Recognition 1

Introduction to Computer Vision

CSE 152

Lecture 17

Announcements

- HW 3 due on Thursday.
- HW 4 to be posted shortly

Two shape-from-X methods that use shading

- Photometric stereo: Single viewpoint, multiple images under different lighting.
  1. Arbitrary known BRDF
  2. Lambertian BRDF, known lighting
  3. Lambertian BRDF, unknown lighting.

An example of photometric stereo

Gradient Space (p,q)

Reflectance Map of Lambertian Surface

What does the intensity (Irradiance) of one pixel in one image tell us?

\( i = 0.5 \)

E.g., Normal lies on this curve

It constrains the surface normal projecting to that point to
Two Light Sources
Two reflectance maps

A third image would disambiguate match

Recovering the surface \( f(x,y) \)

Many methods: Simplest approach
1. From normal field \( n = (n_x, n_y, n_z) \), \( p = n_x/n_z \), \( q = n_y/n_z \)
2. Integrate \( p = df/dx \) along a row \((x,0)\) to get \( f(x,0) \)
3. Then integrate \( q = df/dy \) along each column starting with value of the first row

What might go wrong?

- Height \( z(x,y) \) is obtained by integration along a curve from \((x_0, y_0)\):
  \[ z(x,y) = z(x_0, y_0) + \int \left( p(x,y) \cdot dx + q(x,y) \cdot dy \right) \]
- If one integrates the derivative field along any closed curve, one expects to get back to the starting value.
- Might not happen because of noisy estimates of \((p,q)\)

Lambertian Surface

At image location \((u,v)\), the intensity of a pixel \( x(u,v) \) is:
\[
e(x(u,v)) = \langle a(u,v) \cdot (n(u,v) \cdot s) \cdot \hat{s} \rangle
\]
where
- \( a(u,v) \) is the albedo of the surface projecting to \((u,v)\).
- \( n(u,v) \) is the direction of the surface normal.
- \( s \) is the light source intensity.
- \( \hat{s} \) is the direction to the light source.
Lambertian Photometric stereo
- If the light sources \( s_1, s_2, \) and \( s_3 \) are known, then we can recover \( b \) from as few as three images. (Photometric Stereo: Silver 80, Woodham 81).
- \([e_1, e_2, e_3] = b[b, s_1, s_2, s_3]\)
- i.e., we measure \( e_1, e_2, \) and \( e_3 \) and we know \( s_1, s_2, \) and \( s_3 \). We can then solve for \( b \) by solving a linear system.
- \( b^T = [e_1 \ e_2 \ e_3 [s_1 \ s_2 \ s_3]^{-1}\)
- Normal is: \( n = b/b \), albedo is: \( b \)
- Let \( E_1, E_2, E_3 \) be vectorized images, we can solve for all normals, scaled by the albedos as:
  \[
  B = \begin{bmatrix} E_1 \ E_2 \ E_3 \end{bmatrix} [s_1 \ s_2 \ s_3]^{-1}
  \]

III. Photometric Stereo with unknown lighting and Lambertian surfaces

How do you construct subspace?
- \([E_1 \ E_2 \ E_3] = B^0[s_1 \ s_2 \ s_3]\)
- Given three or more images \( E_1 ... E_m \), estimate \( B \) and \( s \).
- How? Given images in form of \( E = [E_1 \ E_2 ...] \).
- Compute SVD(E) and let \( B^* \) be the \( n \) by \( 3 \) matrix formed by first \( 3 \) singular values.

Matrix Decompositions
- Definition: The factorization of a matrix \( M \) into two or more matrices \( M_1, M_2, ..., M_k \) such that \( M = M_1 M_2 ... M_k \).
- Many decompositions exist...
  - QR Decomposition
  - LU Decomposition
  - LDU Decomposition
  - Etc.

Singular Value Decomposition
Excellent ref: “Matrix Computations,” Golub, Van Loan
- Any \( m \) by \( n \) matrix \( A \) may be factored such that
  \[
  A = U \Sigma V^T
  \]
  \([m \times n] = [m \times m][m \times n][n \times n]\)
- \( U: m \) by \( m \), orthogonal matrix
  - Columns of \( U \) are the eigenvectors of \( AA^T \)
- \( V: n \) by \( n \), orthogonal matrix,
  - Columns are the eigenvectors of \( A^T A \)
- \( \Sigma: m \) by \( n \), diagonal with non-negative entries \( \sigma_1, \sigma_2, ..., \sigma_k \) with \( k = \min(m,n) \) are called the called the singular values
  - Singular values are the square roots of eigenvalues of both \( AA^T \) and \( A^T A \)
- Result of SVD algorithm: \( \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_k \)

Applying SVD to Photometric stereo
- The images are formed by
  \[
  [e_1 \ e_2 \ e_3 ... e_n] = B^0[s_1 \ s_2 \ s_3 ... s_n]
  \]
- So, \( \text{svd}(E) = U \Sigma V^T \) where \( U \) is \( N \) by \( n \), \( \Sigma \) is \( n \) by \( n \), and \( V^T \) is \( n \) by \( N \)
- Without noise, we expect 3 non-zero singular values, and so \( U \Sigma V^T \rightarrow U^T \Sigma V^T \)
  - where \( U^T \) is \( N \) by \( 3 \), \( \Sigma^* \) is \( 3 \) by \( 3 \), and \( V^T \) is \( 3 \) by \( n \).
- In particular \( B = U^T A \) where \( A \) is some \( 3 \times 3 \) matrix.
Do Ambiguities Exist? Yes

- Is \( B \) unique?
- For any invertible matrix \( A, B^* = BA \) also a solution
- For any image of \( B \) produced with light source \( S \), the same image can be produced by lighting \( B^* = BA \) with \( S^* = A^{-1}S \) because \( X = B^*S^* = B AA^{-1}S = BS \)
- When we estimate \( B \) using SVD, the rows are NOT generally the normal times the albedo.

GBR Transformation

Only Generalized Bas-Relief transformations satisfy the integrability constraint:

\[
A = G^* = \begin{bmatrix}
\lambda & 0 & -\mu \\
0 & \lambda & -\nu \\
0 & 0 & 1
\end{bmatrix}
\]

Uncalibrated photometric stereo

1. Take \( n \) images as input without knowledge of light directions or strengths
2. Perform SVD to compute \( B^* \).
3. Find some \( A \) such that \( B^*A \) is close to integrable.
4. Integrate resulting gradient field to obtain height function \( f(x,y) \).

Comments:
- \( f(x,y) \) differs from \( f(x,y) \) by a GBR.
- Can use specularities to resolve GBR for non-Lambertian surface.

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.
Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

How many visual object categories are there?

\[ \sim 10,000 \text{ to } 30,000 \]

Specific recognition tasks
Scene categorization

- outdoor/indoor
- city/forest/factory/etc.

Image annotation/tagging

- street
- people
- building
- mountain
- ...

Object detection

- find pedestrians

Image parsing

- sky
- mountain
- building
- tree
- banner
- street lamp
- market
- people

Object Recognition: The Problem

Given: A database D of “known” objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object

Segmentation

(where is it 2D)

Recognition

(where is it 3D)

Pose Est.

(what is it)

WHAT AND WHERE!!!

Within-class variations
Recognition Challenges

• Within-class variability
  – Different objects within the class have different shapes or different material characteristics
  – Deformable
  – Articulated
  – Compositional
• Pose variability:
  – 2-D Image transformation (translation, rotation, scale)
  – 3-D Pose Variability (perspective, orthographic projection)
• Lighting
  – Direction (multiple sources & type)
  – Color
  – Shadows
• Occlusion – partial
• Clutter in background -> false positives

• Categories near top of tree (e.g., vehicles) – lots of within class variability
• Fine grain categories (e.g., species of birds) -- Moderate within class variation
• Instance recognition (e.g., person identification) – within class mostly shape articulation, bending, etc.

Sketch of a Pattern Recognition Architecture

Sliding window approaches

Example: Face Detection

• Scan window over image.
• Search over position & scale.
• Classify window as either:
  – Face
  – Non-face

• So, what are the features?
• So, what is the classifier
The Space of Images

- We will treat a d-pixel image as a point in an d-dimensional space, \( \mathbf{x} \in \mathbb{R}^d \).
- Each pixel value is a coordinate of \( \mathbf{x} \).

More features

- Filtered image
- Filter with multiple filters (bank of filters)
- Histogram of colors
- Histogram of Gradients (HOG)
- Haar wavelets
- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Feature (SURF)

Face Space:

- A set of face images construct a face space in \( \mathbb{R}^d \)
- Appearance-based methods analyze the distributions of individual faces in face space

Some questions:
1. How are images of an individual, under all conditions, distributed in this space?
2. How are the images of all individuals distributed in this space?

Nearest Neighbor Classifier

\[ \{ \mathbf{R}_j \} \text{ are set of training images.} \]

\[ \mathbf{I} = \arg\min_j \text{dist}(\mathbf{R}_j, \mathbf{I}) \]

Comments on Nearest Neighbor

- Sometimes called “Template Matching”
- Variations on distance function (e.g. \( L_1 \), robust distances)
- Multiple templates per class—perhaps many training images per class.
- Expensive to compute \( k \) distances, especially when each image is big (d-dimensional).
- May not generalize well to unseen examples of class.
- No worse than twice the error rate of the optimal classifier — if enough training samples
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction
Do features vectors have structure in the image space?

- Faces of individuals cluster in the image space. (Not true).
- Faces of individuals are confined to a linear or affine subspace of $\mathbb{R}^d$.
- Faces of an individual are approximated by a linear subspace.
- Faces and objects lie on or near a manifold in the space of images.

An idea:
Represent the set of images as a linear subspace.

What is a linear subspace?

Let $F$ be a vector space and let $W$ be a subset of $F$. Then $W$ is a subspace if:
1. The zero vector, $0$, is in $W$.
2. If $u$ and $v$ are elements of $W$, then any linear combination of $u$ and $v$ is an element of $W$: $au + bv \in W$.
3. If $u$ is an element of $W$ and $c$ is a scalar, then the scalar product $cu \in W$.

A $k$-dimensional subspace is spanned by $k$ linearly independent vectors.
It is spanned by a $k$-dimensional orthogonal basis.

Linear Subspaces & Linear Projection

- An $d$-pixel image $x \in \mathbb{R}^d$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^k$ by:
  \[ y = Wx \]
where $W$ is a $k \times d$ matrix.

- Recognition is performed in $\mathbb{R}^1$ using, for example, nearest neighbor.

- How do we choose a good $W$?

Linear Subspaces & Recognition

1. Approximate all training images as a single linear subspace (Eigenfaces).
2. Represent lighting variation w/o shadowing for a single individual as a 3-D linear subspace. A collection of $n$ individuals is modeled as $n$ 3-D linear subspaces.
3. Represent lighting variation w/ shadowing for a single individual as a 9-D linear subspace. A collection of $n$ individuals is modeled as $n$ 9-D linear subspaces.
4. Project all training images to a single subspace that enhances discriminability (Fisherfaces).

Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of $n$ feature vectors $x_i$ ($i = 1, \ldots, n$) in $\mathbb{R}^d$. Write:

\[
\mu = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

\[
\Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T
\]

The unit eigenvectors of $\Sigma$ — which we write as $v_1, v_2, \ldots, v_k$, where the order is given by the size of the eigenvalues and $k$ is the largest eigenvalue — give a set of features with the following properties:
- They are independent.
- Projection onto the basis $(v_1, \ldots, v_k)$ gives the $k$-dimensional set of linear features that preserve the most variance.

Algorithm 22.3: Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.

PCA Example

First Principal Component
Direction of Maximum Variance

Mean
**Eigenfaces**

- **Modeling**
  1. Given a collection of $n$ training images $x_i$, represent each one as a $d$-dimensional column vector.
  2. Compute the mean image and covariance matrix.
  3. Compute $k$ Eigenvectors of covariance matrix corresponding to $k$ largest Eigenvalues and form matrix $W = [v_1, v_2, ..., v_k]$ (Or perform using SVD!!) (note that these are images).
  4. Project the training images to the $k$-dimensional Eigenspace: $y_i = Wx_i$

- **Recognition**
  1. Given a test image $x$, project vectorized image to Eigenspace by $y = Wx$
  2. Perform classification to the projected training images.