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: $C_{n\times m} = A_{n\times m} + B_{n\times m}; c_{ij} = a_{ij} + b_{ij}$

2.2 Geometrical transformations

-

2.3 Application: removing perspective distortion - the DLT algorithm

–