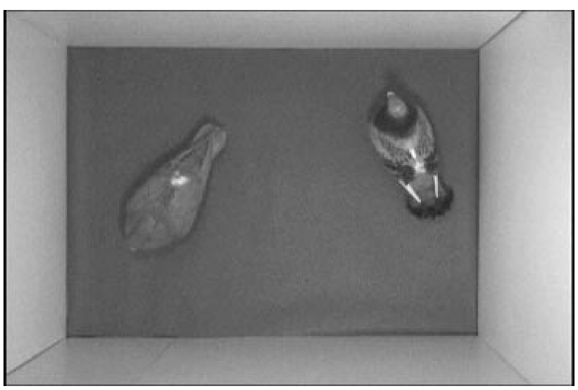
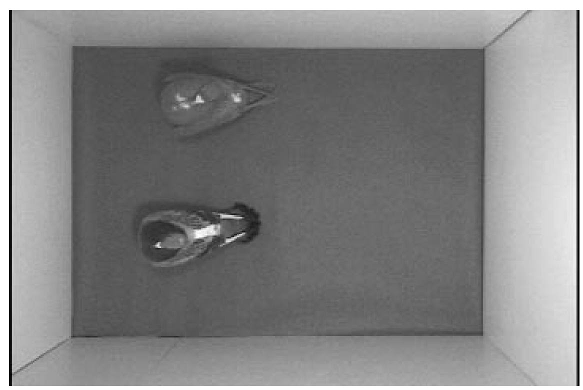
Image Appearance Models

Task: Model the detailed image appearance of a class of objects under a wide range of viewing conditions.

Motivation: Such a model could be used for:

- tracking,
- detection,
- recognition (of object's class or of the individual).

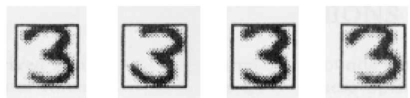



Difficulty: Even with restricted viewpoints, poses, and lighting conditions, the image appearance of an object can vary significantly over time.

For example, a single image template (from the first frame of a sequence) is unlikely to provide a good fit to the appearance of the pigeons over a lengthy sequence.

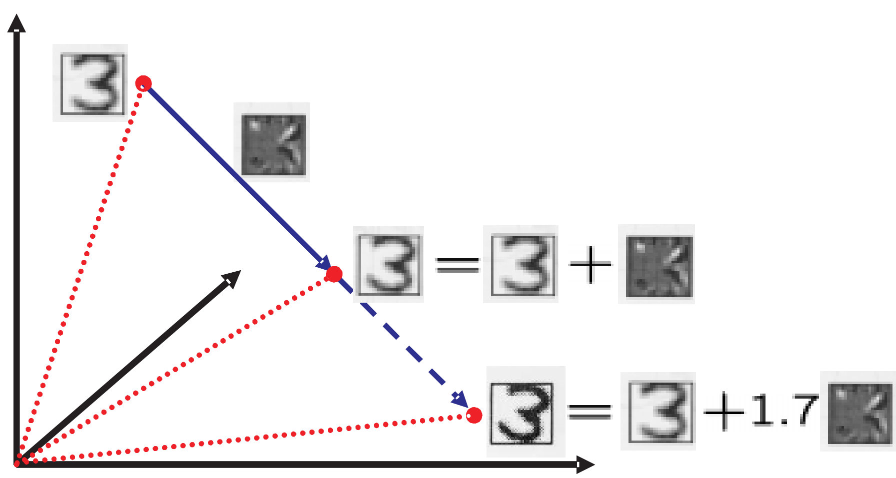
Simple Approach: Subspace Appearance Models

Idea: Images are not random, especially those of an object, or similar objects, under different viewing conditions.



Rather, than storing every image, we might try to represent the images more effectively, e.g., in a lower dimensional *subspace*.

For example, let's represent each $N \times N$ image as a point in an N^2-dim vector space (e.g., ordering the pixels lexicographically to form the vectors).



(red points denote images, blue vectors denote image differences)

How do we find a low-dimensional basis to accurately model (approximate) each image of the training ensemble (as a linear combination of basis images)?

Linear Subspace Models

We seek a linear basis with which each image in the ensemble is approximated as a linear combination of basis images $b_k(\vec{\mathbf{x}})$

$$I(\vec{\mathbf{x}}) \;\approx\; \sum_{k=1}^{K} a_k \, b_k(\vec{\mathbf{x}}) \tag{1}$$

The *subspace coefficients* $\vec{\mathbf{a}} = (a_1, ..., a_K)$ comprise the representaion.

With some abuse of notation, assuming basis images $b_k(\vec{\mathbf{x}})$ with N^2 pixels, let's define

- $\vec{\mathbf{b}}_k$ – an $N^2 \times 1$ vector with pixels arranged in lexicographic order
- $\mathbf{B}$ – a matrix with columns $\vec{\mathbf{b}}_k$, *i.e.*, $\mathbf{B} = [\vec{\mathbf{b}}_1, ..., \vec{\mathbf{b}}_K] \in \mathcal{R}^{N^2 \times K}$

With this notation we can rewrite Eq. (1) in matrix algebra as

$$\vec{\mathbf{I}} \;\approx\; \mathbf{B}\,\vec{\mathbf{a}} \tag{2}$$

Choosing The Basis

Orthogonality: Let's assume orthogonal basis functions,

$$\| \vec{\mathbf{b}}_k \|_2 \;=\; 1 \quad , \qquad \vec{\mathbf{b}}_j^{\,T} \vec{\mathbf{b}}_k \;=\; \delta_{jk} \; .$$

Subspace Coefficients: It follows from the linear model in Eq. (2) and the orthogonality of the basis functions that

$$\vec{\mathbf{b}}_k^{\,T} \vec{\mathbf{I}} \;\approx\; \vec{\mathbf{b}}_k^{\,T} \mathbf{B}\, \vec{\mathbf{a}} \;=\; \vec{\mathbf{b}}_k^{\,T} [\vec{\mathbf{b}}_1, ..., \vec{\mathbf{b}}_K]\, \vec{\mathbf{a}} \;=\; a_k$$

This selection of coefficients, $\vec{\mathbf{a}} = \mathbf{B}^T \vec{\mathbf{I}}$, minimizes the sum of squared errors (or sum of squared pixel differences, SSD):

$$\min_{\vec{\mathbf{a}} \in \mathcal{R}^K} \| \vec{\mathbf{I}} - \mathbf{B}\, \vec{\mathbf{a}} \|_2^2$$

Basis Images: In order to select the basis functions $\{\vec{\mathbf{b}}_k\}_{k=1}^K$, suppose we have a training set of images

$$\{ \vec{\mathbf{I}}_l \}_{l=1}^L \quad , \quad \text{with } \; L \gg K$$

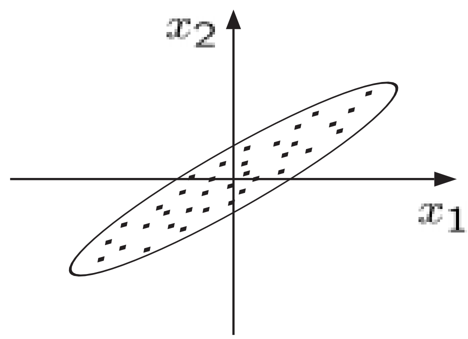
(Let's also assume the images are mean zero. If the mean is nonzero, subtract the mean image, $\frac{1}{L} \sum_l \vec{\mathbf{I}}_l$, from each training image.)

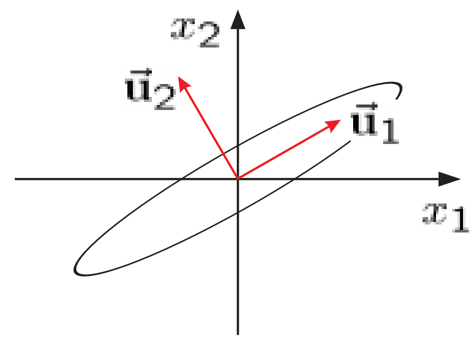
Finally, let's select the basis, $\{\vec{\mathbf{b}}_k\}_{k=1}^K$, to minimize squared reconstruction error:

$$\sum_{l=1}^{L} \min_{\vec{\mathbf{a}}_l} \| \vec{\mathbf{I}}_l - \mathbf{B}\, \vec{\mathbf{a}}_l \|_2^2$$

Intuitions

Example: let's consider a set of images $\{\vec{\mathbf{I}}_l\}_{l=1}^{L}$, each with only two pixels. So, each image can be viewed as a 2D point, $\vec{\mathbf{I}}_l \in \mathcal{R}^2$.





For a model with only one basis image, what should $\vec{\mathbf{b}}_1$ be?

Approach: Fit an ellipse to the distribution of the image data, and choose $\vec{\mathbf{b}}_1$ to be a unit vector in the direction of the major axis.

Define the ellipse as $\vec{\mathbf{x}}^T C^{-1} \vec{\mathbf{x}} = 1$, where C is the sample covariance matrix of the image data,

$$\mathbf{C} = \frac{1}{L} \sum_{l=1}^{L} \vec{\mathbf{I}}_l \vec{\mathbf{I}}_l^T$$

The eigenvectors of $\mathbf{C}$ provide the major axis, i.e.,

$$\mathbf{C}\,\mathbf{U} \;=\; \mathbf{U}\,\mathbf{D}$$

for orthogonal matrix $\mathbf{U} = [\vec{\mathbf{u}}_1, \vec{\mathbf{u}}_2]$, and diagonal matrix $\mathbf{D}$ with elements $d_1 \geq d_2 \geq 0$. The direction $\vec{\mathbf{u}}_1$ associated with the largest eigenvalue is the direction of the major axis, so let $\vec{\mathbf{b}}_1 = \vec{\mathbf{u}}_1$.

Principal Components Analysis

Theorem: *(Minimum reconstruction error)* The orthogonal basis $\mathbf{B}$, of rank $K < N^2$, that minimizes the squared reconstruction error over training data, $\{\vec{\mathbf{I}}_l\}_{l=1}^L$, i.e.,

$$\sum_{l=1}^{L} \min_{\vec{\mathbf{a}}_l} \| \vec{\mathbf{I}}_l - \mathbf{B}\,\vec{\mathbf{a}}_l \|_2^2$$

is given by the first K eigenvectors of the data covariance matrix

$$\mathbf{C} = \frac{1}{L} \sum_{l=1}^{L} \vec{\mathbf{I}}_l \, \vec{\mathbf{I}}_l^{\,T} \;\in\; \mathcal{R}^{N^2 \times N^2} \;\;, \quad \text{for which} \quad \mathbf{C}\,\mathbf{U} \;=\; \mathbf{U}\,\mathbf{D}$$

where $\mathbf{U} = [\vec{\mathbf{u}}_1, ..., \vec{\mathbf{u}}_{N^2}]$ is orthogonal, and $\mathbf{D} = \text{diag}(d_1, ..., d_{N^2})$ with $d_1 \geq d_2 \geq \, ... \, \geq d_{N^2}$.

That is, the optimal basis vectors are $\vec{\mathbf{b}}_k = \vec{\mathbf{u}}_k$, for $k = 1...K$. The corresponding basis images $\{b_k(\vec{\mathbf{x}})\}_{k=1}^K$ are often called eigen-images.

Proof: see the derivation below.

Derivation of PCA

To begin, we want to find $\mathbf{B}$ in order to minimize squared error in subspace approximations to the images of the training ensemble.

$$E = \sum_{l=1}^{L} \min_{\vec{\mathbf{a}}_l} \| \vec{\mathbf{I}}_l - \mathbf{B}\,\vec{\mathbf{a}}_l \|_2^2$$

Given the assumption that the columns of $\mathbf{B}$ are orthonormal, the optimal coefficients are $\vec{\mathbf{a}}_l = \mathbf{B}^T\vec{\mathbf{I}}_l$, so

$$E = \sum_{l=1}^{L} \min_{\vec{\mathbf{a}}_l} \| \vec{\mathbf{I}}_l - \mathbf{B}\,\vec{\mathbf{a}}_l \|_2^2 = \| \vec{\mathbf{I}}_l - \mathbf{B}\,\mathbf{B}^T\vec{\mathbf{I}}_l \|_2^2 \quad (3)$$

Furthermore, writing the each training image as a column in a matrix $\mathbf{A} = \left[\vec{\mathbf{I}}_1, ..., \vec{\mathbf{I}}_L\right]$, we have

$$E = \sum_{l=1}^{L} \| \vec{\mathbf{I}}_l - \mathbf{B}\,\mathbf{B}^T\vec{\mathbf{I}}_l \|_2^2 = \| \mathbf{A} - \mathbf{B}\,\mathbf{B}^T\mathbf{A} \|_F^2 = trace\left[\mathbf{A}\,\mathbf{A}^T\right] - trace\left[\mathbf{B}^T\mathbf{A}\,\mathbf{A}^T\mathbf{B}\right]$$

You get this last step by expanding the square and noting $\mathbf{B}^T\mathbf{B} = \mathbf{I}_K$, and using the properties of $trace$, e.g., $trace[\mathbf{A}] = trace[\mathbf{A}^T]$, and also $trace[\mathbf{B}^T\mathbf{A}\mathbf{A}^T\mathbf{B}] = trace[\mathbf{A}^T\mathbf{B}\mathbf{B}^T\mathbf{A}]$.
So the minmize the average squared error in the approximation we want to find $\mathbf{B}$ to maximize

$$E' = trace\left[\mathbf{B}^T\mathbf{A}\,\mathbf{A}^T\mathbf{B}\right] \quad (4)$$

Now, let's use the fact that for the data covariance, $\mathbf{C}$ we have $\mathbf{C} = \frac{1}{L}\mathbf{A}\,\mathbf{A}^T$. Moreover, as defined above the SVD of $\mathbf{C}$ can be written as $\mathbf{C} = \mathbf{U}\,\mathbf{D}\,\mathbf{U}^T$. So, let's substitute the SVD into E':

$$E' = trace\left[\mathbf{B}^T\mathbf{U}\,\mathbf{D}\,\mathbf{U}^T\mathbf{B}\right] \quad (5)$$

where of course $\mathbf{U}$ is orthogonal, and $\mathbf{D}$ is diagonal.

Now we just have to show that we want to choose $\mathbf{B}$ such that the trace strips off the first K elements of $\mathbf{D}$ to maximize E'. Intuitively, note that $\mathbf{B}^T\mathbf{U}$ must be rank K since $\mathbf{B}$ is rank K. And note that the diagonal elements of $\mathbf{D}$ are ordered. Also the trace is invariant under matrix rotation. So, the highest rank K trace we can hope to get is by choosing $\mathbf{B}$ so that, when combined with $\mathbf{U}$ we keep the first K columns of $\mathbf{D}$. That is, the columns of $\mathbf{B}$ should be the first K orthonormal rows of $\mathbf{U}$. We need to make this a little more rigorous, but that's it for now...

Other Properties of PCA

Maximum Variance: The K-D subspace approximation captures the greatest possible variance in the training data.

- For $a_k = \vec{\mathbf{b}}_k^T \vec{\mathbf{I}}$, and $\vec{\mathbf{a}} = (a_1, ..., a_K)$, the subspace coefficient covariance is $\mathrm{E}[\vec{\mathbf{a}}\,\vec{\mathbf{a}}^T] = \mathrm{diag}(d_1, ..., d_K)$. That is, the diagonal entries of $\mathbf{D}$ are marginal variances of the subspace coefficients:

  $$\sigma_k^2 \equiv \mathrm{E}[a_k^2] = d_k \, .$$

  Also note $\mathrm{E}[a_j \, a_k] = 0$ for $j \neq k$, so distinct coefficients are uncorrelated.

- The total variance V in the training set (recall the images are zero mean) is the sum of the squared pixel responses over all images. This is equivalent to $V = \sum_{k=1}^{N^2} \sigma_k^2$.
- Total variance *explained* by the optimal K-dimensional basis B is $\sum_{k=1}^{K} \sigma_k^2$. Similarly the residual *unexplained* variance is $\sum_{k=K+1}^{N^2} \sigma_k^2$.
- Expressing these terms as fractions of the total variance, we have:

$$dQ_k \equiv \sigma_k^2 / V, \text{ fraction of variance explained by } k^{th} \text{ basis vector,}$$

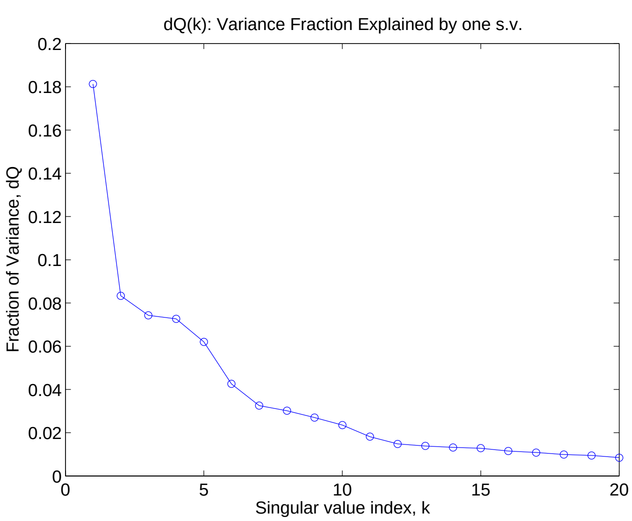
$$Q_K \equiv \sum_{k=1}^{K} \sigma_k^2 / V, \text{ fraction of variance explained by subspace } B.$$

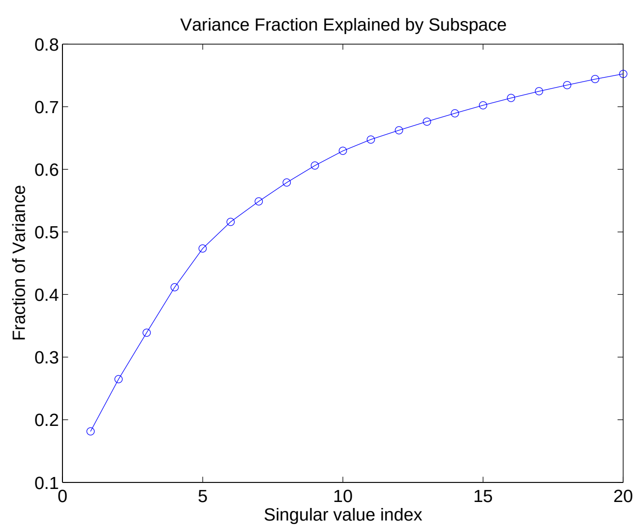
Eye Subspace Model

Subset of 1196 eye images (25×20):

Left Eyes Right Eyes

Variance captured:



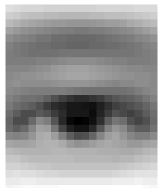


Left plot shows dQ_k, the marginal variance for each principal direction divided by the total variance in the training data, as a function of the singular value index k.

Right plot shows Q_k the fraction of the total variance captured by the principal subspace $\mathbf{B}$, as a function of the subspace dimension K.

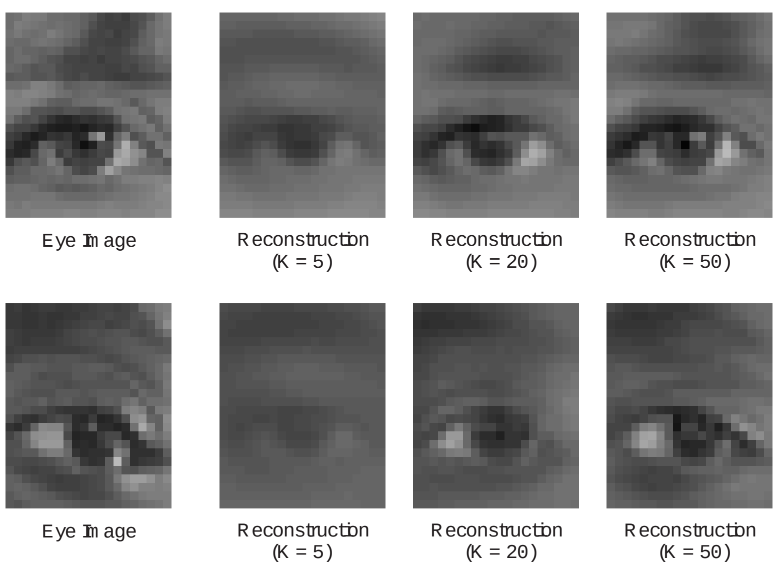
Eye Subspace Model

Mean Eye:



Basis Images $(1-6,\ \text{ and }\ 10\!:\!5\!:\!35)$**:**

Reconstructions (for $K = 5, 20, 50$)**:**

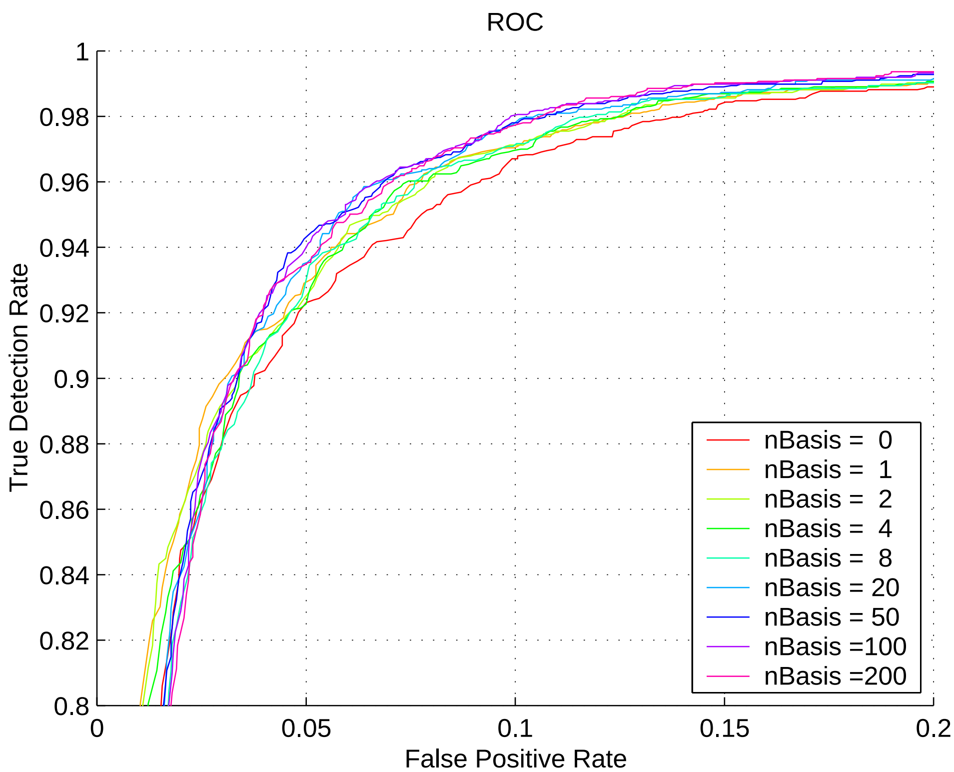


Eye Detection

The PCA analysis can be used to develop a simple Gaussian model for eye images, which then motivates an eye detector based on (essentially) the residual squared error (see the utvis eigenTut tutorial).

ROC Curves:

- true detection rate vs false positive rate,
- trade-off (as a function of a decision threshold) between detection rate and specificity.



Here the eye images in the test set were different from the those in the training set. Non-eyes were drawn at random from textured regions in images.

Eigen-Eye Detection Examples

A 5% false positive rate provides a weak detector which will respond on many non-eyes. This could be combined with other sources of information (eg. skin colour, motion, blink detectors) in order to provide a more specific detector.

EigenTracking

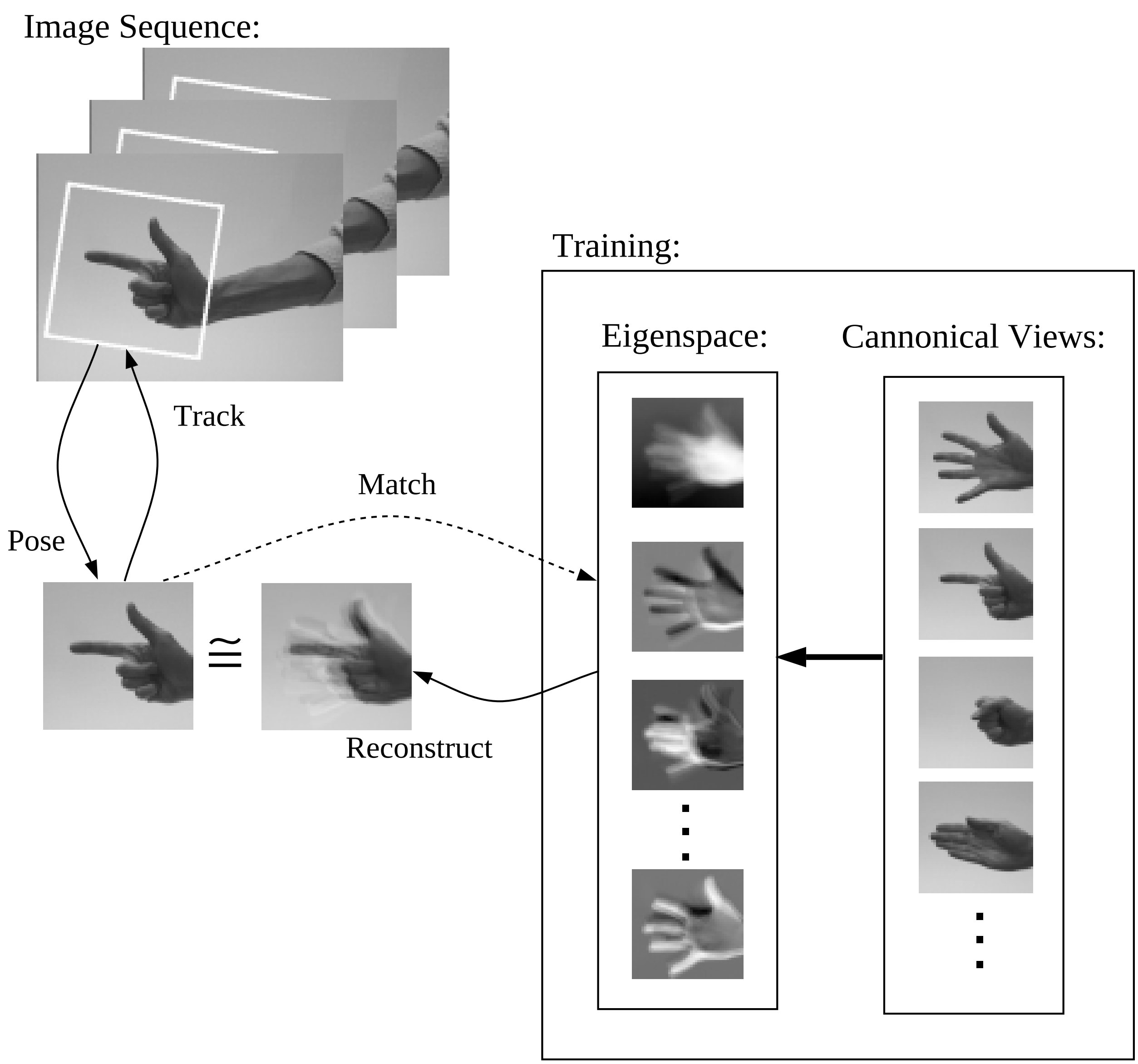
Problems with eigen-matching:

1. Sensitive to unmodelled variations (e.g. backgrounds or occlusions).
2. Testing every image patch, possibly across a range of scales and image orientations, is expensive. Alternatively, the image of the object needs to be segmented and normalized.
3. Lack of specificity, too many false-positives.

The eigentracking paper addresses the first two of these issues by:

1. Introducing a robust error norm, $\rho(\vec{I} - B\vec{a})$,
2. Simultaneously aligns a new image with the eigenspace and determines the appearance coefficients $\vec{a}$ (given a suitable initial guess, say from a previous frame in tracking.)

EigenTracking Overview

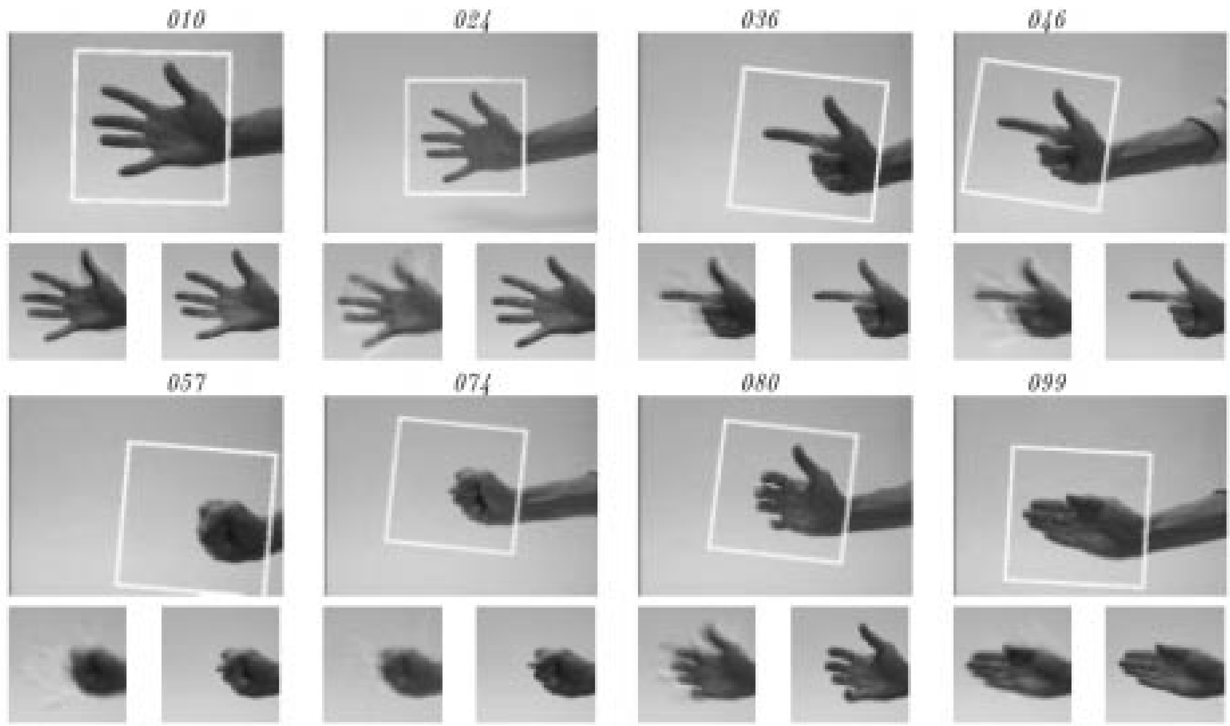


This involves the joint minimization of

$$E(\vec{p}, \vec{a}) \equiv \sum_{\vec{x}} \rho(I(\vec{W}(\vec{x}; \vec{p}), t) - \sum_{k} B_k(\vec{x}) a_k)$$

Here $I(\vec{x}, t)$ is the current image, $\vec{a}$ are the subspace appearance parameters, $\vec{p}$ are the pose parameters, $\vec{W}(\vec{x}; \vec{p})$ is the image warp (e.g. affine), and $\rho(e)$ is a robust error norm.

EigenTracking Results



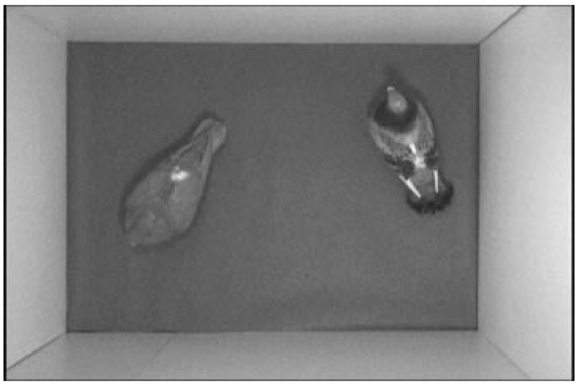
Implementation Details.

- Gradient-based image matching, similar to motion estimation.
- Coarse to fine processing to deal with large motions. Eigen-pyramids.
- A $K = 25$ dimensional basis was used.
- Here $\rho(e) \equiv e^2$ for processing speed, since the background is uncluttered.
- In order to handle a cluttered background a redescending estimator should be used. However, difficulties can be expected when the hand covers a small part of a cluttered frame.

EigenTracking Discussion

Discussion Points

1. Would you expect eigen-tracking to work on the pigeon sequence? What would be involved?




2. Would you expect the same approach to work on the Bahen hallway sequence? Explain.

3. What is the trade-off between explicitly representing pose variability using the warps, versus restricting the warps and treating variations in poses as generic image variations? For example:

   (a) Suppose we used eigen-models to represent the pigeons in some canonical coordinate frame, but allowed their image orientation to vary (e.g. we used only image translation for the warps $\vec{W}(\vec{x}, \vec{p})$). How would that effect the results?

   (b) Suppose we removed the warp entirely, and used one eigen-model to represent the entire frame in the pigeon sequence (possibly at lower resolution). How could we use the results for tracking?

4. In a cluttered environment it might be useful to use eigentracking in a particle filtering framework. What are some of the sources of difficulty you might expect?

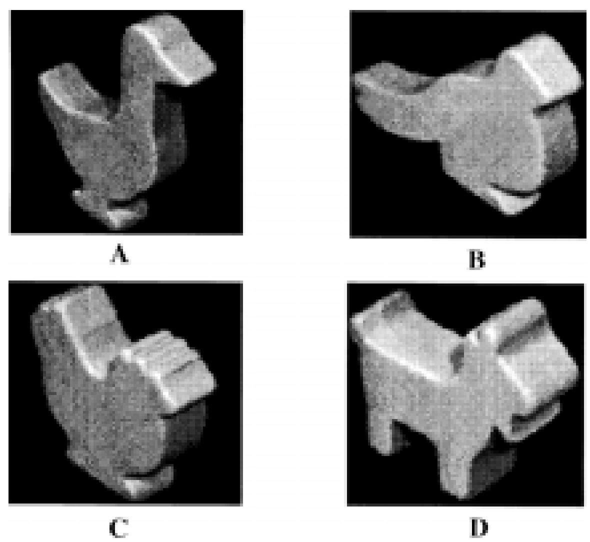
5. Could we learn the eigen-space appearance model (with warps) on the fly?

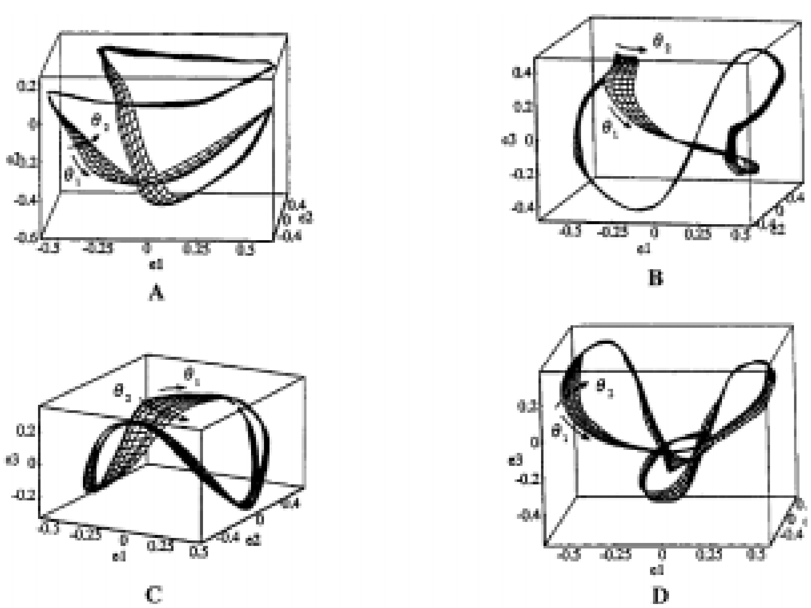
6. How do these approaches for learning 2D image models compare with those for learning 3D models?

Object Recognition and Pose Identification

Murase and Nayar (1995)

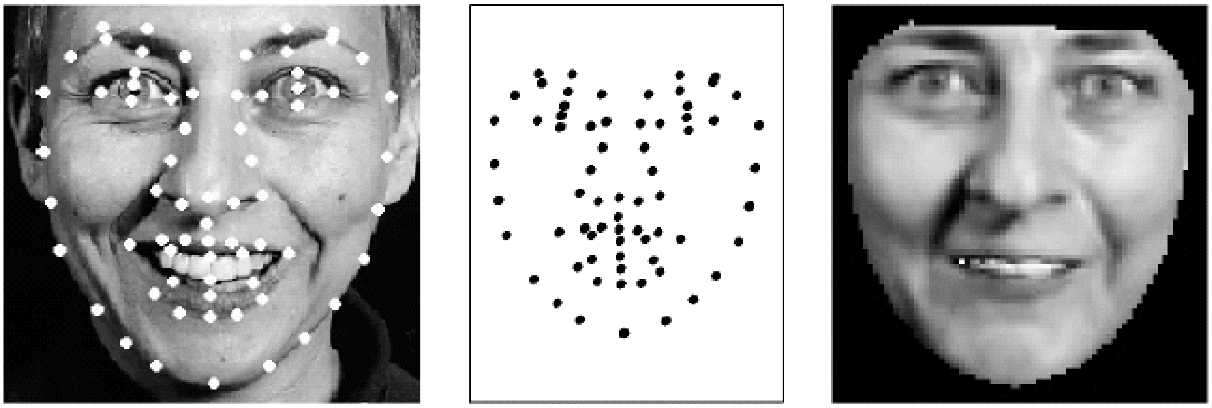
- images of multiple objects, taken with different camera positions (θ_1) and lighting directions (θ_2).
- each object occupies a manifold in the subspace (as a function of θ_1, θ_2).
- recognition: given a segmented and normalized object, nearest neighbour on manifold found, providing pose and lighting information.





Active Appearance Models

An example of learning a richer family of image warps together with an appearance model is provided by the work of Cootes, Taylor, and colleagues.



Labelled image Points Shape-free patch

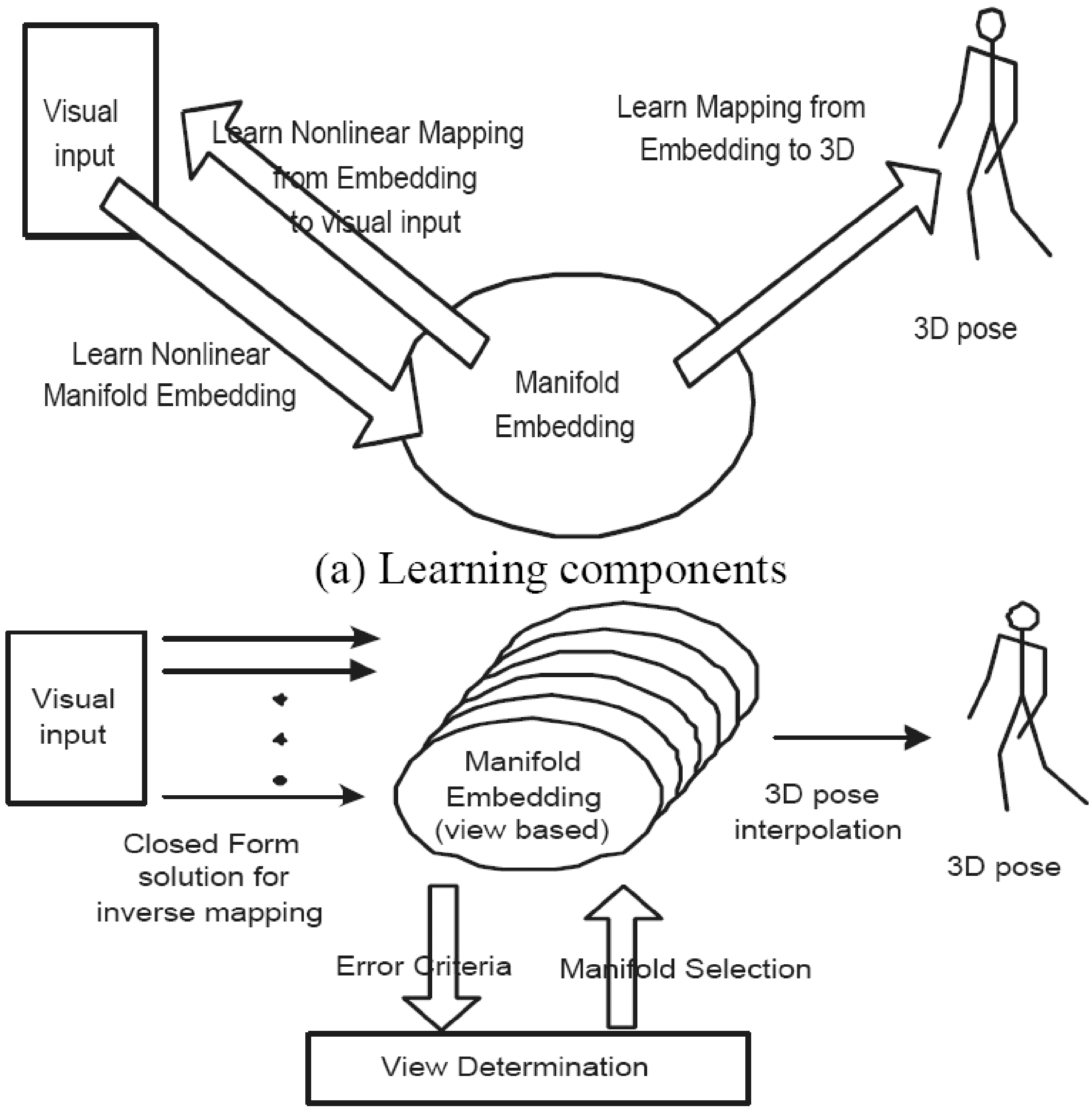
Eigen-models are used to represent both the pose and the appearance variability. As in eigen-tracking, both the pose and appearance parameters are fit to new images.

Initial 3 its 8 its 11 its Converged Original

Pose Identification from Silhouettes

Elgammal and Lee (2004)

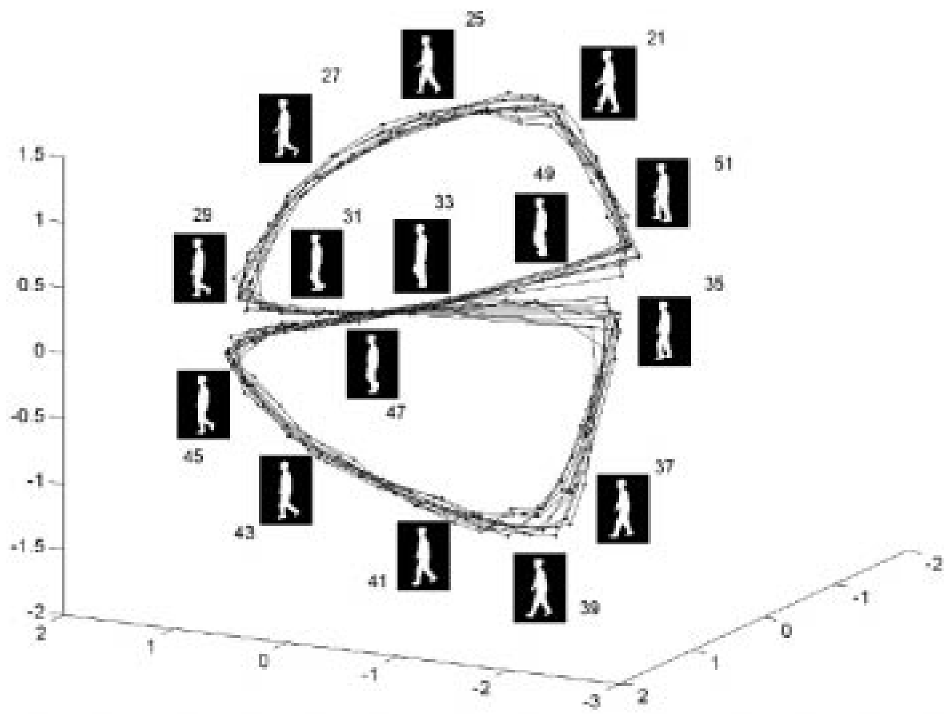
- A low dimensional model for silhouettes of a walking human figure are learned (using non-linear embedding),
- The mappings from this embedded manifold to silhouettes, along with a mapping from the embedded manifold to pose space, are interpolated.

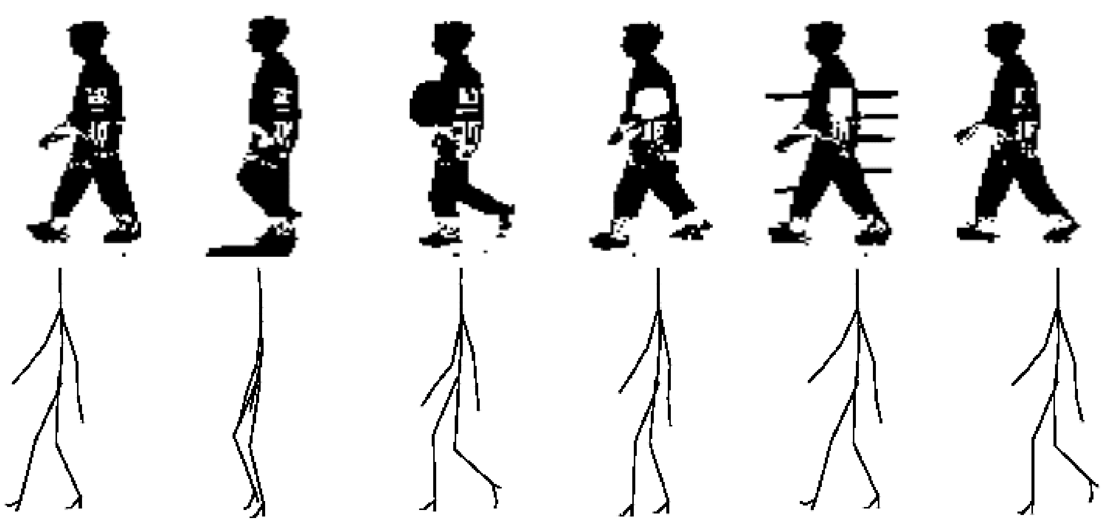


(b) pose estimation.

Pose Identification from Silhouettes, Cont.

Pose identification: Given a segmented and normalized difference from background image, a nearest neighbour on the manifold is found.





EigenTracking with a Particle Filter

Khan, Balch, and Dellaert [2004] define the state to be estimated at time t to be

$$\vec{x}_t \equiv \begin{pmatrix} \vec{l}_t \\ \vec{a}_t \end{pmatrix}$$

where $\vec{l}_t$, $\vec{a}_t$ are the pose and appearance parameters, respectively, for an appearance model.

A standard particle filter could be applied to this state;

Pro We represent the specific appearance of the object being tracked and its variablility over time.

Pro An appearance model adapted to recent past data should improve tracking in clutter.

Con A key disadvantage is the increase in the state dimension over just using 2D location (pose) parameters. The discrete approximation of the distributions is therefore considerably tougher.

Hybrid Particles. Suppose we use a discrete approximation for the pose variables $\vec{l}_t$, but we maintain a Gaussian distribution for the appearance model. That is, the i^{th} particle has location $\vec{l}_t^{\,i}$ and the appearance parameters are distributed according to the normal distribution $N(\vec{a}_t | \vec{\mu}_t^{\,i}, \Sigma_t^i)$ (see the example on the next page).

Is the recursive estimation of $(\pi_t^{\,i}, \vec{l}_t^{\,i}, \vec{\mu}_t^{\,i}, \Sigma_t^i)$ feasible? (Here π denotes the particle weight.)

Gaussian Appearance Model

Consider the previous PCA eye model. A possible generative model, $\mathcal{M}$, for random eye images is:

$$\vec{\mathbf{I}} \;=\; \vec{\mathbf{m}} + \left(\sum_{k=1}^{K} a_k \vec{\mathbf{b}}_k \right) \;+\; \vec{\mathbf{e}}$$

where $\vec{\mathbf{m}}$ is the mean eye image, $\vec{\mathbf{e}} \sim \mathcal{N}(0, \sigma_e^2 \, \mathbf{I}_{N^2})$ where $\sigma_e^2 = \frac{1}{N^2} \sum_{k=K+1}^{N^2} \sigma_k^2$ is the per pixel out-of-subspace variance, $a_k \sim \mathcal{N}(0, \sigma_k^2)$, where $\sigma_k^2 + \sigma_e^2$ is the sample variance associated with the k^{th} principal direction in the training data. (This model matches the first and second order statistics of the data within the subspace.)

Random Eye Images:

Random draws from generative model (with K = 5,10,20,50,100,200)

So the likelihood of an image of a eye given this model $\mathcal{M}$ is

$$p(\vec{\mathbf{I}} \,|\, \mathcal{M}) \;=\; \left(\prod_{k=1}^{K} p(a_k|\mathcal{M}) \right) \; p(\vec{\mathbf{e}} \,|\, \mathcal{M})$$

where

$$p(a_k|\mathcal{M}) \;=\; \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{a_k^2}{2\sigma_k^2}} \quad , \qquad p(\vec{\mathbf{e}} \,|\, \mathcal{M}) = \prod_{j=1}^{N^2} \frac{1}{\sqrt{2\pi}\sigma_e} e^{-\frac{e_j^2}{2\sigma_e^2}} \quad .$$

Recursive Hybrid Particle Filter

Recursive Filtering

$$p(\vec{x}_t|\vec{Z}_{1:t}) = kp(\vec{Z}_t|\vec{x}_t)\int_{\vec{x}_{t-1}} p(\vec{x}_t|\vec{x}_{t-1})p(\vec{x}_{t-1}|\vec{Z}_{1:t-1}).$$

Marginalizing over appearance $\vec{a}_t$ and using $\vec{x}_t \equiv (\vec{a}_t, \vec{l}_t)$,

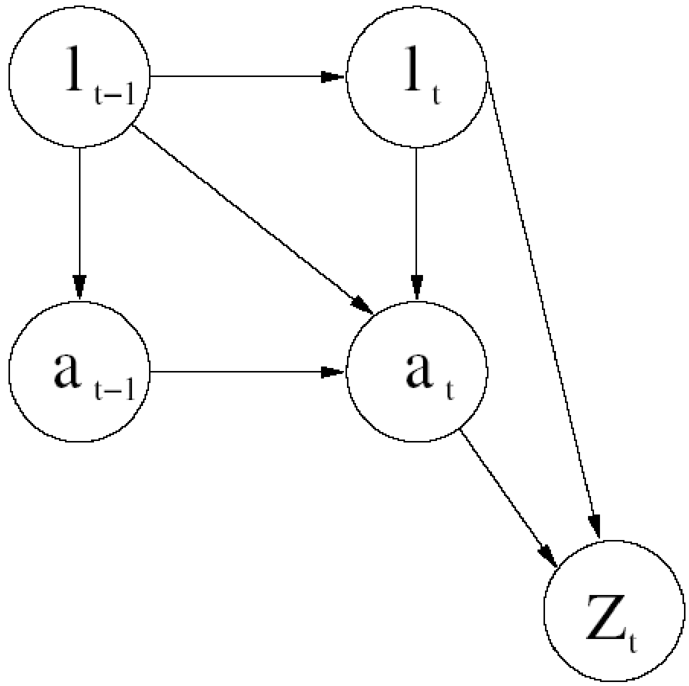
$$p(\vec{l}_t|\vec{Z}_{1:t}) = k\int_{\vec{a}_t} p(\vec{Z}_t|\vec{l}_t,\vec{a}_t)\times$$
$$\int_{\vec{l}_{t-1}}\int_{\vec{a}_{t-1}} p(\vec{l}_t|\vec{l}_{t-1},\vec{a}_{t-1})p(\vec{a}_t|\vec{l}_t,\vec{l}_{t-1},\vec{a}_{t-1})p(\vec{l}_{t-1},\vec{a}_{t-1}|\vec{Z}_{1:t-1}).$$

The integral over $\vec{l}_{t-1}$ will be replaced by the sum over weighted particles.

Rao-Blackwellization. We could also use samples to approximate the integrals over the appearance variables (but, according to the Rao-Blackwell theorem, this would increase the variance of the estimator). Instead we wish to analytically integrate over the appearance variables. Specifically,

- $p(\vec{l}_{t-1},\vec{a}_{t-1}|\vec{Z}_{1:t-1})$ is the estimate of the previous posterior, sampled in $\vec{l}_t$ and Gaussian in $\vec{a}_{t-1}$.
- The term $p(\vec{a}_t|\vec{l}_t,\vec{l}_{t-1},\vec{a}_{t-1})$ is modeled by a Gaussian distribution over $\vec{a}_t$ (larger pose changes $\vec{l}_t - \vec{l}_{t-1}$ are taken to increase the variance).
- $p(\vec{l}_t|\vec{l}_{t-1},\vec{a}_{t-1})$ ... ouch.

Recursive Hybrid Particle, Cont.



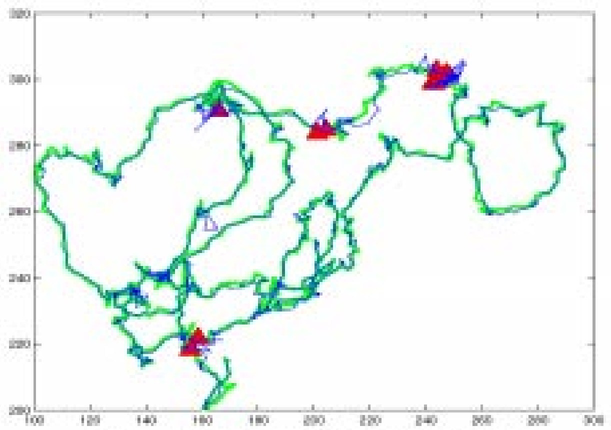
The previous integrals become a product of Gaussians (and tractable) if we assume

$$p(\vec{l}_t|\vec{l}_{t-1}, \vec{a}_{t-1}) = p(\vec{l}_t|\vec{l}_{t-1}),$$

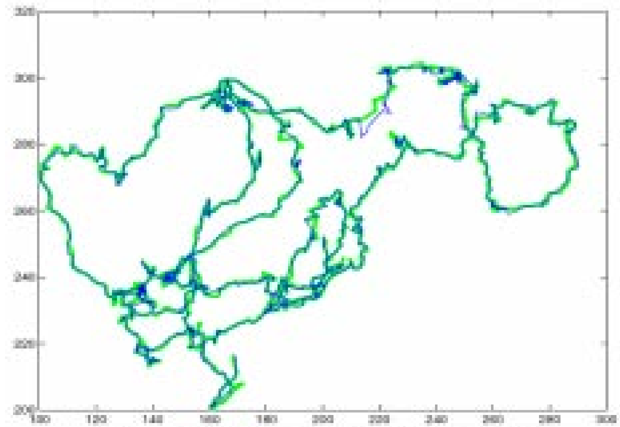
That is, the conditional distribution for the pose at time t depends only on the pose at the previous frame, and not on the appearance at the previous frame. In many situations this is probably an innocuous assumption.

Recursive Hybrid Particle, Results

The results indicate improved tracking when a richer appearance model (12D) is used in the Hybrid filter:

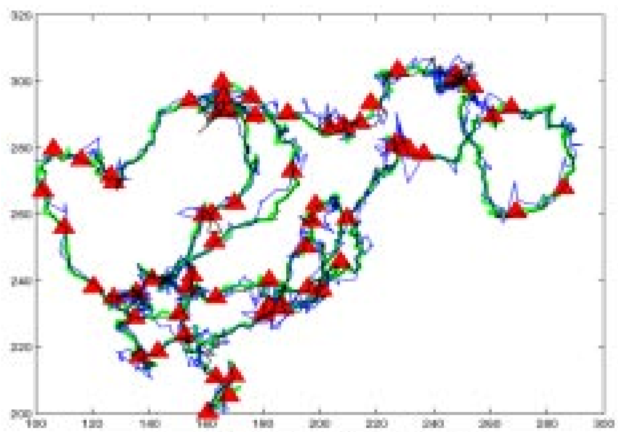


(a) $q = 0$, using 500 hybrid particles



(b) $q = 12$, using 500 hybrid particles

However, when a 12D appearance model is used with the standard particle filter, the increased variance causes more tracking failures (for the same number of particles).



$q = 12$, using 500 particles

Moreover, the paper demonstrates that substantially more particles do not provide results of the quality of the hybrid particle filter approach.

Further Readings

Belhumeur et al (1997) Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(7):711-720

Black, M. and Jepson, A.D. (1998) EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Computer Vision* 26(1):63-84.

Chennubhotla, C. and Jepson, A.D. (2001) Sparse PCA (S-PCA): Extracting multi-scale structure from data. *Proc. IEEE ICCV*, Vancouver, pp. 641-647.

Cootes, T.F., Wheeler, G.V., Walker, K.N. and Taylor, C.J. (2002) View-Based Active Appearance Models, Image and Vision Computing, Vol.20, 2002, pp.657-664.

Elgammal, A. and Lee, C.-S. (2004) Inferring 3D body pose from silhouettes using activity manifold learning, Proc. CVPR.

Golub, G. and van Loan, C. (1984) Matrix Computations. Johns Hopkins Press, Baltimore.

Hager, G. and Belhumeur, P. (1999) Real-time tracking of image region with changes in geometry and illumination, Proc. CVPR.

Khan, Z., Balch, T. and Dellaert, F., A Rao-Blackwellized particle filter for EigenTracking, CVPR 2004.

Murase, H. and Nayar, S. (1995) Visual learning and recognition of 3D objects from appearance. *Int. J. Computer Vision* 14:5–24.

Moghaddam, B., Jebara, T. and Pentland, A. (2000) Bayesian face recognition. *Pattern Recognition*, 33(11):1771-1782

Ross, D., Lim, J. and Yang, M.-H. (2004) Adaptive Probabilistic Visual Tracking with Incremental Subspace Update, Proc. ECCV.

Turk, M. and Pendland, A. (1991) Face recognition using eigenfaces, *J. Cognitive Neuroscience*, 3(1):71–86.