State of The Art

Vincent FALCONIERI

February 2019 - August 2019

Contents

0.1 Introduction

A general overview was made through standard web lookup. [Bupe,] A look was given to libraries, which also provide detailed and useful information. [Fea,]

Chapter 1

Standard algorithms

Goal is to transform visual information into vector space

1.1 SIFT- Scale Invariant Feature Transform

[Lowe,]

Pro

- test

Con

- Patented algorithm, not included in OpenCV (only non-free module)
- Slow (HOW MUCH TO CHECK)

Steps of the algorithm

1. Extrema detection
   Uses an approximation of LoG (Laplacian of Gaussian), as a Difference of Gaussion, made from difference of Gaussian blurring of an image at different level of a Gaussian Pyramid of the image. Kept keypoints are local extrema in the 2D plan, as well as in the blurring-pyramid plan.
2. Keypoint localization and filtering
   Two threesholds has to be set :
   - Contract Threeshold : Eliminate low contract keypoint (0.03 in original paper)
   - Edge Threeshold : Eliminate point with a curvature above the threeshold, that could match edge only. (10 in original paper)
3. Orientation assignement
   Use an orientation histogram with 36 bins covering 360 degrees, filled with gradient magnitude of given directions. Size of the windows on which the gradient is calculated is linked to the scale at which it's calculated. The average direction of the peaks above 80% of the highest peak is considered to calculate the orientation.
4. Keypoint descriptors
   A 16x16 neighbourhood around the keypoint is divided into 16 sub-blocks of 4x4 size, each has a 8 bin orientation histogram. 128 bin are available for each Keypoint, represented in a vector of float, so 512 bytes per keypoint. Some tricks are applied versus illumination changes, rotation.
5. Keypoint Matching
   Two distance calculation :
   - Finding nearest neighboor.
   - Ratio of closest distance to second closest is taken as a second indicator when second closest match is very near to the first. Has to be above 0.8 (original paper) (TO CHECK what IT MEANS)

1.2 SURF – Speeded-Up Robust Features

[Bay et al.,]

Pro

- Faster than SIFT (x3) : Parralelization, integral image ..
- Tradeoffs can be made :
  - Faster : no more rotation invariant, lower precision (dimension of vectors)
  - More precision : **extended** precision (dimension of vectors)
- Good for blurring, rotation

Con

- Patented algorithm
- Not good for illumination change, viewpoint change

Steps of the algorithm

1. Extrema detection
   Approximates Laplacian of Guassian with a Box Filter. Computation can be made in parrallel at different scales at the same time, can use integral images . . .
   Roughly, does not use a gaussian approximation, but a true "square box" for edge detection, for example.
   The sign of the Laplacian (Trace of the Hessian) give the "direction" of the contrast : black to white or white to black. So a negative picture can match with the original ? (TO CHECK)
2. Keypoint localization and filtering
3. Orientation assignement
   Dominant orientation is computed with wavlet responses with a sliding window of 60°
4. Keypoint descriptors
   Neighbourhood of size 20sX20s is taken around the keypoint, divided in 4x4 subregions. Wavelet response of each subregion is computed and stored in a 64 dimensions vector (float, so 256 bytes), in total. This dimension can be lowered (less precise, less time) or extended (e.g. 128 bits ; more precise, more time)
5. Keypoint Matching

1.3 U-SURF – Upright-SURF

Rotation invariance can be “desactivated” for faster results, by bypassing the main orientation finding, and is robust up to 15°rotation.

1.4 BRIEF – Binary Robust Independent Elementary Features

Extract binary strings equivalent to a descriptor without having to create a descriptor

See BRIEF [BRI,]

Pro

- Solve memory problem

Con

- Only a keypoint descriptor method, not a keypoint finder
- Bad for large in-plan rotation

Steps of the algorithm

1. Extrema detection
2. Keypoint localization and filtering
3. Orientation assignement
4. Keypoint descriptors
   Compare pairs of points of an image, to directly create a bitstring of size 128, 256 ou 512 bits. (16 to 64 bytes)
   Each bit-feature (bitstring) has a large variance ad a mean near 0.5 (TO VERIFY). The more variance it has, more distinctive it is, the better it is.
5. Keypoint Matching Hamming distance can be used on bitstrings.

1.5 R-BRIEF – Rotation (?) BRIEF

Variance and mean of a bit-feature (bitstring) is lost if the direction of keypoint is aligned (TO VERIFY : would this mean that there is a preferential direction in the pair of point selection ?)

Uncorrelated tests (TO CHECK WHAT IT IS) are selected to ensure a high variance.

1.6 CenSurE

Pro

Con

1.7 ORB – Oriented FAST and Rotated BRIEF

From [Rublee et al.,] which is rougly a fusion of FAST and BRIEF. See also [ORB,]

Pro

- Not patented

Con

Steps of the algorithm

1. Extrema detection
   FAST algorithm (no orientation)
2. Keypoint localization and filtering
   Harris Corner measure : find top N keypoints
   Pyramid to produce multi scale features
3. Orientation assignement
   The direction is extracted from the orientation of the (center of the patch) to the (intensity-weighted centroid fo the patch). The region/patch is circular to improve orientation invariance.
4. Keypoint descriptors
   R-BRIEF is used, as Brief Algorithm is bad at rotation, on rotated patches of pixel, by rotating it accordingly with the previous orientation assignement.
5. Keypoint Matching
   Multi-probe LSH (improved version of LSH)

1.8 KASE -

Shipped in OpenCV library. Example can be found at [Nikishaev,]

Pro

Con

Steps of the algorithm

1.9 FAST – Features from Accelerated Segment Test

Pro

Con

Steps of the algorithm

1. Extrema detection
2. Keypoint localization and filtering
3. Orientation assignement
4. Keypoint descriptors
5. Keypoint Matching

Chapter 2

Classic Computer vision techniques - White box algorithms

2.1 Step 1 - Key Point Detection

- Corner detectors to find easily localizable points.

Harris

[Harris and Stephens,]

Distinctive features :

- Rotation-invariant
- NOT scaling invariant

FAST

Distinctive features :

- Not rotation-invariant (no orientation calculation)
- ? scaling invariant

2.2 Step 2 - Descriptor Extraction

Extract a small patch around the keypoints, preserving the most relevant information and discaring necessary information (illumination ..)

Can be :

- Pixels values
- Based on histogram of gradient
- Learnt

Usually :

- Normalized
- Indexed in a searchable data structure

Example :

Vector descriptors based on our keypoints, each descriptor has size 64 and we have 32 such, so our feature vector is 2048 dimension.

2.3 Step 3 - Matching

Linked to correspondence problem ?

2.3.1 Distance

Hamming distance

Bruteforce

- $O(N^2)$, N being the number of descriptor per image
- One descriptor of the first picture is compared to all descriptor of a second candidate picture. A distance is needed. The closest is the match.
- Ratio test
- CrossCheck test : list of "perfect match" (TO CHECK)

Best match

- Returns only the best match
- Returns the K (parameter) best matchs

FLANN – Fast Library for Approximate Nearest Neighboors

- Collections of algorithm, optimized for large dataset/high dimension
- Returns the K (parameter) best matchs
- [Ope,]

2.3.2 Compression of descriptors before matching

LSH – Locally Sensitive Hashing

- O(˜N)
- Returns the K (parameter) best matchs
- [Ope,]
- Convert descriptor (floats) to binary strings. Binary strings matched with Hamming Distance, equivalent to a XOR and bit count (very fast with SSE instructions on CPU)

BBF – Best bin first Kd-tree

- O(˜N)
- Example : SIFT – Scale Invariant Feature Tranform

2.4 Step 4 - Model Fitting

- Identify inliers and outliers ˜ Fitting a homography matrix ˜ Find the transformation of (picture one) to (picture two)
- Inliers : "good" points matching that can help to find the transformation
- outliers : "bad" points matching

- See [Fea,]

RANSAC – Random Sample Consensus

Estimation of the homography

Least Meadian

Chapter 3

Neural networks – Black box algorithms

3.1 ConvNet - Convolutional Neural Networks

Learn a metric between any given two images. The distance can be threesholded to decide if images match or not.

Training phase

Goal :

- Minimizing distance between "same image" examples
- Maximizing distance between "not same image" examples

Evaluation phase

Apply an automatic threeshold.

SVM - Support Vector Machine

Bibliography

[BRI,] BRIEF (Binary Robust Independent Elementary Features) — OpenCV 3.0.0-dev documentation.

[Fea,] Feature Matching + Homography to find Objects — OpenCV 3.0.0-dev documentation.

[Ope,] OpenCV: Feature Matching.

[ORB,] ORB (Oriented FAST and Rotated BRIEF) — OpenCV 3.0.0-dev documentation.

[Bay et al.,] Bay, H., Tuytelaars, T., and Van Gool, L. SURF: Speeded Up Robust Features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, volume 3951, pages 404–417. Springer Berlin Heidelberg.

[Bupe,] Bupe, C. What algorithms can detect if two images/objects are similar or not? - Quora.

[Harris and Stephens,] Harris, C. and Stephens, M. A Combined Corner and Edge Detector. In *Procedings of the Alvey Vision Conference 1988*, pages 23.1–23.6. Alvey Vision Club.

[Lowe,] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. 60(2):91–110.

[Nikishaev,] Nikishaev, A. Feature extraction and similar image search with OpenCV for newbies.

[Rublee et al.,] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE.