Course Notes

EECS 442

Computer Vision

Matthew Johnson-Roberson - Fall 2015

Contributors: Max Smith

Latest revision: September 10, 2015

Contents

Abstract

Computational methods for the recovery, representation and application of visual information. Topics from image formation, binary images, digital geometry, similarity and dissimilarity detection, matching, curve and surface fitting, constraint propagation relaxation labeling, stereo, shading texture, object representation and recognition, dynamic scene analysis and knowledge based techniques. Hardware, software techniques.

1 Introduction

- Computer vision studies the tools and theories that enable the design of machines that can extract useful information from imagery data (images and videos) toward the goal of interpreting the world.
  - **Information**: visual cues, 3D structure, motion flows, etc.
  - **Interpretation**: recognize objects, scenes, actions, events

1.1 Course overview

Geometry

- How to extract 3D information
- Which cues are useful
- What are the mathematical tools
- **Visual cues**: texture, shading, contours, shadows, reflections
- **Number of observers**: monocular, multiple views
- **Active lighting**: laser stripes, structured lighting patterns

Low & Mid-level vision

- Extract useful building blocks
- Region segmentation
- Motion flows

High level vision

- Recognition of objects and people
- Places
- Actions & events

2 Linear Algebra & Geometry

2.1 Basic definitions and properties

Vectors

- **Vectors (2D or 3D)**:
  - $v = (x_1, x_2)$
  - Magnitude: $||v|| = \sqrt{x_1^2 + x_2^2}$
  - **Unit vector**: iff $||v|| = \frac{v}{||v||} = 1$
  - Orientation: $\theta = \tan^{-1}\left(\frac{x_2}{x_1}\right)$
- Vector addition: $\vec{v} + \vec{w} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y + 2)$
- Vector subtraction: $\vec{v} + \vec{w} = (x_1, x_2) - (y_1, y_2) = (x_1 - y_1, x_2 - y + 2)$
- Scalar product: $a\vec{v} = a(x_1, x_2) = (ax_1, ax_2)$
- Inner (dot) product: $\vec{v} + \vec{w} = (x_1, x_2) \cdot (y_1, y_2) = x_1y_1 + x_2y_2 = ||\vec{v}|| \cdot ||\vec{w}|| \cos(\alpha)$
  - Where α is the angle between the vectors
- Orthonormal basis:
  - $\vec{i} = (1, 0), \vec{j} = (0, 1)$
  - You can use them to represent vectors: $\vec{v} = (x_1, x_2) = x_1\vec{i} + x_2\vec{j}$
- Vector (cross) product:
  - $\vec{u} = \vec{v} \times \vec{w}$
  - Magnitude: $||\vec{u}|| = ||\vec{v} \times \vec{w}|| = ||\vec{v}||||\vec{w}|| \sin(\alpha)$
  - $(x_1, x_2, x_3) \times (y_1, y_2, y_3) = (x_2y_3 - x_3y_2, x_3y_1 - x_1y_3, x_1y_2 - x_2y_1)$

Matrices

- Matrices:
  - e.g.,

$$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

  - Can be used to represent an image, where each entry is a pixel.
- Sum: $C_{n\times m} = A_{n\times m} + B_{n\times m}; c_{ij} = a_{ij} + b_{ij}$
- Product: $C_{n\times p} = A_{n\times m}B_{m\times p}$

$$\begin{bmatrix} 3 & 0 \\ 0 & 7 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 7 & 0 \end{bmatrix}$$

- Transpose: $C_{m\times n} = A^T_{n\times m}; c_{ij} = a_{ji}$
- Determinant:
  - Must be square
  - 2D: $det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{21}a_{12}$
  - 3D: $det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} = \end{bmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{23} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$
- Inverse:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

  - Must be square
  - $A^{-1}A = I$
- Orthogonal: $Q_{n\times n}Q^T_{n\times n} = Q^T_{n\times n}Q_{n\times n} = I$
  - Rows and columns are unit vectors
- Block forms:
  - Represent submatrices as variables
  - e.g.,

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S & t \\ 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, t = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Eigenvalues and Eigenvectors

- **Eigen relation**: Matrix A acts on vector $\vec{u}$ and produces a scaled version of $\vec{u}$

$$A\vec{u} = \lambda\vec{u}$$

- $\vec{u}$: eigenvector
- λ: eigenvalue
- The eigenvalues of A are the roots of the characteristics equation:

$$p(\lambda) = det(\lambda I - A) = 0$$

- **Eigendecomposition**:

$$A = S\Lambda S^{-1}$$

  - Λ: diagonal matrix of eigenvalues
  - S: matrix with eigenvetors for columns

Singular Value decomposition

- $A = U\Sigma V^{-1}$
- U, V: orthogonal matrix
- Σ: diagonal matrix of singular values
- **Singular value**: $sigma_i = \sqrt{\lambda_i}$
  - λ: eigenvalue of $A^T A$

2.2 Geometrical transformations

- **Translation**:

$$P \rightarrow P' P = (x, y), t = (t_x, t_y) P' = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

  - This will shift P by t
- **Homogeneous Coordinates**
  - Multipl;y the coordinate by a non-zero scalar and add an extra coordinate equal to that scalar. e.g.,

$$(x, y) \rightarrow (xz, yz, z), z \neq 0$$

  - To go back to cartesian: divide by last coordinate and eliminate it:

$$(x, y, z), z \neq 0 \rightarrow (x/z, y/z)$$

- Translation using homogeneous coordinates:

$$P' = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} i & T \\ 0 & 1 \end{bmatrix} P = TP$$

- **Scaling**:

$$P' = \begin{bmatrix} s_x t_x \\ x_y t_y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} P = SP$$

- Scaling & Translating:

$$P' = TSP = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Translation and scaling is not communative: $STP \neq TSP$
- **Rotation**:
  - Rotation by angle θ

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$P' = RP$$

- Translation + Rotation + Scaling =

$$P' = (TRS)PP' = \begin{bmatrix} RS & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2.3 Transformation in 2D

Isometries

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_e \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve distance (areas)
- 3 DOF
- Regulate motion of a rigid object

Similarities

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} SR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserve:
  - ratio of lengths
  - angles
- 4 DOF

Affinities

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_a \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = R(\theta)R(-\phi)DR(\phi), D = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

- Preserve:
  - Parallel lines
  - Ratio of areas
  - Ratio of lengths on collinear lines
  - others...
- 6 DOF

Projective

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H_p \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 8 DOF
- Preserve:
  - cross ratio of 4 collinear points
  - collinearity
  - and a few others...