

General remarks:

- Two approaches of robustly matching two images without a-priori information shall be compared. They differ in the initial computation of putative matches and the robust estimation. The first approach is summarized in the textbook R. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition (2003), p. 290, Algorithm 11.4. The second one can be found in the paper of Zhengyou Zhang, Richard Deriche, Olivier Faugeras, and Quang-Tuan Luong: *A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry*, INRIA, Technical Report, 1994.
- For the whole programming assignment we propose an evolutionary approach. First get the most simple solution running and then add more and more features.
- This structure is not exactly a rule, it is more like a guideline. Cool ideas, extensions and improvements are welcome!

1 Approach described by Hartley/Zisserman

All page numbers in this section refer to the Hartley/Zisserman book.
The algorithm is summarized on page 290.

1.1 Interest Points

Use the Harris corner detector (`corner_detector.m`) available on the web side. For more information on corner definition/detection have a look at *Introductory Techniques for 3-D Computer Vision* by Emanuele Trucco & Alessandro Verri

1.2 Putative Correspondences

see p.124.

Here, use the cross correlation method as described below in 2.1.2.

1.3 RANSAC Robust Estimation

see pages 116-122.

You should start with a simple version and add features like automatic sample number computation on and on. You can try using the 8-point or 7-point algorithm for the estimation of the fundamental matrix F (And please compare the results if you try both). As distance measure we propose the symmetric transfer error for starters...

1.4 Non-linear Estimation

see pages 597-602. You can use the MatLAB function `levmar.m` available on the web side. It expects a function pointer that evaluates the error vector and the jacobian of the function.

1.5 Guided Matching

see page 291 (iii).

The guided matching is optional in the sense that it is possible to get results without guided matching (needed for full points, however ;-)).

2 Zhang's Approach

2.1 Finding Candidate Matches by Correlation

2.1.1 Extracting Points of Interest

Use the Harris corner detector (`corner_detector.m`) available on the web side.

2.1.2 Matching Through Correlation

In a search window (n is quarter of image height, m is quarter of image width) define a correlation window ($d_u = 7 \times d_v = 7$ pixels) and let it evaluate the correlation between two points (pixels) (m_1, m_2) using the formula

$$C(m_1, m_2) = \frac{\sum_{i=-n}^n \sum_{j=-m}^m [I_1(u_1 + i, v_1 + j) - \overline{I_1(u_1, v_1)}] \times [I_2(u_2 + i, v_2 + j) - \overline{I_2(u_2, v_2)}]}{(2n + 1)(2m + 1) \sqrt{\sigma^2(I_1) \times \sigma^2(I_2)}}.$$

For further details refer to Zhang's paper, page 10.

2.2 Disambiguating Matches Through Relaxation

By just using correlation a feature point in the first image can correspond to more *candidate* matches in the second image and vice versa. That is, the correlation gives ambiguous matches since a feature point has only one true match.

2.2.1 Strength of Match

The strength of match is a measure of support for each candidate match. It examines the neighborhood of both feature points of the candidate match (m_1, m_2) and assumes to have a good match if there are many candidate matches in the neighborhood which have the same position relative to the respective feature point. This criteria can be encapsulated in a formula given on page 11 in the Zhang paper.

2.2.2 Relaxation

If we sum up the strength of matches of all candidate matches, we can get rid of ambiguous matches by minimizing this sum (because then, all candidate matches that lie within one neighborhood are collapsed to one single match. This iteration can be summarized as

iterate {

- compute the matching strength for each candidate match
- update the matches by minimizing the total energy function

} **until the convergence of the energy**

The update can be achieved using some strategy, like *winner-takes-all*, *looser-take-nothing*, *some-winners-take-all*. Follow the approach of Zhang by implementing *some-winners-take-all*.

2.3 Robust Estimation of the Epipolar Geometry

Zhang uses a different robust matching (Least Median of Squares) as well as a different parameterization (both epipoles as parameters) of the fundamental matrix to enforce the singularity of the matrix. It is well described on pages 14-20. Use this method to estimate F . Use this constraint to find new matches.

2.4 Stereo Matching

Now the matches can be recalculated using the epipolar constraint. We know that the corresponding point m_2 in the second image of a feature point m_1 in the first image has to lie on the epipolar line given by $l = Fm_1$.

3 Comparison of Methods

3.1 Visual Inspection

Visualize your results adequately. Draw the epipolar lines in the images as well as the points. You can use the methods available on the web side (`hcross.m`, `hline.m`, `plotPoints.m`).

3.2 Residual

Compare the root mean square errors of both implementations.

3.3 Performance

Which method is faster? Try to speed up your implementation of Zhang's paper by using the bucketing technique proposed in the paper on pages 18-20.

That's it. We hope you enjoyed the flight. Now you should get really cool matches in your two favorite images. Maybe you want to test other images with different properties (only translation, only rotation, etc.) - up to you.