Probabilistic Tracking in a Metric Space

Kentaro Toyama
Microsoft Research
Redmond, WA, U.S.A.
kentoy@microsoft.com

Andrew Blake
Microsoft Research Ltd.
Cambridge, United Kingdom
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. The M^2 model has several valuable properties. 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.

1 Introduction

There is, of course, a substantial literature on tracking, driven either by image features [1, 18] or by raw intensity [3, 4, 15], or both [7]. Tracking can be formulated in a probabilistic framework in both the feature-driven [24] and intensity-driven [23] 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* offers an alternative that can tackle this problem [6, 8, 10, 11, 12]. 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 [12], 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 [11] have demonstrated elegantly how exemplars can be embedded in learned probabilistic models by treating them as centers in probabilistic mixtures. 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, their 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 [12];
- finally, image noise is treated as white despite known, strong statistical correlations between pixels [9].

The problem, therefore, is to combine exemplars in a metric space [12] with a probabilistic treatment [11], retaining the best features of each approach. Unfortunately, this combination is not trivial – the very techniques which make probabilistic treatment possible (*i.e.*, modelling with Gaussians, PCA, K-means, EM, etc.), are not applicable without a vector-space structure for exemplars, which the former lacks. 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.

0-7695-1143-0/01 $10.00 (C) 2001 IEEE

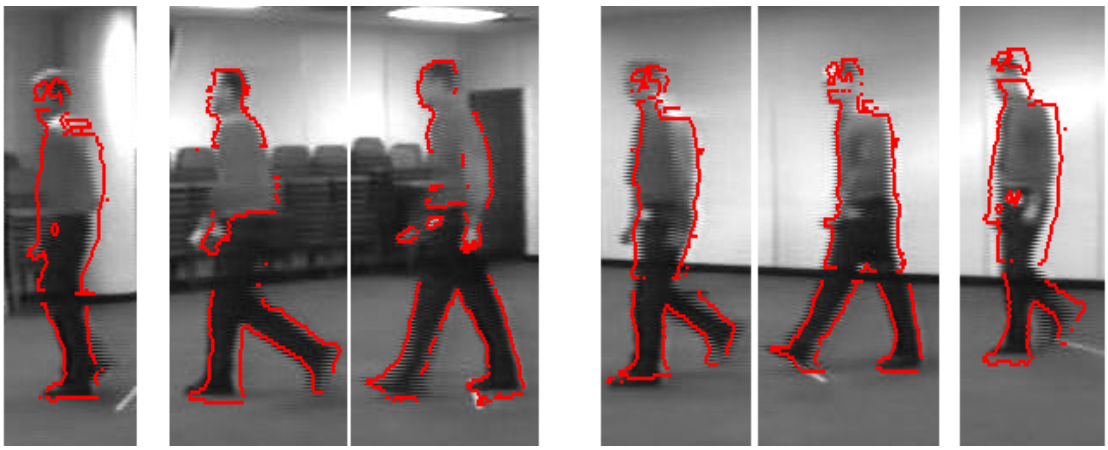


Figure 1. Cropped, sample frames from a tracked sequence. The overlays represent the maximum *a posteriori* exemplars. The person tracked appears in the training sequences.

One note on our terminology: The theory and algorithms presented were developed 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, applies also 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 analysed 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 edge pixels marked. A given image z is to be approximated, in the familiar pattern theoretic manner [20], 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, modelling geometrical distortions, and $\mathcal{X}$ could be a space of textures, in the manner of [7, 25]. 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 [12]. 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 z; *i.e.*, each component is a distribution $p(z|X)$, where $X = (\alpha, k)$ 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 $p(X_t|X_{t-1})$ to represent the (typically strong) prior dependency between states at successive timesteps.

These elements (together with a prior $p(X_1)$) form a structured prior distribution for a randomly sampled image se-

quence $z_1, \ldots, z_T$, which can be tested for plausibility by random simulation (see Figure 3, for example).

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

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

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 Modelling of Images and Observations

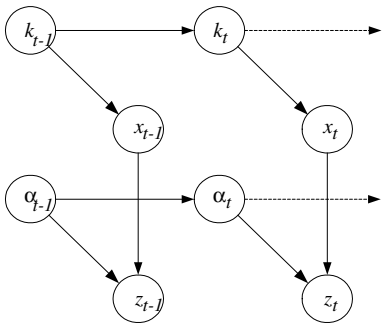


Figure 2. Probabilistic graphical structure for the M^2 model: The observation z_t at time t is an image drawn from a mixture with centers $\{\mathcal{T}_\alpha \tilde{x}_k,\ k = 1, \ldots, K\}$, where $\{\tilde{x}_k, k = 1, \ldots, K\}$ are exemplars; $\mathcal{T}_\alpha$ is a geometrical transformation,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

The probabilistic dependency structure for the M^2 model is depicted in Figure 2 and is similar to [11]. 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.

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 need be known about the nature of the patch basis is its dimensionality d. There is no requirement to know the form of the I_i. A suitable distance function ρ is needed for patches. For robustness we will use a "shuffle distance" [19], in which each pixel in one image is first associated with the most similar pixel in a neighbourhood around the corresponding pixel in the other image. (Figure 6 shows why we chose this function over others.)

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 [2]). In this case, the measure $\rho(x, \tilde{x})$ used is a (non-symmetric) "chamfer" distance [12]. The chamfer distance can be computed directly from the (binary) images x and $\tilde{x}$, using a chamfer image constructed from $\tilde{x}$, and without recourse to any parametric representation of the underlying curves.

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 functions, at the heart of the M^2 approach, can now be specified. We exploit the fact that we only need to know enough about $p(z|X)$ 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.

Exemplars as Mixture Centers

The object class is defined in terms of a set $\mathcal{X} = \{\tilde{x}_k,\ k = 1, \ldots, K\}$ 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:

$$p(z|\tilde{z}) \propto \frac{1}{Z} \exp -\lambda\rho(z, \tilde{z}) \tag{4}$$

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

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

$$p(z|X) \equiv p(z|\alpha, k) \propto \frac{1}{Z} \exp -\lambda\rho(z, T_\alpha \tilde{x}_k). \tag{5}$$

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

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

a mixture with component priors π_k.

Partition Function

In order to learn the value of an exponential parameter λ from training data, we need to know something about the partition function Z. This is difficult in general, but straightforward in the case that ρ is a quadratic chamfer function:

$$\rho(z, \tilde{z}) = \min_{s'(s)} \|\mathbf{r}_z(s) - \mathbf{r}_{\tilde{z}}(s')\|^2, \tag{6}$$

is then the squared-Hausdorff distance [16], which is approximately quadratic [5, Section 6.2], giving an approximately Gaussian distribution. Similarly, an L_2 norm on patches leads to a Gaussian mixture distribution. In that case, the exponential constant λ in the observation likelihood is interpreted as $\lambda = \frac{1}{2\sigma^2}$, where σ is an image-plane distance constant, and the partition function is $Z \propto \sigma^d$. From this, it can be shown (see appendix) that the chamfer distance $\rho|\tilde{z} \equiv \rho(z, \tilde{z})$ is a $\sigma^2\chi^2_d$ random variable (*i.e.*, ρ/σ^2 has a χ^2_d distribution). This allows the parameters σ, d of the observation likelihood (5) 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 (4), 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 in cases where the training set is, in fact, easy to extract from raw data). To improve the initial alignment, first a datum, z^*_0, is chosen such that

   $$z^*_0 \leftarrow \arg \min_{z^* \in \mathcal{Z}^*} \max_{z' \in \mathcal{Z}^* - \{z^*\}} \rho(z^*, z').$$

   Then,

   $$\alpha^*_t = \arg\min_\alpha \rho(T^{-1}_\alpha z^*_t, z^*_0) \text{ and } x^*_t = T^{-1}_{\alpha^*_t} z^*_t,$$

   minimizing by direct descent.
2. To initialise 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, $t = 1 \ldots T^*$, find the cluster that minimizes the distance from x^*_t to the cluster center:

   $$k_t(x^*_t) = \arg\min_k \rho(x^*_t, \tilde{x}_k). \tag{7}$$

   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:

   $$\tilde{x}_k \leftarrow \arg \min_{x \in \mathcal{C}_k} \max_{x' \in \mathcal{C}_k - \{x\}} \rho(x, x'). \tag{8}$$

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 are analogous to the iterative computation of cluster centers in the K-means algorithm, but adapted here to work in spaces where it is impossible to compute a cluster mean. Instead, an existing member of the training set is chosen by a minimax distance computation, 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:

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

Now, following Section 3.3, we treat the distances

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

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

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

which after manipulation of expressions for the χ^2 mean and variance, give rise to estimates for d_k and σ_k.

$$d_k = \frac{\bar{\rho}_k^2}{\bar{\rho^2}_k - \bar{\rho}_k^2} \text{ and } \sigma_k = \sqrt{\bar{\rho}_k / d}. \tag{9}$$

Alternatively, the full maximum likelihood solution, complete with integer constraint on d, yields σ values exactly as above, and integer $d \geq 1$ as the smallest value for which $L'(d) < 0$, where (dropping the k-subscripts for simplicity)

$$L'(d) = (d+1) \log \frac{d+1}{d} - 1 - \log(\bar{\rho}_a / \bar{\rho}_g) \tag{10}$$

is the differential log-likelihood for the model and $\bar{\rho}_a$, $\bar{\rho}_g$ are respectively the arithmetic and geometric means of the ρ-samples. [Notes: 1) If $\bar{\rho}_a / \bar{\rho}_g > 4/e$ the solution for d is the trivial $d = 1$. 2) This estimation procedure is equivalent to fitting a Γ-distribution to d_k.] The value of d captures the effective dimensionality of the local space in which exemplars exist. As $\bar{\rho}_k$ increases, so does d – this is consistent with the statistician's intuition that Gaussians in higher-dimensional spaces hold more of their "weight" in the periphery than their lower-dimensional counterparts.

4.3 Learning Dynamics

In line with recent developments in probabilistic tracking [5], 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 $p(X_t|X_{t-1})$:

1. a Markov matrix M for $p(k_t|k_{t-1})$, learned by histogramming transitions;
2. a first order auto-regressive process (ARP) for $p(\alpha_t|\alpha_{t-1})$, with coefficients calculated using the Yule-Walker algorithm [13].

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 track 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 [7, 21] have difficulty tracking lip and tongue articulation).

5.1 Tracking Human Motion

Figure 3. A randomly generated sequence using only learned dynamics. Edges shown represent the contours of model exemplars.

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 handful of exemplars are shown in Figure 3. To the extent that topology fluctuates within a given mixture component, the linearity assumption of Section 3.1 is met only approximately.

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 on the basis of dynamics alone. The full sequence is available as `generatd.mpg`.[1]

Validity of the M^2 model

An essential assumption of the M^2 approach is that the d values computed from Equation 9 give rise to reasonable

[1] All movie files mentioned in this paper are available at `http://research.microsoft.com/vision/papers/ICCV2001ToyamaBlake`.

Figure 4. Cropped, sample frames from a tracked sequence. The person tracked does not appear in the training sequences.

partition functions. We tested the suitability of this assumption for the chamfer distance by conducting experiments on synthetically generated ellipses with up to 4 degrees of freedom. Results given in Figure 5 support the argument that d

Object	Avg. Cluster Size	Actual DOF of Curve	d	σ
Synthesized ellipse	100	1	0.8	89.1
	100	2	1.2	93.3
	100	3	1.5	86.4
	100	4	3.6	71.9
Person contour	5	?	2.8	21.6
	10	?	4.1	14.4
	20	?	5.1	18.3
	40	?	5.0	17.9

Figure 5. Computed d values using the chamfer distance.

can be computed from training data alone, given a reasonable distance function, and that d does in fact correlate with the degrees of freedom of curve variation.

The table also shows values of d for the pedestrian exemplars. Note that dimensionality increases with cluster size up to a point, but it eventually converges to $d \approx 5$. We read this as assurance that d is a function of the local dimensionality rather than of cluster size.

Practical Tracking

We can now compute observation likelihoods as in Equation 5 and track using the following Bayesian framework. A classical forward algorithm [22] would give $p_t(X_t) \equiv p(X_t|Z_1, \ldots, Z_t)$ as:

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

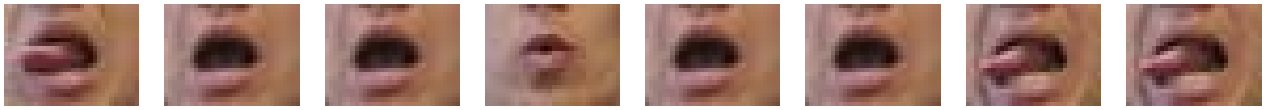
where $p(z|X)$ is computed according to Equation 5. Exact inference is infeasible given that α is real-valued, so the integral is performed using a form of particle filter [14, 17]. To display results, we calculate $\hat{X}_t = \arg\max p_t(X_t)$.

Figure 1 shows cropped, sample images of tracking on a sequence that was not in the training sequence (see also, `walk1.mpg`). Tracking in this case is straightforward and accurate. Figure 4 shows the same exemplar set (trained on one person) used to track a different person entirely (see `walk3.mpg`). 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 `walk3occ.mpg`, 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.

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 30Hz. We tested on a longer test sequence of 570 frames (of which 270 are shown



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6. 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.

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.

The results for this experiment can be seen in video format: `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. In these video files, 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 crumbles, whereas tracking based on the shuffle distance remains largely successful.

Figure 6 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 [12] with a probabilistic framework [11] 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 [12, 26], 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. See `ballet0.mpg` for preliminary results with larger exemplar sets (training and test sets here are the same).
- The current clustering algorithm can be extended to a "soft" EM-like algorithm by assigning exemplar membership probabilities based on d values computed at each step.
- Our results make it clear that the M^2 approach works for some non-metric distances. One open question is to determine to what extent metric assumptions can be violated.

Acknowledgments

We thank P. Anandan, Brendan Frey, Nebojsa Jojic, Neil Lawrence, and Chris Williams for stimulating discussions; John MacCormick kindly provided video data.

References

[1] A. Amini, S. Tehrani, and T. Weymouth. Using dynamic programming for minimizing the energy of active contours

in the presence of hard constraints. In *Proc. 2nd Int. Conf. on Computer Vision*, pages 95–99, 1988.

[2] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[3] B. Bascle and R. Deriche. Region tracking through image sequences. In *Proc. 5th Int. Conf. on Computer Vision*, pages 302–307, Boston, Jun 1995.

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

[5] A. Blake and M. Isard. *Active contours*. Springer, 1998.

[6] M. Brand. Shadow puppetry. In *Proc. Int. Conf. on Computer Vision*, pages 1237–1244, 1999.

[7] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. European Conf. Computer Vision*, pages 484–498, 1998.

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

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

[10] W. Freeman and E. Pasztor. Learning to estimate scenes from images. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

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

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

[13] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

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

[15] G. Hager and K. Toyama. Xvision: combining image warping and geometric constraints for fast tracking. In *Proc. 4th European Conf. Computer Vision*, pages 507–517, 1996.

[16] D. Huttenlocher, J. Noh, and W. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proc. 4th Int. Conf. on Computer Vision*, pages 93–101, 1993.

[17] M. Isard and A. Blake. Visual tracking by stochastic propagation of conditional density. In *Proc. 4th European Conf. Computer Vision*, pages 343–356, Cambridge, England, Apr 1996.

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

[19] K. Kutulakos. Approximate N-view stereo. In *Proc. European Conf. Computer Vision*, volume 1, pages 67–83, 2000.

[20] D. Mumford. Pattern theory: a unifying perspective. In D. Knill and W. Richard, editors, *Perception as Bayesian inference*, pages 25–62. Cambridge University Press, 1996.

[21] H. Neven and E. Interfaces. In *Siggraph Demo Session*. Los Angeles, 2000.

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

[23] G. Storvik. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(10):976–986, 1994.

[24] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. In A. Blake and A. Yuille, editors, *Active Vision*, pages 3–20. MIT, 1992.

[25] T. Vetter and T. Poggio. Image synthesis from a single example image. In *Proc. 4th European Conf. Computer Vision*, pages 652–659, Cambridge, England, Apr 1996.

[26] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proc. ACM Siggraph*. ACM, 2000.

Appendix

Quadratic Chamfer distance has a scaled χ^2 distribution. We have, from (6),

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

From (3),

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

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 [5] which can be diagonalized as $\mathcal{H} = UDU^\top$, in which U is orthogonal and D is diagonal. Now, from (4), and using the normalization properties of Gaussians,

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

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

$$\mathbf{y} = B\mathbf{w} \text{ where } \mathbf{w} \sim \mathcal{N}(0, I_d) \text{ and}$$

$$B = \sigma\mathcal{H}^{-1/2} = \sigma U D^{-1/2} U^\top.$$

Finally,

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

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