chosen shape as prototype.

After the registration and clustering step we apply a principal component analysis in each cluster of registered shapes to obtain compact shape parameterizations known as "Point Distribution Models" [4]. From the N example shapes $\boldsymbol{s_i}$ of each cluster the mean shape $\bar{\boldsymbol{s}}$ is calculated. The covariance matrix K is derived from the deviations of the mean

$$K = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{s_i} - \bar{\boldsymbol{s}})(\boldsymbol{s_i} - \bar{\boldsymbol{s}})^T. \tag{1}$$

By solving the eigensystem $K\boldsymbol{e_i} = \lambda_i \boldsymbol{e_i}$ one obtains the $2n$ orthonormal eigenvectors, corresponding to the "modes of variation". The k most significant "variation vectors" $E = (\boldsymbol{e_1 e_2} \ldots \boldsymbol{e_k})$ with the highest eigenvalues λ_i are chosen to capture a user-specified proportion of total variance contained in the cluster. Shapes can then be generated from the mean shape plus a weighted combination of the variation vectors $\tilde{\boldsymbol{s}} = \bar{\boldsymbol{s}} + E\boldsymbol{b}$. To ensure that the generated shapes are not outliers, we constrain the weight vector $\boldsymbol{b}$ to lie in a hyperellipsoid about the origin. Therefore $\boldsymbol{b}$ is scaled such that the weighted distance from the mean is less than a user-supplied threshold M_{max} [4]

$$\sum_{i=1}^{k} \frac{b_i^2}{\lambda_i} \leq M_{max}^2. \tag{2}$$

Finally, the transition probabilities between the different subspace models are determined and stored in a Markov state transition matrix T. An entry $T_{i,j}$ represents the probability of a discrete state transition from cluster i to j.

4 Tracking

Multimodal tracking is performed via an adaption of the "Condensation" algorithm [13], which can deal with "mixed" discrete/continuous state spaces. The "Condensation" algorithm approximates the probability density function $p(\boldsymbol{x_t}|\mathcal{Z}_t)$ of the object's configuration $\boldsymbol{x_t}$ at times t conditioned by the observations $\mathcal{Z}_t = \{\boldsymbol{z_t}, \boldsymbol{z_{t-1}}, ..., \boldsymbol{z_1}\}$ by a set of weighted samples. At each iteration the

samples are predicted with a stochastic dynamical model $p(\boldsymbol{x_t}|\mathcal{X}_{t-1})$ over time, where $\mathcal{X}_{t-1} = \{\boldsymbol{x_{t-1}}, \boldsymbol{x_{t-2}}, ..., \boldsymbol{x_1}\}$. Usually the Markov-assumption is made so that $p(\boldsymbol{x_t}|\mathcal{X}_{t-1})$ only depends on a predefined number of prior states, the order of the Markov model. When new measurements are available the samples are weighted according to an observation model $p(\boldsymbol{z_t}|\boldsymbol{x_t})$. Proportional to these weights, they are chosen to approximate the prior for the next iteration using factored sampling.

In our implementation, the state vector $\boldsymbol{x} = (\boldsymbol{c}, d)$ of each sample consists of a discrete parameter d modeling the PDM membership and continuous parameters $\boldsymbol{c}$ corresponding to object translation, scale, rotation (similarity transform) velocity and the PDM shape parameters. Because the PDMs usually utilize a different number of parameters, the size of the state vector may vary in time.

The dynamical model of the tracker is decomposed, to account for discontinuous shape changes, corresponding to PDM transitions, during tracking [12]. They occur according to the transition probabilities $T_{i,j}$ of our spatio-temporal shape model. The decomposition is as follows:

$$p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}) = p(\boldsymbol{c_t}|d_t, \boldsymbol{x_{t-1}})p(d_t|\boldsymbol{x_{t-1}}). \tag{3}$$

Since the transition probabilities $T_{i,j}$ of our shape model are assumed to be independent of the previous continuous shape and transformation parameters $\boldsymbol{c_{t-1}}$

$$p(d_t = i|\boldsymbol{c_{t-1}}, d_{t-1} = j) = T_{i,j}(\boldsymbol{c_{t-1}}) = T_{i,j}. \tag{4}$$

In the case of $i = j$, when no PDM transition occurs, we assume

$$p(\boldsymbol{c_t}|d_t = j, d_{t-1} = i, \boldsymbol{c_{t-1}}) = p_{i,j}(\boldsymbol{c_t}|\boldsymbol{c_{t-1}}) \tag{5}$$

to be a Gaussian random walk. For $i \neq j$ the PDM membership is switched from i to j. In this case the transformation parameters are maintained with random displacement, while the shape parameters are assumed to be normally distributed about the mean shape of PDM j.

Our observation density, determining the "goodness" of a sample, is based on multi-feature distance transforms [6]. As features we use the position of directed edges in the experiments. If the image I is considered the observation $\boldsymbol{z_t}$ at time t and S is the projection of the shape parameters $\boldsymbol{x_t}$ into the image we assume that [16]

$$\log p(\boldsymbol{z_t}|\boldsymbol{x_t}) \equiv \log p(I|S) \propto -\frac{1}{|S|}\sum_{s \in S} D_I(s), \tag{6}$$

where $|S|$ denotes the number of features s in S and $D_I(s)$ is the distance of the closest feature in I to s. At this point we iterate the following algorithm [12]:

From the prior sample set $\{s_{t-1}^{(n)}, \pi_{t-1}^{(n)},\ n \in \{1, .., N\}\}$ at time $t-1$ the n^{th} sample $s_t^{(n)}$ with weight $\pi_t^{(n)}$ at time t is derived as follows to yield the posterior sample set $\{s_t^{(n)}, \pi_t^{(n)},\ n \in \{1, .., N\}\}$:

Select a sample j of the prior population with probability $\pi_{t-1}^{(j)}$ and insert it into the new population $s_t'^{(n)} = s_{t-1}^{(j)}$

Predict (by sampling from $p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}} = s_t'^{(n)})$ to find $s_t^{(n)}$)

- the transformation parameters assuming a Gaussian random walk and constant velocity.
- the discrete shape parameter by sampling the transition probabilities $T_{i,j}$. If $s_t'^{(n)}$ is in cluster a, this is done by generating a random number $r \in \{0, ..., 1\}$ and choosing the smallest b such that $C_{a,b} > r$, where $C_{r,c} = \sum_{i=1}^{c} T_{r,i}$ and r and c index the rows and columns of T and C.
- the continuous shape parameters. If $a = b$ the old parameters are maintained with random displacement, otherwise they are assumed to be normally distributed about the cluster mean of b.

Weight according to the observation density $\pi_t^{(n)} = p(\boldsymbol{z_t}|\boldsymbol{x_t} = s_t^{(n)})$.
Finally, the weights are normalized such that $\sum_n \pi_t^{(n)} = 1$.

5 Experiments

To demonstrate our system we performed preliminary experiments on tracking pedestrians from a moving vehicle. The different linear subspace models of the proposed spatio-temporal model were automatically learned from 500 pedestrian shapes of our database following the method described in section 3. Figure 2

Fig. 2. Varying the first mode of variation for three different clusters between $\pm 2\sigma$

illustrates the changes in shape while varying the first mode of variation for three different clusters between ± 2 standard deviations σ. For the moment the transition probabilities between the different subspace models were set to equal values. We are in the process of compiling them automatically from a large database of pedestrian sequences including thousands of images.

About 1000 samples per cluster were used to track the pedestrians in Figure 3 with that model. In both sequences we show the "modal" shape of each track in black or white, the one with the highest sample weight according to equation 6. Note that because of the discrete parameter in our state space and the different object parameterizations it is no longer possible to evaluate the mean properties of the posterior directly from the samples as in [10]. To illustrate the estimated probability density function of the first sequence we additionally show the sample set projected onto the edge image in black (second and fourth row of Figure 3). To initialize the tracks we generated a random population of shapes about a given starting position. The first part of the upper sequence is particularly difficult for the tracker (first row of Figure 3), because of the low

contrast between the upper part of the body and the background. One observes that the sample set (approximating the probability density function) splits while the pedestrian passes the bushes behind him. The ambiguity is solved, because the "correct" mode of the estimated probability function dominates according to our observation density in time. In the second sequence the results of two different tracks are displayed. Although the scene is quite complex, the trackers correctly keep lock on the desired targets.

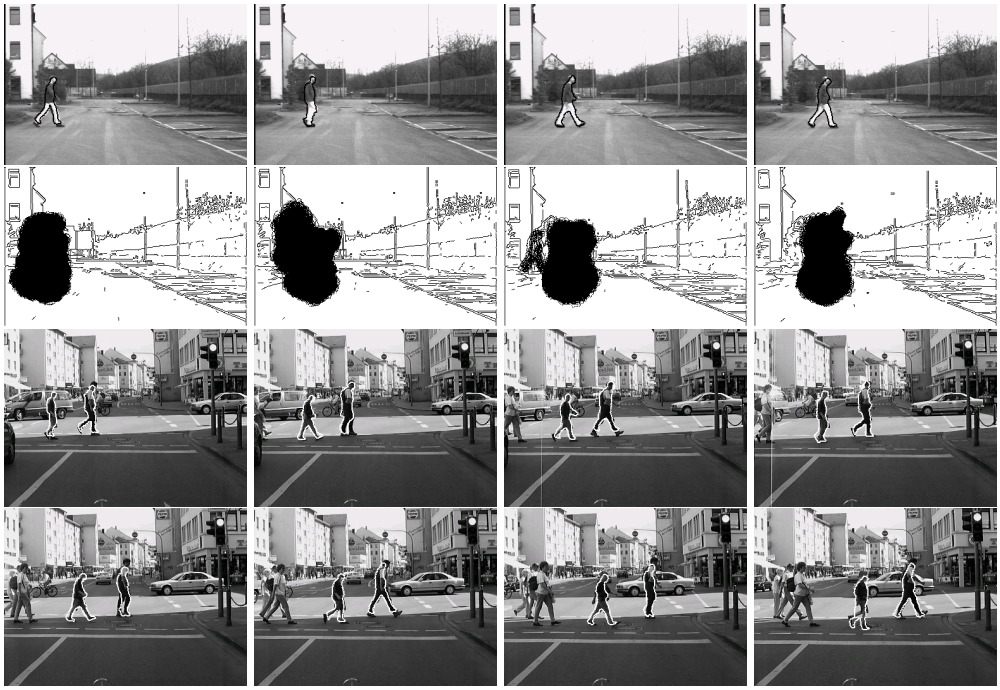

Fig. 3. Tracking results

6 Conclusions

This paper presented a general method to learn a spatio-temporal shape model from examples and showed how it can be used for tracking in a stochastic framework. A set of linear parameterizations was automatically learned from our training set to represent the shape distribution. A Markov model was applied to constrain the temporal changes in shape over time. For tracking we used an adaption of the "Condensation" algorithm, which can cope with mixed discrete/continuous state spaces. We obtained quite promising results on difficult image data using our proposed multi-modal shape tracking technique. Work in

progress involves using more efficient sampling methods (we desire realtime performance), combining the tracker with hierarchical shape-based object detection using distance transforms, and testing our approach on a large database with several thousands of pedestrian images with given ground truth data.

References

1. A. Baumberg and D. Hogg. Learning flexible models from image sequences. *Proc. of the ECCV*, pages 299–308, 1999.
2. M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. of Computer Vision*, 26(1):63–84, January 1998.
3. C. Bregler and S.M. Omohundro. Surface learning with applications to lipreading. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 43–50. Morgan Kaufmann Publishers, Inc., 1994.
4. T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. *CVIU*, pages 38–59, 1995.
5. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Trans. on PAMI*, 23(6):681–684, June 2001.
6. D. M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Proc. of the ICPR*, pages 439–444, Brisbane, 1998.
7. D. M. Gavrila, J. Giebel, and H. Neumann. Learning shape models from examples. In *Proc. of the Deutsche Arbeitsgemeinschaft für Mustererkennung*, pages pp. 369–376, Munich, Germany, 2001.
8. T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In Adrian F. Clark, editor, *British Machine Vision Conference*, 1997.
9. T. Heap and D. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. of the ICCV*, pages 344–349, 1998.
10. M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV96*, pages I:343–356, 1996.
11. M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of the ECCV*, pages I:893–908, 1998.
12. M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *Proc. of the ICCV*, pages 107–112, 1998.
13. Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of the ECCV*, pages 343–356, 1996.
14. E.B. Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, pages 93–105, 2001.
15. B. North, A. Blake, M Isard, and J. Rittscher. Learning and classification of complex dynamics. *PAMI*, 22(8):781–796, August 2000.
16. V. Philomin, R. Duraiswami, and L.S. Davis. Quasi-random sampling for condensation. In *Proc. of the ECCV*, pages 134–149, 2000.
17. D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving objects with a mobile robot. In *Proc. of the IEEE CVPR Conf.*, 2001.
18. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.