International Journal of Computer Vision 48(1), 9–19, 2002

Probabilistic Tracking with Exemplars in a Metric Space

KENTARO TOYAMA
Microsoft Research,[*] Redmond, WA USA
kentoy@microsoft.com

ANDREW BLAKE
Microsoft Research Ltd., Cambridge, UK
ablake@microsoft.com

Abstract. A new, exemplar-based, probabilistic paradigm for visual tracking is presented. Probabilistic mechanisms are attractive because they handle fusion of information, especially temporal fusion, in a principled manner. *Exemplars* are selected representatives of raw training data, used here to represent probabilistic mixture distributions of object configurations. Their use avoids tedious hand-construction of object models, and problems with changes of topology.

Using exemplars in place of a parameterized model poses several challenges, addressed here with what we call the "Metric Mixture" (M^2) approach, which has a number of attractions. Principally, it provides alternatives to standard learning algorithms by allowing the use of metrics that are not embedded in a vector space. Secondly, it uses a noise model that is learned from training data. Lastly, it eliminates any need for an assumption of probabilistic pixelwise independence.

Experiments demonstrate the effectiveness of the M^2 model in two domains: tracking walking people using "chamfer" distances on binary edge images, and tracking mouth movements by means of a shuffle distance.

Keywords: probabilistic tracking, exemplar-based tracking

1. Introduction

There is, of course, a substantial literature on tracking, driven either by image features (Amini et al., 1988; Kass et al., 1987) or by raw intensity (Bascle and Deriche, 1995; Black and Jepson, 1996; Hager and Toyama, 1996), or both (Cootes et al., 1998). Tracking can be formulated in a probabilistic framework in both the feature-driven (Terzopoulos and Szeliski, 1992) and intensity-driven (Storvik, 1994) settings. The probabilistic formulation has the attraction that uncertainty is handled in a systematic fashion, allowing principled handling of sensor fusion and temporal fusion. Many such tracking algorithms, however, demand that complex models be defined and trained for each object class to be tracked—a process that is often laborious and difficult to automate fully.

Our aim, therefore, is to develop a paradigm which retains the probabilistic setting while avoiding the use of explicit models to describe target objects. The use of *exemplars*, for example, the contour exemplars in Fig. 1, offers an alternative that can tackle this problem (Brand, 1999; Efros and Leung, 1999; Freeman and Pasztor, 1999; Frey and Jojic, 2000; Gavrila and Philomin, 1999). Exemplar-based models can be constructed very directly from training sets, without the need to set up complex intermediate representations, such as parameterized contour models or 3D articulated models.

Existing tracking algorithms that use exemplar-based models have certain limitations. Single-frame exemplar-based tracking (Gavrila and Philomin, 1999),

[*] http://research.microsoft.com/vision.

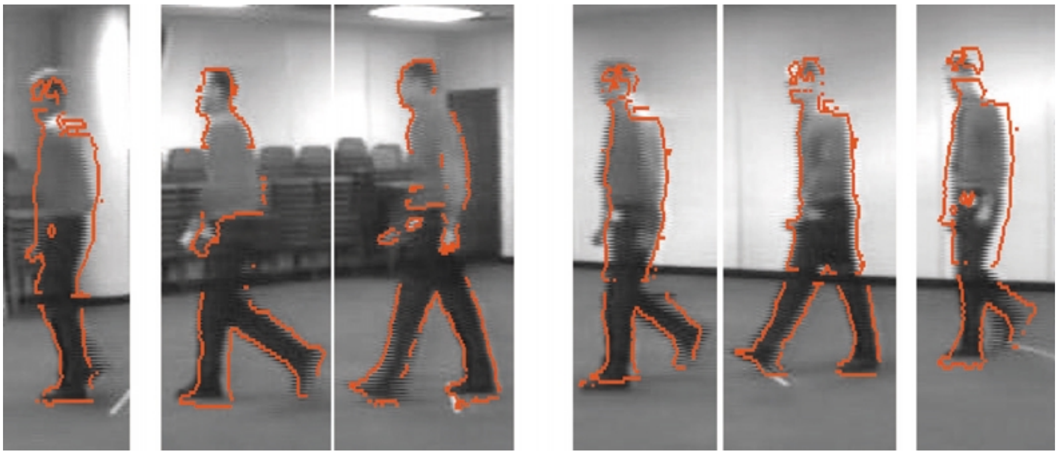

Figure 1. Cropped, sample frames from a tracked test sequence. The overlays represent the maximum *a posteriori* exemplars. Exemplars and dynamics were learned from an independent training sequence of the same individual walking along a similar path (see also the video `walk1.mpg`, viewable at `http://research.microsoft.com/vision/papers`).

though effective, is limited by its inability to incorporate temporal constraints, resulting in jerky recovered motion and reduced power to recover from occlusion. Full temporal tracking can be obtained via Kalman filtering or particle filtering, for which a probabilistic framework is needed. Frey and Jojic (2000) have demonstrated elegantly how exemplars can be embedded in learned probabilistic models by treating them as centers in probabilistic mixtures. Their motion-sequence analysis is, in principle, fully automated, requiring only the structural form of a generative image-sequence model to be specified in advance. However, the approach has serious drawbacks:

- inference is done with online expectation-maximization (EM), which is computation intensive and limited, for practical purposes, to low resolution images;
- images have to be represented simply as arrays of pixels, ruling out nonlinear transformations that can help with invariance to scene conditions, including the conversion of images to edge maps that proves so powerful with non-probabilistic exemplars (Gavrila and Philomin, 1999);
- finally, image noise is treated as white despite known, strong statistical correlations between pixels (Field, 1987).

The problem, therefore, is to combine exemplars in a metric space (Gavrila and Philomin, 1999) with a probabilistic treatment (Frey and Jojic, 2000), retaining the best features of each approach. Unfortunately, this combination is not straightforward. The very techniques which make probabilistic treatment possible (i.e., modeling with Gaussians, PCA, k-means, EM, etc.), are not applicable given that exemplar-based models need have no vector-space structure. (There is no clear sense in which two of the outline contours in Fig. 1 can be added together.) We propose the *Metric Mixture* (M^2) model, described below, to solve this problem. Figure 1 shows the approach applied to tracking a walking person.

One note on terminology: the theory and algorithms could be presented as for true metrics. A function ρ is a metric when (1) $\rho(a, b) \geq 0$, $\forall a, b$, (2) $\rho(a, b) = 0$ iff $a = b$, (3) $\rho(a, b) = \rho(b, a)$, and (4) $\rho(a, b) + \rho(b, c) \geq \rho(a, c)$. The M^2 theory, however, can also apply to certain functions without axioms (3) and (4). We will refer to these latter functions as "distance functions."

2. Pattern-Theoretic Tracking

Test image sequences $\mathcal{Z} = \{z_1, \ldots, z_T\}$ are to be analyzed in terms of a probabilistic model learned from a training image sequence $\mathcal{Z}^* = \{z_1^*, \ldots, z_{T^*}^*\}$. Images may be preprocessed for ease of analysis, for example by filtering to produce an intensity image with certain features (e.g., ridges) enhanced, or nonlinearly filtered to produce a sparse binary image with feature pixels marked. A given image z is to be approximated, in a standard pattern theoretic manner (Mumford, 1996), as an ideal image or object $x \in \mathcal{X}$ that has been subjected to a geometrical transformation $\mathcal{T}_\alpha$ from a continuous set $\alpha \in \mathcal{A}$, i.e.:

$$z \approx \mathcal{T}_\alpha x. \tag{1}$$

2.1. Transformations and Exemplars

The partition of the underlying image space into the transformation set $\mathcal{A}$ and class $\mathcal{X}$ of normalized images could take a variety of forms. For example, in analysis of face images, $\mathcal{A}$ could be a shape space, modeling geometrical distortions, and $\mathcal{X}$ could be a space of textures, in the manner of Cootes et al. (1998) and Vetter and Poggio (1996). Alternatively $\mathcal{A}$ could be a space of planar similarity transformations, leaving $\mathcal{X}$ to absorb both distortions and texture/shading distributions. In any case, $\mathcal{A}$ is to be defined analytically in advance, leaving $\mathcal{X}$ to be inferred from the training sequence $\mathcal{Z}^*$. A feature of this work is that the class $\mathcal{X}$ of normalized images is not assumed to be amenable to straightforward analytical description; instead $\mathcal{X}$ is defined in terms of a set $\{\tilde{x}_k, k = 1, \ldots, K\}$ of exemplars, together with a distance function ρ, in the spirit of Gavrila (Gavrila and Philomin, 1999). For example, the face of a particular individual, might be represented by a set of exemplars $\tilde{x}_k$ consisting of normalized (registered), frontal views of that face, wearing a variety of expressions, and in a variety of poses and lighting conditions. Crucially, exemplars will be interpreted probabilistically, so that the uncertainty inherent in the approximation (1) is accounted for explicitly. The interpretation of an image z is then as a state vector $X = (\alpha, k)$.

2.2. Learning

Aspects of the probabilistic model that must be learned from $\mathcal{Z}^*$ include:

1. The set of exemplars $\{\tilde{x}_k, k = 1, \ldots, K\}$.
2. Component distributions, centered on each of the $T_\alpha \tilde{x}_k$, for some α for observations given state $X = (\alpha, k)$. The details of this density, and the algorithm for learning it, constitute a new approach to the vexed question of how to model image observations probabilistically without tripping over the issue of statistical independence.
3. A predictor in the form of a conditional density $p(X_t \mid X_{t-1})$ to represent the (typically strong) prior dependency between states at successive time steps.

These elements (together with a prior $p(X_1)$) form a structured prior distribution for a randomly sampled image sequence $z_1, \ldots, z_T$, which can be tested for plausibility by random simulation (see Fig. 3, for example).

The prior model then forms a basis for interpretation of image sequences via the posterior

$$p(X_1, X_2, \ldots \mid z_1, z_2, \ldots; \Lambda)$$

where Λ is the set of learned parameters of the probabilistic model, including the exemplar set, the noise parameters, and the dynamic model.

3. Probabilistic Modeling of Images and Observations

The probabilistic dependency structure for the M^2 model is depicted in Fig. 2 and is similar to Frey and Jojic (2000). However, the similarity of dependency structure belies crucial innovations in representation and probability distributions which are explained below.

3.1. Objects

An object in the class $\mathcal{X}$ is taken to be an image that has been preprocessed to enhance certain features, resulting in a preprocessed image x. The M^2 approach is general enough to apply to a variety of such images—we will consider two: unprocessed raw images, and sparse binary images with true-valued pixels marking a set of feature curves.

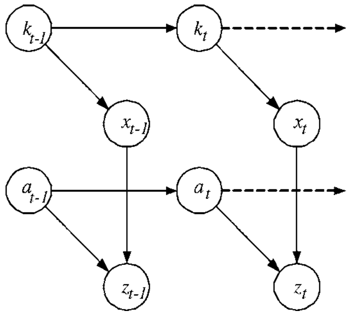

Figure 2. Probabilistic graphical structure for the M^2 model: The observation z_t at time t is an image drawn from a mixture with centers $\{\mathcal{T}_\alpha \tilde{x}_k, k = 1, \ldots, K\}$, where $\{\tilde{x}_k, k = 1, \ldots, K\}$ are exemplars; $\mathcal{T}_\alpha$ is a geometrical transformation, indexed by the (real-valued) parameter α.

3.1.1. Patches. In the case of real-valued output from preprocessing, z is an image subregion, or *patch*, visible as an intensity function $I_z(\mathbf{r})$. As mentioned earlier, it is undesirable to have to assume a known parameterization of the intensity function on that patch. For now, we make the conservative assumption that some linear parameterization, with parameters $\mathbf{y} \in \mathcal{R}^d$, of a priori unknown form and dimension d, exists, so that:

$$I_z(\mathbf{r}) = \sum_{i=1}^{d} I_i(\mathbf{r}) y_i \tag{2}$$

where $I_1(\mathbf{r}), \ldots, I_d(\mathbf{r})$ are independent image basis functions and $\mathbf{y} = (y_1, \ldots, y_d)$. Given the linearity assumption, all that will need be inferred about the nature of the patch basis is its dimensionality d. There is no requirement to know or infer the form of the I_i. A suitable distance function ρ is needed for patches. For robustness we will use a "shuffle distance" (Kutulakos, 2000), which is an L_2 norm applied after first associating each pixel in one image with the most similar pixel in a neighborhood around the corresponding pixel in the other image. (We show later why we chose this distance over others.)

3.1.2. Curves. The situation for binary images is similar to that for patches, except that a different distance function is needed, and the interpretation of the linear parameterization is a little different, too. Now z is visible as a curve $\mathbf{r}_z(s)$, with curve-parameter s, and linearly dependent on $\mathbf{y} \in \mathcal{R}^d$:

$$\mathbf{r}_z(s) = \sum_{i=1}^{d} \mathbf{r}_i(s) y_i, \tag{3}$$

where $\mathbf{r}_1(s), \ldots, \mathbf{r}_d(s)$ are now independent curve basis functions such as parametric B-splines (Bartels et al., 1987). In this case, the distance measure ρ used is a (non-symmetric) *chamfer* distance (Gavrila and Philomin, 1999). The chamfer distance is defined to be

$$\rho(\tilde{z}, z) = \min_{s'(s)} \int ds\, g(|\mathbf{r}_z(s') - \mathbf{r}_{\tilde{z}}(s)|), \tag{4}$$

where $g(\cdot)$ is the profile of the chamfer. A particularly interesting case is the quadratic chamfer, in which $g(u) = u^2$, or a truncated form $g(u) = \min(u^2, g_0)$. In that case, chamfer distance (4) is known to approximate a curve-normal-weighted L2 distance between the two curves, in the limit that they are similar. (Note that chamfer distance is related to Hausdorff distance which has been used successfully in tracking (Huttenlocher et al., 1993); the difference is that the integral in (4) becomes a max operator in the Hausdorff distance.)

A great attraction of the chamfer distance is that it can be computed directly from the (binary) images z and $\tilde{z}$, as

$$\rho(\tilde{z}, z) = \int ds\, \gamma(z, \mathbf{r}_{\tilde{z}}(s)) \tag{5}$$

using a chamfer image

$$\gamma(z, \mathbf{r}) = \min_{s'} g(|\mathbf{r}_z(s') - \mathbf{r}|)$$

constructed directly from z. This allows $\rho(\tilde{z}, z)$ to be evaluated repeatedly for a given z and various $\tilde{z}$ directly from (5) which, being simply a curve integral (approximated), is numerically very efficient.

3.2. Geometric Transformations

Geometric transformations $\alpha \in \mathcal{A}$ are applied to exemplars to give transformed mixture centers:

$$\tilde{z} = T_\alpha \tilde{x}.$$

For example, in the case of Euclidean similarity, $\alpha = (\mathbf{u}, \theta, s)$ and vectors transform as

$$T_\alpha \mathbf{r} = \mathbf{u} + R(\theta)\, s\, \mathbf{r},$$

in which $(\mathbf{u}, \theta, s)$ are offset, rotation angle and scaling factor respectively. Where the observations are curves, this induces a transformation

$$\mathbf{r}_z(s) = T_\alpha \mathbf{r}_x(s)$$

and in the case of patches, the induced transformation is

$$I_z(T_\alpha \mathbf{r}) = I_x(\mathbf{r}).$$

3.3. The Metric Mixture (M^2) Model

The observation likelihood function, at the heart of the M^2 approach, can now be specified. Note that the observation is deemed to be the finite dimensional vector y, rather than the infinite dimensional image or curve z.

The full image/curve is accessed only as a "machine" for computing an observation density. Note also, We exploit the fact that we only need to know enough about $p(y \mid X)$ to *evaluate* it. There is no call to *sample* from it. Hence no constructive form for the observer need be given, and we can avoid controversies about pixelwise independence.

3.3.1. Exemplars as Mixture Centers. The object class is defined in terms of a set $\mathcal{X} = \{\tilde{x}_k, k = 1, \dots, K\}$ of untransformed exemplars, to be inferred from the training set $\mathcal{Z}$. A transformed exemplar $\tilde{z}$ serves as a center in a mixture component:

$$p(y \mid \tilde{z}) \propto \frac{1}{Z} \exp - \lambda \rho(\tilde{z}, z) \tag{6}$$

—a "metric exponential" distribution—whose normalization constant or "partition function" is Z.

3.3.2. Metric-Based Mixture Kernels. For tracking of the full state, both motion and shape, the hypothesis is $X = (\alpha, k)$. The mixture model above leads to an observation likelihood

$$p(y \mid X) \equiv p(y \mid \alpha, k) \propto \frac{1}{Z} \exp - \lambda \rho(T_\alpha \tilde{x}_k, z). \tag{7}$$

If only motion is to be tracked, the hypothesis is simply α so the observation likelihood becomes

$$p(y \mid \alpha) \propto \sum_k \pi_k \frac{1}{Z} \exp - \lambda \rho(T_\alpha \tilde{x}_k, z),$$

a mixture with component priors π_k. For this interpretation to make sense, it is necessary to "tie" the dimension d_k of the y-space associated with each component to be a constant $d_k = d$. Henceforth, we deal with the joint α, k space as in (7) so that tying the d_k will not be necessary.

3.3.3. Partition Function. In order to learn the value of an exponential parameter λ from training data, it is necessary to know something about the partition function Z. This is difficult in general, but straightforward in the case that ρ is a (truncated) quadratic chamfer function because that gives an approximately Gaussian distribution. Similarly, an L_2 norm on patches leads to a Gaussian mixture distribution, as does the shuffle-metric used in experiments reported here.[1] In that case, the exponential constant λ in the observation likelihood is interpreted as $\lambda = \frac{1}{2\sigma^2}$, where σ is an distance constant, and the partition function is $Z \propto \sigma^d$. From this, it can be shown (see appendix) that the chamfer distance $\rho \mid \tilde{z} \equiv \rho(\tilde{z}, z)$ is a $\sigma^2 \chi_d^2$ random variable (i.e., ρ/σ^2 has a χ_d^2 distribution) which is in fact also a Γ distribution. This allows the parameters σ, d of the observation likelihood (7) to be learned from training data, as set out below.

4. Learning Algorithms

4.1. *Learning Mixture Kernel Centers*

Following the probabilistic interpretation of exemplars as kernel centers $\tilde{x}_k$ in (6), we exploit the temporal continuity of the training sequence $\mathcal{Z}^*$ to choose initial mixture centers, and proceed to cluster iteratively.

1. The training set is assumed to be approximately aligned from the outset (this is easily achieved provided the foreground in the training sequence is reasonably easy to separate, for example by background subtraction/filtering). To improve the initial alignment, first a datum, z_0^*, is chosen from the entire training sequence $\mathcal{Z}^*$ according to a minimax rule:

$$z_0^* \leftarrow \arg \min_{z \in \mathcal{Z}^*} \max_{z' \in \mathcal{Z}^* - \{z\}} \rho(z, z').$$

Then,

$$\alpha_t^* = \arg \min_\alpha \rho\left(T_\alpha^{-1} z_t^*, z_0^*\right) \quad \text{and} \quad x_t^* = T_{\alpha_t^*}^{-1} z_t^*,$$

minimizing by direct descent.

2. To initialize centers, a subsequence of the x_t^* is chosen to form the initial $\tilde{x}_k$, selected in such a way as to be evenly spaced in chamfer distance. Thus the $\tilde{x}_k$ are chosen so that $\rho(\tilde{x}_{k+1}, \tilde{x}_k) \approx \rho_c$, for some appropriate choice of ρ_c that gives approximately the required number K of exemplars.
3. For the remainder of the aligned training data x_t^*, $t = 1 \dots T^*$, find the cluster that minimizes the distance from x_t^* to the cluster center:

$$k_t(x_t^*) = \arg \min_k \rho(\tilde{x}_k, x_t^*). \tag{8}$$

Label the set of all elements in cluster k as $\mathcal{C}_k = \{x_t^* : k_t(x_t^*) = k\}$ and let $N_k = |\mathcal{C}_k|$.

4. For each cluster k, find the new representative, which is the element in that cluster that minimizes the maximum distance to all other elements in that cluster:

$$\tilde{x}_k \leftarrow \arg\min_{x \in \mathcal{C}_k} \max_{x' \in \mathcal{C}_k - \{x\}} \rho(x, x'). \qquad (9)$$

5. Repeat Steps 3 and 4 for a fixed number of iterations or until convergence and save the final exemplars $\tilde{x}_k$.
6. Set mixture weights: $\pi_k \propto N_k$.

Steps 3 and 4 implement a k-medoids algorithm (Ripley, 1996). This is analogous to the iterative computation of cluster centers in the k-means algorithm, but is applicable in a non-metric space where it is impossible to compute cluster means. In place of a mean, an existing member of the training set is chosen by a minimax rule, since that is equivalent to the mean in the limit that the training set is dense and is defined over a vector space with a Euclidean distance.

4.2. Learning the M^2 Kernel Parameters

To learn observation likelihood parameters σ, d, we obtain a validation set $\mathcal{Z}_v$. (This could simply be the training set $\mathcal{Z}$ less the (unaligned) exemplars $\{\tilde{z}_k\}$.) For each z_v from $\mathcal{Z}_v$, the corresponding aligning transformation α_v and mixture center $\tilde{x}_v$ is estimated by minimizing, by direct descent, the distance:

$$\min_{\alpha \in \mathcal{A}, \tilde{x} \in \mathcal{X}} \rho(T_\alpha \tilde{x}, z_v).$$

Now, following Section 3.3, we treat the distances

$$\rho_v(z_v) = \rho\big(T_{\alpha_v} \tilde{x}_v, z_v\big), \quad z_v \in \mathcal{Z}_v$$

as $\sigma^2 \chi^2_d$ distributed. An approximate but simple approach to parameter estimation is via the sample moments

$$\bar{\rho}_k = \frac{1}{N_k} \sum_{z_v \in \mathcal{C}_k} \rho_v(z_v) \quad \text{and} \quad \bar{\rho}^2_k = \frac{1}{N_k} \sum_{z_v \in \mathcal{C}_k} \rho^2_v(z_v),$$

which, from the form of the mean and variance of the χ^2 statistic, in terms of ρ, σ, gives the following estimates for d_k and σ_k:

$$d_k = 2 \frac{\bar{\rho}^2_k}{\bar{\rho}^2_k - \bar{\rho}^2_k} \quad \text{and} \quad \sigma_k = \sqrt{\bar{\rho}_k / d}. \qquad (10)$$

Intuitively, d_k is estimated here in terms of the histogram of ρ-values. A histogram whose mass is concentrated at low ρ-values gives a lower d estimate.

Alternatively, the full maximum likelihood (MLE) solution, complete with integer constraint on d, yields σ values in terms of d, exactly as above, and integer $d \geq 1$ as the value maximizing the likelihood

$$L(d) = -\log \Gamma(d/2) + (d/2)(\log(d/2) \\ -1 - \log(\bar{\rho}_a / \bar{\rho}_g)) \qquad (11)$$

(dropping the k-subscripts for simplicity), where $\bar{\rho}_a$, $\bar{\rho}_g$ are respectively the arithmetic and geometric means of the ρ-samples, and $\Gamma(\cdot)$ is the well-known, transcendental Γ-function. Such a d can always be found since $L(d)$ is asymptotically a decreasing function of d.

4.2.1. Notes

1. If $\bar{\rho}_a / \bar{\rho}_g > 4/e$ the solution for d is the trivial $d = 1$.
2. The estimation procedures are equivalent to fitting a Γ-distribution to the ρ-values to determine parameters d_k. The moments estimator fits an unconstrained Γ-distribution, so the integer constraint on d is not applied.
3. The MLE applies the integer constraint to d. However, in practice the MLE turns out to be less robust than a moments estimator, in cases when the observed ρ statistic does not follow the assumed distribution closely.

4.3. Learning Dynamics

In line with recent developments in probabilistic tracking (Blake and Isard, 1998), sequences of estimated X_t from a training set are treated as if they were fixed time-series data, and used to learn two components (assumed independent) of $p(X_t \mid X_{t-1})$:

1. a Markov matrix M for $p(k_t \mid k_{t-1})$, learned by histogramming transitions;
2. a first order auto-regressive process (ARP) for $p(\alpha_t \mid \alpha_{t-1})$, with coefficients calculated using the Yule-Walker algorithm (Gelb, 1974).

5. Results

In order to demonstrate the necessity for, and applicability of, the M^2 model, we performed tracking experiments in two separate domains. In the first, we tracked walking people using contour edges. Here, background clutter and simulated occlusion threaten to distract tracking without a reasonable dynamic model and a good likelihood function.

In the second, we track a person's mouth based on raw pixel values. Unlike the pedestrian-tracking domain, images are cropped such that only the mouth, and no background, is visible. While distraction is not a problem, the complex articulations of the mouth make tracking difficult (even state-of-the-art face-tracking algorithms (Cootes et al., 1998; Neven, 2000) have difficulty tracking lip and tongue articulation).

5.1. Tracking Human Motion

For the person tracking experiments, training and test sequences show various people walking from right to left in front of a stationary camera. The background in all of the training sequences is fixed, allowing us to use simple background subtraction and edge-detection routines to automatically generate the exemplars (naturally, we took advantage of the fixed background only for the purposes of generating exemplars—not for tracking). Examples of a few exemplars are shown in Fig. 3.

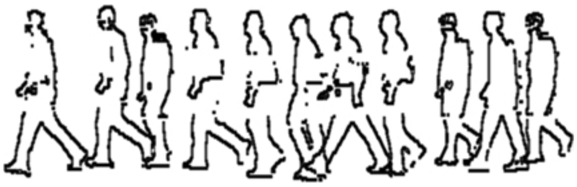

Figure 3. A sequence generated at random from a model based on learned dynamics and exemplars. Edges shown represent the contours of successive model exemplars.

Dynamics were learned as described in Section 4.3 on 5 sequences of the same walking person, each about 100 frames long. Figure 3 overlays several frames from a sequence generated at random from the learned model. The full sequence is available as `generatd.mpg`.[2]

5.1.1. *Validity of the M^2 Model.* A practical test of the M^2 methodology is whether consistent d values can be estimated from Eq. (10). We tested this for chamfer distance by conducting experiments on synthetically generated polygons with d vertices, with the results shown in Fig. 4.

Figure 5 shows values of dimension d for the pedestrian contour exemplars. Note that dimensionality increases with cluster size up to a point, but it eventually converges to $d \approx 10$. We read the fact that d does not simply increase unboundedly with cluster size, as an indication that d reflects an intrinsic local dimensionality.

5.1.2. *Practical Tracking.* We can now compute observation likelihoods as in Eq. (7) and track using the following Bayesian framework. A classical forward algorithm (Rabiner, 1989) would give $p_t(X_t) \equiv p(X_t \mid z_1, \ldots, z_t)$ as:

$$p_t(X_t) = \sum_{k_{t-1}} \int_{\alpha_{t-1}} p(y_t \mid X_t) p(X_t \mid X_{t-1}) p_{t-1}(X_{t-1}),$$

where $p(y_t \mid X_t)$ is computed according to Eq. (7). Exact inference is infeasible given that α is real-valued,

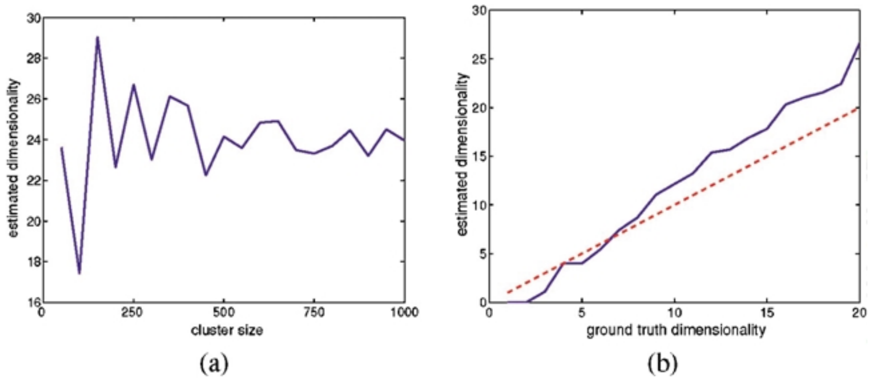

Figure 4. Convergence of d-estimate with cluster size, for synthesized polygons. (a) Estimated dimension converges to around $d = 24$ as cluster size increases to 1000 (true dimension $d = 22$). (b) Estimated dimensionality (solid) closely follows ground truth dimensionality (dashed). Dimension appears to be consistently slightly overestimated (cluster size $N = 1000$). This may be due to the approximation inherent in using the chamfer distance.

Object	Average cluster size	d	σ
Person contour	5	2.8	21.6
	10	4.1	14.4
	20	5.1	18.3
	40	5.0	17.9

Figure 5. Estimated dimension d for image contour exemplars, using quadratic chamfer distance, appears to behave consistently: it converges to $d = 10$ as data set size increases.

so the integral is performed using a form of particle filter (Gordon et al., 1993; Isard and Blake, 1996). To display results, we calculate $\hat{X}_t = \arg\max p_t(X_t)$.

Figure 1 shows cropped, sample images of tracking on a sequence that was not in the training sequence. Tracking in this case is straightforward and accurate. Figure 6 shows the same exemplar set (trained on one person) used to track a different person entirely. Although the swing of this subject's arms is not captured by the existing exemplars, the gait is nevertheless accurately tracked. Finally, we ran an experiment to verify tracking robustness against occlusion and other visual disturbances. In Fig. 7, we simulated occlusions by rendering black two adjacent frames out of every ten frames in the test sequence, and so tracking was forced to rely on the prior in these frames.

The sequence was accurately tracked in the non-occluded frames, bridged by reasonable state estimates in the black frames—something that would be impossible without incorporation of a dynamic model.

Experiments with a more complex and agile set of movements are shown in Fig. 8. In this case it is necessary to use a greater number of exemplars ($K = 300$). Note that the experiments here show unsupervised learning—parameter estimation and tracking—on a single sequence.

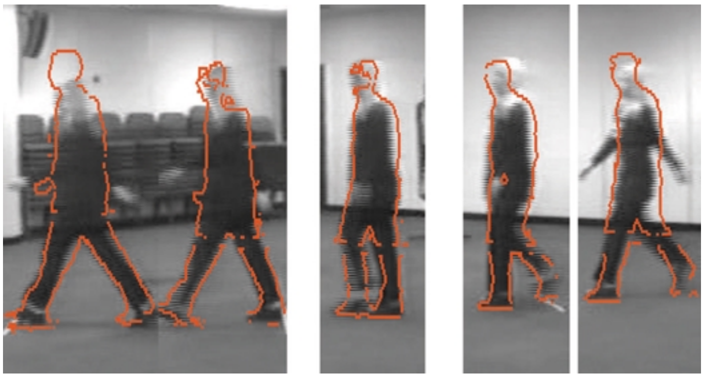

Figure 6. Cropped, sample frames from a tracked test sequence. The same learned model is used as Fig. 1, but now the test sequence contains a new individual walking. The motion is captured nonetheless, though the match is not quite so close—note the arms in the final frame (see also video `walk3.mpg`).

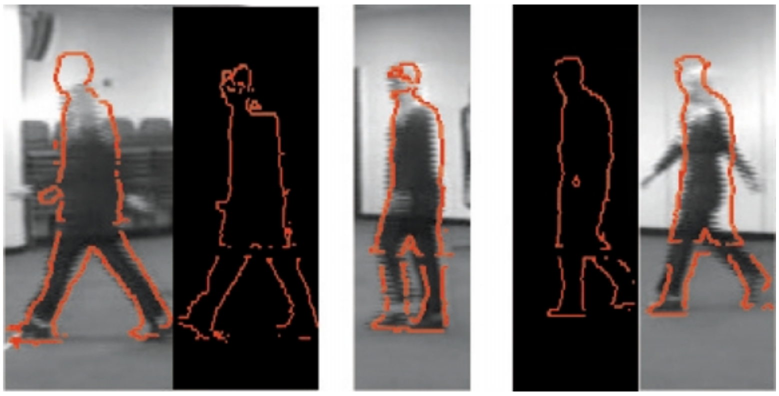

Figure 7. Cropped, sample frames from a tracked test sequence. The same learned model and test sequence is used as in Fig. 6, but now the test sequence is periodically blanked out, to test robustness to occlusion (see also video `walk3occ.mpg`).

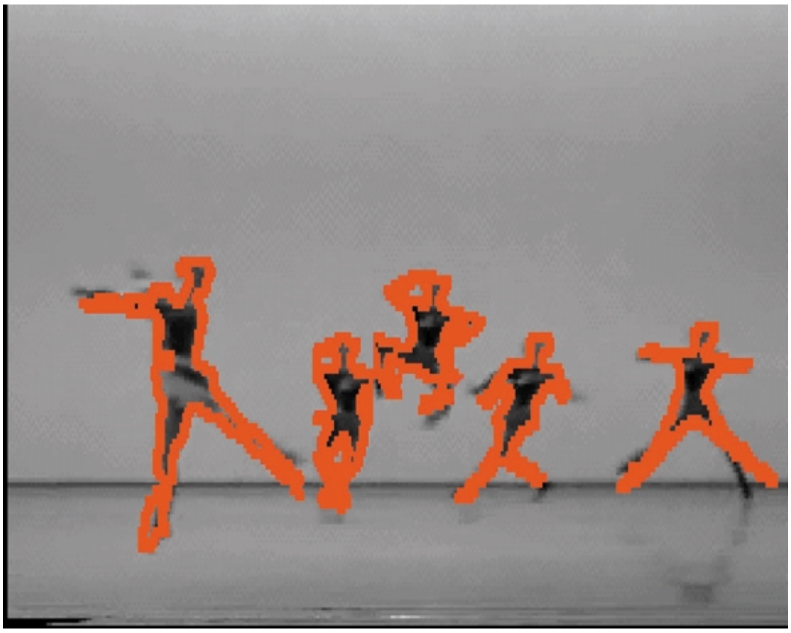

Figure 8. Results on learning only, with a more varied set of motions, requiring a larger number of exemplars. The sequence is taken from the movie, *Center Stage* (see also video `ballet0.mpg`).

5.2. Mouth Tracking

The mouth tracking sequences consisted of closely cropped images of a single subject's mouth while the person was speaking and making faces. The training sequence consisted of 210 frames captured at 30 Hz. We tested on a longer test sequence of 570 frames (of which 270 are shown in the video files described below). Dynamics were learned as in Section 4.3, with $K = 30$ exemplar clusters. Tracking was performed as in Section 5.1, but with no α transformations, since the images were largely registered. On this training set, the shuffle distance d values exhibited greater variance (the extremes running from 1.2 to 13.8), but the majority of clusters showed a dimensionality of $d = 4 \pm 1$, indicating again that the dimension constant d in the M^2 model is learned consistently (see Fig. 9 for a histogram showing the distribution of estimated dimensions).

The results for this experiment can be seen in video format (see also, Fig. 10): `ml2.mpg` shows the result of tracking based on the L_2 distance (Euclidean distance between vectors formed by concatenating the raw pixel values of an image), and `mshuffle.mpg` shows tracking using the shuffle distance.

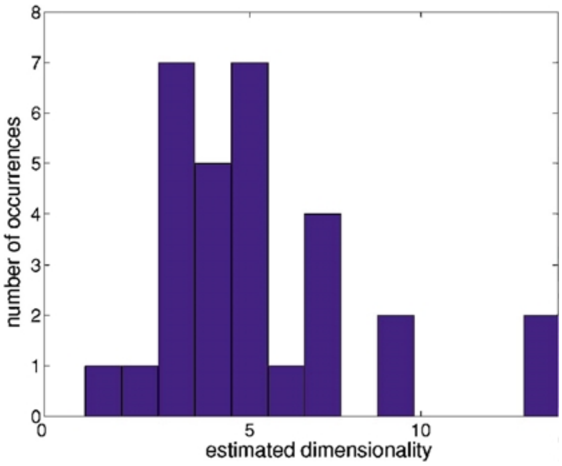

Figure 9. A histogram of estimated dimensionality for clusters learned for the mouth-tracking sequence.

In the videos, the left-hand image shows the test image, and the right-hand image shows the *a posteriori* best-match exemplar from the training sequence. Both functions do well with the initial two-thirds of the test sequence, while the subject is speaking. As soon as the subject begins to make faces and stick out his tongue, the L_2-based likelihood fails, whereas tracking based on the shuffle distance continues largely successfully.

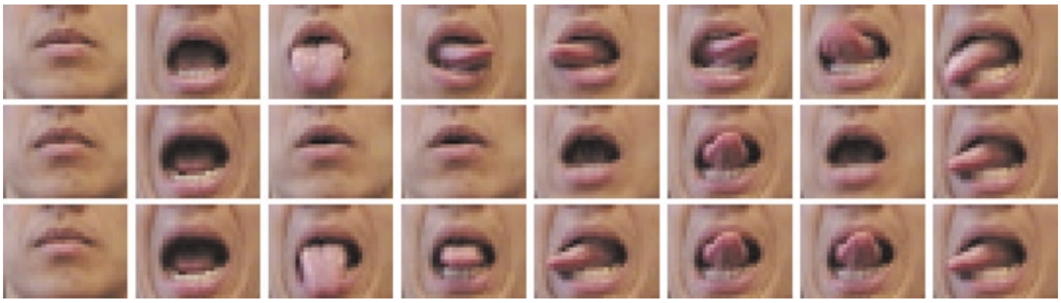

Figure 10. A sampling of frames from the mouth sequence. Top row, the test sequence; middle row, tracking using the L_2 distance; bottom row, tracking using the shuffle distance. The shuffle distance produces matches that are perceptually more similar to the test sequence.

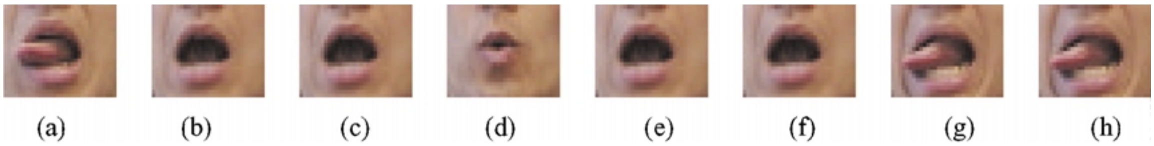

Figure 11. Best match, based on various distance functions: (a) test image, (b) L_2 distance, (c) L_2 after blurring, (d) histogram matching, (e) L_2 distance after projecting to PCA subspace with 20 bases, (f) L_2 after projection to PCA subspace with 80 bases, (g) L_2 after image warp based on optic flow, (h) shuffle distance as described in text.

Figure 11 shows a comparison of maximum-likelihood matches, on one of the difficult test images—a tongue sticking out to the left—for a variety of distance functions. Most of the functions prefer an exemplar without the tongue. This may be because of the high contrast between pixels projected dimly by the inside of the mouth and those projected brightly by lip and tongue; even a small difference in tongue configuration can result in a large difference in L_2, and other, distances. On the other hand, the flow-based distance and the shuffle distance (really an inexpensive version of the flow-based distance) return exemplars that are perceptually similar. These functions come closer to approximating perceptual distances by their relative invariance to local warping of images. These observations were what originally led to our experiments with different distance functions, and they justify the need for the ability to handle metrics that are not embedded in a vector space.

6. Conclusion

The Metric Mixture approach combines the advantages of exemplar-based models (Gavrila and Philomin, 1999) with a probabilistic framework (Frey and Jojic, 2000) into a single probabilistic exemplar-based paradigm. The power of the M^2 technique comes from its generality: both object models and noise models can be learned automatically, and metrics can be chosen without significant restrictions on the structure of the metric space (a drawback of Markov random field models of image-pixel dependencies, for example).

We intend to explore several avenues in future work:

- One problem with exemplar sets is that they can grow exponentially with object complexity. Tree structures appear to be an effective way to deal with this problem (Gavrila and Philomin, 1999; Wei and Levoy, 2000), and we would like to find effective ways of using them in a probabilistic setting. Note however, that the use of a dynamical model for prediction greatly reduces the effective size (perplexity) of the exemplar set, so the lack of tree structure has not been a serious limiting factor yet.
- We propose to continue testing on sequences with more intense background clutter, and with more varied transformations α, to explore the limits of the exemplar approach.

Appendix: Quadratic Chamfer Distance has a Scaled χ^2 Distribution

We have, from (4) with quadratic $g(u) = u^2$,

$$\rho \mid \tilde{z} \equiv \rho(\tilde{z}, z) = \|\mathbf{r}_z(s) - \mathbf{r}_{\tilde{z}}(s)\|^2.$$

From (3),

$$\rho \mid \tilde{z} = \mathbf{y}^\top \mathcal{H}^{-1}\mathbf{y} + \mathcal{O}(\mathbf{y})$$

where $\mathcal{O}(\mathbf{y})$ is a linear term in the parameter vector $\mathbf{y}$. Matrix $\mathcal{H}_{i,j}$ is a nonsingular, symmetric, metric matrix (Blake and Isard, 1998) which can be diagonalized as $\mathcal{H} = UDU^\top$, in which U is orthogonal and D is diagonal. Now, from (6), and using the normalization properties of Gaussians,

$$p(z \mid \tilde{z}) = (\sqrt{2\pi}\sigma)^{-d}|\mathcal{H}|^{-1/2} \exp -\frac{1}{2\sigma^2}(\rho \mid \tilde{z}),$$

where $1/(2\sigma^2) = \lambda$ as before. Therefore $\mathbf{y}$ is a normal random variable:

$$\mathbf{y} = B\mathbf{w} \quad \text{where } \mathbf{w} \sim \mathcal{N}(0, I_d) \quad \text{and}$$
$$B = \sigma\mathcal{H}^{-1/2} = \sigma U D^{-1/2} U^\top.$$

Finally,

$$\rho \mid \tilde{z} = \mathbf{w}^\top B^\top \mathcal{H}^{-1} B\mathbf{w} = \sigma^2 \mathbf{w}^\top \mathbf{w}$$

so $(\rho \mid \tilde{z})$ is a $\sigma^2\chi_d^2$ random variable, as claimed.

Acknowledgments

We thank P. Anandan, Neil Lawrence, and Chris Williams for stimulating discussions. John MacCormick kindly provided video data; Rick Szeliski, code for performing flow-based matching.

Notes

1. The shuffle-metric can be thought of as using an image-sized array s of hidden variables augmenting the state vector X, before applying a classical L_2 norm.
2. All movie files mentioned in this paper are available at `http://research.microsoft.com/vision/papers`.

References

Amini, A., Tehrani, S., and Weymouth, T. 1988. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Proc. 2nd Int. Conf. on Computer Vision*, pp. 95–99.

Bartels, R., Beatty, J., and Barsky, B. 1987. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann: San Mateo, CA.

Bascle, B. and Deriche, R. 1995. Region tracking through image sequences. In *Proc. 5th Int. Conf. on Computer Vision*, Boston, June 1995, pp. 302–307.

Black, M. and Jepson, A. 1996. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. 4th European Conf. Computer Vision*, pp. 329–342.

Blake, A. and Isard, M. 1998. *Active Contours*. Springer: Berlin.

Brand, M. 1999. Shadow puppetry. In *Proc. Int. Conf. on Computer Vision*, pp. 1237–1244.

Cootes, T., Edwards, G., and Taylor, C. 1998. Active appearance models. In *Proc. European Conf. on Computer Vision*, pp. 484–498.

Efros, A. and Leung, T. 1999. Texture synthesis by non-parametric sampling. In *Proc. Int. Conf. on Computer Vision*, pp. 1033–1038.

Field, D. 1987. Relations between the statistics of natural images and the response properties of cortical cells. *J. Optical Soc. of America A.*, 4:2379–2394.

Freeman, W. and Pasztor, E. 1999. Learning to estimate scenes from images. In *Advances in Neural Information Processing Systems*, Vol. 11. MIT Press: Cambridge, MA.

Frey, B. and Jojic, N. 2000. Learning graphical models of images, videos and their spatial transformations. In *Proc. Conf. Uncertainty in Artificial Intelligence*.

Gavrila, D. and Philomin, V. 1999. Real-time object detection for smart vehicles. In *Proc. Int. Conf. on Computer Vision*, pp. 87–93.

Gelb, A. (Ed.). 1974. *Applied Optimal Estimation*. MIT Press: Cambridge, MA.

Gordon, N., Salmond, D., and Smith, A. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc.* F, 140(2):107–113.

Hager, G. and Toyama, K. 1996. XVision: Combining image warping and geometric constraints for fast tracking. In *Proc. 4th European Conf. Computer Vision*, pp. 507–517.

Huttenlocher, D., Noh, J., and Rucklidge, W. 1993. Tracking nonrigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, pp. 93–101.

Isard, M. and Blake, A. 1996. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. Computer Vision*, Cambridge, UK, April 1996, pp. 343–356.

Kass, M., Witkin, A., and Terzopoulos, D. 1987. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pp. 259–268.

Kutulakos, K. 2000. Approximate N-view stereo. In *Proc. European Conf. Computer Vision*, Vol. 1, pp. 67–83.

Mumford, D. 1996. Pattern theory: A unifying perspective. In *Perception as Bayesian Inference*, D. Knill and W. Richard (Eds.), Cambridge University Press: Cambridge, UK, pp. 25–62.

Neven, H. 2000. Eyematic interfaces. In *Siggraph Demo Session*. Los Angeles.

Rabiner, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2): 257–285.

Ripley, B. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press: Cambridge.

Storvik, G. 1994. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. Patt. Anal. Mach. Intel.*, 16(10):976–986.

Terzopoulos, D. and Szeliski, R. 1992. Tracking with Kalman snakes. In *Active Vision*, A. Blake and A. Yuille (Eds.), MIT: Cambridge, MA, pp. 3–20.

Vetter, T. and Poggio, T. 1996. Image synthesis from a single example image. In *Proc. 4th European Conf. Computer Vision*, Cambridge, UK, April 1996, pp. 652–659.

Wei, L.-Y. and Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proc. ACM Siggraph*, ACM: New York.