On-line Face Recognition Using SIFT Features

Zhu Tianhua

09300240004

December 2011

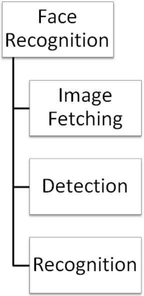

Figure: Framework for the face recognition project

- Face detection using *OpenCV*'s *Haar Feature-based Cascade Classifier* has been quite common. [5, 3].
- Nothing to do with face rotation
- Cope with false faces

CAMShift

- Continuously Adaptive Mean Shift [1]
- Use colour statistic histogram as a pattern
- Search in the adjacent area around last face position to match for a face.
- Fast, moderately accurate.

It inspired us to find only faces in their last positions for fast matching. We can therefore make use of information provided by nearby frames.

SIFT

Scale Invariant Features Transformation (SIFT) was first proposed by Lowe in 1999 and accomplished in 2004 [4]. For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects.

Procedures for SIFT

1. Scale-space extrema detection
2. Key-point localization
3. Orientation assignment
4. Key-point descriptor

DoG Operation

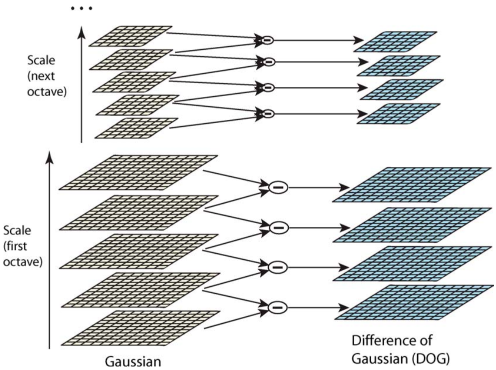

Figure: DoG operation

Finding Local Extrema

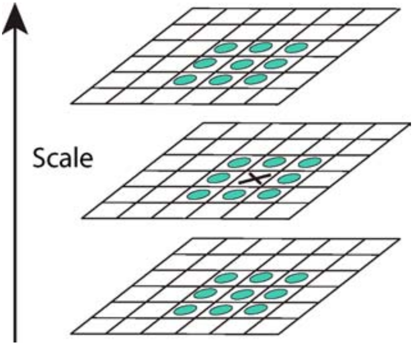

Figure: Scale-space extrema detection

Ideas for Recognition Using SIFT

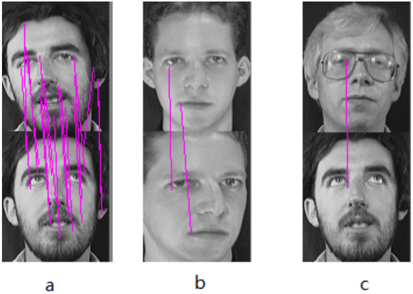

Figure: Tests on examples from the ORL dataset

Implementation

In this project, I use the implementation of SIFT algorithm by Rob Hess [2]. His open-source SIFT library is implemented in C using OpenCV and includes functions for computing SIFT features in images, and matching SIFT features between images using kd-trees.

Single Person

Given an evaluation function $f(a, b)$ and a corresponding threshold θ, where a, b are two series of Scale-invariant features (SIFs), we can perform face recognition using SIFT.

1. Detect faces in an image.
2. For each face detected, perform SIFT in its rectangle area, resulted in a series of SIFs, notated by a.
3. If $\|a\| < N_0$, ignore it.
4. For each series of SIFs s stored in list L, we evaluate their similarity using f. If $min_{s \in L} f(a, s) > \theta$, we regard the input face image as a new face, and thus add a to L, and give it a new id. Otherwise, we just return the id of s.

Two Types of Errors and Assumptions

- Recognizing a person as another
- Regard a preliminarily acquainted face as unknown.

We may guess that the more input images we used for training, the higher the accuracy will be.
People moves slowly in adjacent video frames.

Reducing False Faces

- Set up a threshold N_0 and ignore those areas containing too few details.
- Overlapping areas will also be merged or ignored, so that increase detection accuracy.
- We need enough resolution to identify people, so we may suppose that there won't be many faces in an image.

Recognition Procedure

After such pre-process in detection phase, we just start up recognition phase and check similarities between the SIFs a of input and those of stored acquaintances, one by one. We may simply use the following equation to find the result:

$$\arg\min_{x \in L} f(a, x) \tag{1}$$

Such judgement will result in that a person occurs in more than one place in the image, which is impossible in reality (see figure 5). And also, it may cause chaos when multiple samples trained and recorded for one person.

Figure: The two people are both labelled as yellow and same id

Another Decision Strategy

- Assign each face a person, rather than processing them independently.
- We can build a matrix $A = (f(a_i, l_j))$, where a_i is the i-th face and l_j is the j-th SIFs in database L.
- For those scores lower than threshold *theta*, we assign the id of l_j to face a_i if $l_j = \arg\min A_i$, otherwise we recognize it as unknown and leave it for acquainting phase.
- In acquainting phase, we first find out the nearest face in the last frame, and compare it with that of current frame. If the score is lower than a looser bound $\hat{\theta} > \theta$, we regard the two faces as the same person, and then add the new sample SIFs to L. Otherwise, we will allocate a new id for it.

Hierarchy

1. Object (Person)
   – Recognition result
2. SIF Series (multiple)
   – Similarity determinant
3. SIF
   – Key points for matching

Determining function f

- Our function f simply returns $1 - m/n$, where n is the number of SIFs in the current series.
- An alternative is to use $\sum_{b_i} d_0/d_1$, where d_0 and d_1 are distances between the given b_i SIF in series b and its two NNs.
- As a consideration of using probability distribution p, we may score it as $\hat{f} = f(\cdot) \cdot p(d)$ for valid distance d. We may use Gaussian distribution.

Dataset

- The ORL Database of Faces

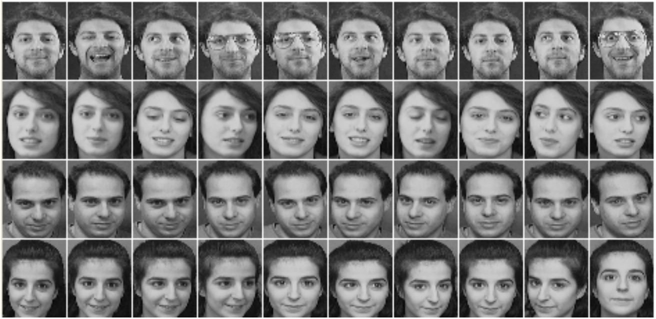

Figure: Four examples from the ORL dataset

Dataset

- Two short video clips recorded in class. One is a teacher's monologue, another is a side-frontal view of students.

Figure: Snapshots for the two video clips

Dataset

The two video clips are all sampled at 29 fps, with height of 480 px and width of 720 px, lengthened 1' 42" and 20", respectively. They are converted from an Advanced Streaming Format (ASF) video of lecture to AVI format, and contains some codec errata, typically are scan lines and noises.

Results for Static Images

A test running on the whole ORL set indicated that pure SIFT method can reach 87.8% accuracy (49 errors in all 400 tests) when trained with 2 pictures per person and the least matches threshold N_0 set to zero.
We can find out that SIFT is more competitive when training samples are highly limited, and we are inconvenient to re-train the model on-line for Eigenface.

Results for Videos

- Works well on single person samples.
- The program is yet not so robust or stable, as illumination conditions differs and colour space is sensitive to it.
- Process at 1.66 fps, and may get even slower as time grows Set up some thresholds for the program, so that it may keep only features occurring frequently and/or recently and eliminate those seldom come around or out of sight for long.

Conclusion

The project has implemented an automatic face recognition program, which can learn on-line new comers and identify different people. It performs well for single person sceneries, and can be used under multi-personal circumstances. Its speed is slow, and far from real-time recognition, but its simplified version can get higher speed, yet lower correctness.

Yizong Cheng.
Mean shift, mode seeking, and clustering.
IEEE Trans. Pattern Anal. Mach. Intell., 17:790–799, August 1995.

Rob Hess.
An open-source siftlibrary.
In *Proceedings of the international conference on Multimedia*, MM '10, pages 1493–1496, New York, NY, USA, 2010. ACM.

Rainer Lienhart and Jochen Maydt.
An extended set of haar-like features for rapid object detection.
In *IEEE ICIP*, pages 900–903, 2002.

D Lowe.
Distinct image features from scale-invariant keypoints.

Thank You!

Zhu Tianhua
09300240004