“Stationary” Video

- Laster time --- we considered how to model the consistent appearance of a scene, by capturing the local statistical distribution at each pixel.
- Intensity
- Spatio-temporal derivatives.

- Last time, we made an effort to capture the global regularities in some video sequences.
- This was based on PCA – linear combinations of images (an average image, plus a weighted combination of “overlays” called principle components).

Sometimes, scene variations are not well modeled by principle components. That is, the scene variation is SIMPLER than the principle component analysis would suggest.

Degrees of Freedom

- For a given data set that contains multiple examples of an object, finding the number of degrees of freedom of that object is one of the most important tasks. And it is not always easy.

Suppose that the object is a (column) vector.

- And that a matrix M has many columns (many examples of the object).
- The "rank" of M is the number of linearly independent columns.

- Images are noisy, so even for simple scenes, the number of linearly independent columns is equal to the number of images.

Half a story

PCA, (rank constraints, condition number, linear dependence, etc) try to reconstruct a point P as a linear combination. 

\[ P = \text{averagePoint} + w_1 \ast \text{PC}_1 + w_2 \ast \text{PC}_2 \ldots \]

Support vector machines 
Non-parametric models

(cheap plug for my other class)

- How do you discover and represent these 1 parameter sets of points?

\[ \theta \quad s \]

\[ \text{radius,} \quad \]

Diagonal matrix of singular values:

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

The number of "large" singular is the number of "interesting" principle images that exist in the data set. The ratio of the largest and smallest singular value is called the condition number, and tells you "how close the matrix is to one of smaller rank".
How many degrees of freedom are there?

Let's use an image-based approach.

One way to arrange images is to embed them in a low-dimensional space. Images can be thought of as points in a high-dimensional space. How can we reduce the dimension?

Principal Component Analysis

- Can reduce the dimensionality of data, by projecting onto basis vectors.
- Finds the linear subspace which best preserves the variance of the original data.

Our pictures don't come from a linear subspace, so this isn't going to work.

Similarity Based Image Analysis

Instead of projection, let the ATOM of image analysis be global similarity measure between images.

Implicitly trading the ability to measure features within an image for the availability of a very large set of images.
Image distances to pose estimates

Suppose we had a distance measure between every pair of images.

**Tool:** Multi-dimensional scaling:

**Input:** all pairwise distances $D$.

**Output:** set of point positions $X$ whose pairwise distances match $D$.

\[
D = \begin{bmatrix}
3 & 5 & \cdot & \cdot \\
5 & 8 & 2 & 1 \\
3 & 8 & 0 & 6 & 4 \\
1 & 2 & 6 & 0 & 7 \\
3 & 1 & 4 & 7 & 0
\end{bmatrix}
\]

A positioning of cities with that set of pairwise distances.

Fact:

If you run MDS on points with original distances (from high dimensional space), it gives the SAME embedding as PCA, up to Euclidean transformation.

But we don't have distances between all pairs of images, (because we don't believe image similarities for anything but very similar images).

Two recent papers present methods of extending local similarities to give global constraints:

- Isomap (Tenenbaum, et al, 2000)
- LLE (Roweis and Saul, 2000)

These methods are techniques for non-linear dimension reduction.

Isomap:

**Define $G(V,E)$:**

- $V$ is the set of points (in our case, images)
- $E$ is the set of comparable points, (images with differences that are very small)
- $w(e)$ is the image difference.

**Algorithm:**

- Run all pairs shortest path algorithm on $G$.
- Define $D$, the pairwise distance matrix to be shortest path distance in $G$.

Run MDS, using $D$ as given pairwise distances.

MDS algorithm:

Squared distance matrix $S$: \( S(i,j) = D(i,j)^2 \).

Centering matrix $H$: \( H = I - 1/N \). (identity – uniform matrix of $1/N$)

Dot-product matrix: \( t(D) = -HSH/2 \). defined so: \( XX' = t(D) \) if for all $ij$ \( (X_i - X_j)'(X_i - X_j) = S(i,j) \).

\[
\begin{bmatrix}
3 \\
1 \\
3 \\
5 \\
8
\end{bmatrix}
\]

Each row is optimal embedding in $k$-dimensional space, if you use $k$ eigenpairs.
Isomap can discover the underlying structure in nonlinear data sets such as the Swiss Roll.

In this case, the dimensions already corresponded to the desired parameters, for the most part, but we can force specific locations for a small number of extreme points.

Complex Video Analysis

Complex Video Analysis

Temporal Super-Resolution

5 x Framerate

20 x Framerate (different input)
Complex Video Analysis

Video Trajectories: Automatic characterization of video

"In-Scene" Content Analysis

Better Distance Function Gives Better Results.

More analytic results - application to MRI images.

Cardiac MRI, courtesy of Nikos Tsekos, Department of Radiology, Washington University Medical School.

Automatic Registration of complex deformation.

Locally linear embedding (LLE)

- Local neighborhood relationship preserving.
- Uses more than distance between neighbors.

First solve for $W$ from $X$,

Then solve for $Y$ from $W$.

$X = \text{pixels} \times \text{images}$

$W = \text{images} \times \text{images}$

$Y = \text{D} \times \text{images}$

$W$ is very sparse. Seems to be a hard eigenvalue problem to solve.

$Ide_{a}$ find $Y$ that minimizes...

$\Phi(Y) = \sum_{i} |Y_{i} - \sum_{j} W_{ij}Y_{j}|^{2}$

Better Distance Function Gives Better Results.
Another Technique: LLE

Nonlinear Dimensionality Reduction by Locally Linear Embedding

Comparison to Isomap