Exercise session 6 : Stereo Vision

This session is about stereo vision. Stereo is one possibility to obtain 3D reconstructions. The exercises will guide you through that process. Beginning with finding correspondences, defining the fundamental matrix, extracting the external calibration and finally ending with a simple 3D reconstruction.

Most exercises begin with a paper-and-pencil theoretical part, which allows you to really think things through before just heading off experimenting with parameters. You can verify your results on Matlab experiments afterwards. Since this is not a Matlab course, most of the programming has already been done for you. You will only have to fill in some blanks, change small parts of code, set some parameters etc...

All necessary images and Matlab code can be found at Toledo

1 Cross-correlation

One way to describe and match features is through Normalized Cross-correlation. For two image regions I(x,y) and J(x,y) the normalized cross correlation is defined as :

$$NCC(I,J) = \frac{\sum_{x,y}(I(x,y)-\mu(I))(J(x,y)-\mu(J))}{\sqrt{\sum_{x,y}(I(x,y)-\mu(I))^2}\sqrt{\sum_{x,y}(J(x,y)-\mu(J))^2}} \tag{1}$$

where $\mu(I)$ and $\mu(J)$ is the average of the image I and J.

- Which image transformations this method invariant to?
- Implement normalized cross-correlation matcher for $N \times N$ patches of an image. Fill in the missing parts in `NCC.m` and test your implementation using `ex0.m`

2 The Automated Correspondence Problem

In order to match a point automatically to a point in another image, one can use different methods. Let us review the definition of NCC:

$$NCC = \frac{N(I_1, I_2)}{\sqrt{N(I_1, I_1), N(I_2, I_2)}} \tag{2}$$

$$N(I, J) = \iint_W (I(x, y) - \overline{I})(J(x, y) - \overline{J}) dxdy \tag{3}$$

In practice the double integral is computed as a summation over a window with size 5x5, 7x7, 9x9 or larger.

- Why will this work?
- What pre-conditions does the algorithm take? When will it in fact work?
- Why is this function called *normalized* cross correlation and what is the purpose of this normalization?

Run the `ex1.m` script to test the results (you use the files `harris_filter.m` and `NCC.m`). Test it on the kasteel and esat images.

- Which images work best? Why?
- Play with the parameters. Try optimizing for a maximum of good correspondeces without any false matches.
- Does it work for all regions in the images? Which ones are difficult?

For the following exercises, work with the images **esat3.jpg** and **esat4.jpg**. Run `ex1.m` again to find good correspondeces. You can add more points by left-clicking near a feature point in the left image. If the match found by NCC is not correct, you can change it by left-clicking near the correct feature point in the right image. You can remove bad matches by right-clicking near one the two corresponding feature points. Click on the right triangle to save your points for the next exercise.

3 Computing Fundamental Matrices

The fundamental matrix comprises the relationship between two images. It describes the epipolar geometry constructed by the two camera setups. As we finally want to reconstruct the scene in 3D, we'll need internal and external calibration from both camera's. Estimating the fundamental matrix is one way to go.

- Which relationship is defined by the fundamental matrix F? Write down the mathematical equation and draw the geometrical consequences.
- What is the rank of F? Why?
- How can we use corresponding points between two images to estimate F?
- How many correspondence point pairs do we need?

Make sure you found some good matches using the `ex1.m script`. The points are only saved if you click on the red triangle in the upper corner of the image. The `ex2.m` matlab script then loads those correspondence pairs, computes the fundamental matrix, gives an error measure and saves the result. You will have to complete it with your answers of the previous questions at the `TODO` marks to make it work.

- Do you get a good estimate for F?
- What error measure was used? Why can't we use $e = \sum p'_{i,1} F p_{i,2}$?
- Look at the numerical conditioning of the problem. Do you see a problem?
- Adapt your code to overcome this problem. Is your estimate better now (look at the error measure)?
- What is the rank of your estimated F? Why does it differ from the theoretical one?

4 Epipolar Geometry

The `ex3.m` script allows you to see the fundamental matrix in action. As you learned from the previous exercise, F defines the epipolar geometry between the image pair. This means that every point in one image is projected onto a line in the other one.

- What's the equation of that line (for both views)?

Fill in the `TODO 1`'s in `ex3.m` as such and run the script using the images and fundamental matrix of the previous exercise. Click on some interest points.

- Do the epipolar lines go through the corresponding points in the other view? Try both views.
- What do you notice about the epipolar lines?
- How can that be explained geometrically?
- What is its mathematical cause?
- Click on interest points all on the same epipolar line. What do you see? Can you explain why?

The intersection point is called the epipole.

- What is the geometrical meaning of the epipole?
- Can you compute both epipoles from F?
- What if the estimated F doesn't have the theoretical rank?

Fill in `TODO 2` accordingly in the `ex3.m` script.

- Did you make a good estimate?

- What does the epipole tell you about the used camera setup?
- What do parallel epipolar lines tell you?

Run the `ex1.m` and `ex2.m` scripts again, but now only click on points from a small neighbourhood in the image, but still with enough features (e.g. a window).

- Is F well estimated?
- Does the epipolar geometry fit well for the whole image?
- Why not? Where in your code would you have noticed?

Run the `ex1.m` and `ex2.m` scripts again, but now only click on points within one plane.

- Is F well estimated?
- Does the epipolar geometry fit well for the whole image?
- Why not? Where in your code would you have noticed?