Eigen local color histograms for object recognition and orientation estimation

Electronic Imaging
2007
Muselet D., Funt B., and Macaire L.

Presented by Dong Chang Lee

School of Electrical Engineering and Computer Science
Kyungpook National Univ.
Proposed method
  - New algorithm for color-based recognition of object in cluttered scenes
  - Determining the 2D pose of each object
Introduction

◆ An entry in the database of prototypes
  – Building from an image of a single object placed on an uniform background

◆ Test image
  – Containing the objects to be recognized

◆ Color histogram
  – Efficient method for object recognition and image indexing
  – Advantages
    • Simple and fast to compute
    • Invariant to rotation and translation
    • Insensitive to partial object occlusion
Related works

- Local-feature-based object recognition methods
  - Extracting interest points from the images
  - Evaluating local descriptors around these points
- Drawback
  - Reliability problems associated with interest point detectors

Proposed method in this paper

- Combining the ideas of color histogram matching and local feature matching
- Analyzing all the local neighborhoods in the image and to extract descriptors for all of them
- Requiring some specific properties
  - Computing a descriptor must as fast as possible
  - A descriptor must require as little storage space as possible
  - The comparison of two descriptors must be as fast as possible
  - Descriptors need to be robust to rotation and translation of the object and partial occlusion between objects

Fig. 1. These images come from the SOIL-47 database from the university of surrey.
Each prototype image and subwindow
- \( p \) prototype images \( I_{pro}^i, \ i \in \{1, \ldots, P\} \)
  - Representing the object \( O^i \)
- Dividing into \( WP^i \) subwindows \( wp_{j}^i, \ j \in \{1, \ldots, WP^i\} \)
  - Each one representing the \( j^{th} \) feature \( Op_{j}^i \) of the object \( O^i \)

Proposed object recognition method
- Requiring the storage and matching of many subwindow histograms
- Reducing the memory and computation requirements
- Applying principal component analysis to the set of prototype local color histogram
  - Eigen local color histograms
  - Using Manhattan distance for matching
\[ D_i = H1_i - H2_i \]  

where \( H_{ni} \) the \( i^{th} \) bin of the histogram \( H_n \)

- After the matching step
  - Using subwindow’s labels \([input\ subwindow,\ object\ area] [wq_k, Op_j^i]\)
  - Determining the best geometrical transformation from the corresponding prototype image to the input image
2D pose estimation

◆ Finding the position and orientation
  – Using the 2D affine transformation
  – Mapping the locations of features in the prototype image to locations of features found to match in the test image

\[
\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix} m_1 & m_3 \\ m_2 & m_4 \end{bmatrix} \begin{bmatrix} x \\
y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}
\]

(2)

where \( t_x, t_y \) the translation parameters
\( m_i \) the rotation parameters

\[
\begin{cases}
m_3 = -m_2 \\
m_4 = m_1
\end{cases}
\]

(3)
\[
\begin{bmatrix}
  x & -y & 1 & 0 \\
  y & x & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  m_1 \\
  t_x \\
  t_y \\
\end{bmatrix}
= 
\begin{bmatrix}
  u \\
  v \\
\end{bmatrix}
\] 
\quad (4)

\[
\begin{bmatrix}
  x & -y & 1 & 0 \\
  y & x & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  m_1 \\
  t_x \\
  t_y \\
\end{bmatrix}
= 
\begin{bmatrix}
  u \\
  v \\
\end{bmatrix}
\] 
\quad (5)

\[
[PROTP] \times [TRANS] = [INPUT] \] 
\quad (6)
- Calculating the least-squares solution for geometric parameters
  - Incorrect labeling → poor least-squares fit
- Proposed method to solve this problem
  - Mismatched features of large residual error → deleting
  - Least-squares approximation for the geometric transformation
A step by step example

- Example

Fig. 2. Input image.

Fig. 3. The input image is decomposed into overlapping subwindows. The figure shows the complete set of subwindows.
Fig. 4. Each subwindow is labeled according to the best matching subwindow from the prototype database. In the figure, the labeling is represented pictorially by the contents of the best matching subwindow.

Fig. 5. The 3 objects matched. A least-squares fit of the parameters of a geometrical transformation determines the mapping from the prototype image to the input test image.
\[ \text{Rotation(object1)} = \begin{bmatrix} 0.9885 & 0.0193 \\ -0.0193 & 0.9885 \end{bmatrix}, \quad \text{Translation(object1)} = \begin{bmatrix} -119.3352 \\ -2.8682 \end{bmatrix} \] (7)

\[ \text{Rotation(object2)} = \begin{bmatrix} 0.0561 & -0.9796 \\ 0.9796 & 0.0561 \end{bmatrix}, \quad \text{Translation(object2)} = \begin{bmatrix} 352.2587 \\ 5.7036 \end{bmatrix} \] (8)

\[ \text{Rotation(object3)} = \begin{bmatrix} 0.9449 & 0.0032 \\ -0.0032 & 0.9449 \end{bmatrix}, \quad \text{Translation(object3)} = \begin{bmatrix} 101.0458 \\ 33.5976 \end{bmatrix} \] (9)

Fig. 6. The method’s results are show pictorially. The three objects shown are the ones identified from the database, and their locations and orientations are those provided by pose estimation step.
Results

- Proposed method
  - Using the SOIL-47 public database for testing
  - Subwindow → 50x50 pixels
  - Distance between two neighboring subwindows’ origin
    - 15 pixels
  - Average number of subwindows within each prototype image
    - 140
  - Raw histogram → $8^3 = 512$ bins
  - After projection on the eigenbasis 64 bins