Sebastian Nowozin (nowozin@cs.tu-berlin.de) **version 1.0, 2004/05/16**

Image align model used in autopano-sift

The document describes the model used in the RANSAC filtering part of my autopano-sift package. As I am not sure if the model used is suitable, and it surely is improvable, I put this forth as request for comments. Please mail me at nowozin@cs.tu-berlin.de.

1 Basic idea

RANSAC (RANdom SAmple Consensus) is an algorithm that allows the filtration of a number of input elements to output elements, removing unwanted ones. What is unwanted is defined by providing a model to the algorithm, which fullfils two things:

- The model can be "fit" using a small number of input elements.

  That is, all the parameters of the model can be reconstructed from a fixed, preferably small number of input elements.

- The model can output a "fitness" value of a novel element once it has been "fit".

Without detailing RANSAC - its not too difficult and good explanations can be found in [1] - it can be summarized as: RANSAC finds the most suitable parameters for the model by randomly testing a number of samples. The amount of testing necessary to be reasonably confident of the correctness of the parameters depends on two things:

- w, the fraction of points that is known to be correct. Often we have to estimate w.
- n, the number of elements required to fit the model.

Then, we would need k iterations and fittings of the model. The final formula with some confidence added is

$$k = w^{-n} + f \cdot SD(k) = w^{-n} + f \cdot \frac{\sqrt{1 - w^n}}{w^n}$$

Where $SD(k)$ is the standard deviation of k. f is a choosen factor, which we usually want to choose around some small integer value.
The rest of this text deals with the model used.

2 Model

The model's purpose is to provide a mapping of coordinate systems between two images. As both images are two dimensional and we assume that for the purpose of panoramic image creation they are both roughly in the same scale, we choose the simplest model possible: a two dimensional transformation matrix M for homogenous coordinates from image I_2 to I_1. To do this, we create the matrix and "fit the model" from two image keypoint pairs. That is, we have two lines in each image, A_1 to B_1 in image I_1, and A_2 to B_2 in image I_2. The coordinate system is anchored so that one could transform coordinates relative to this lines afterwards. The A_1 and A_2 pair is a keypoint match, and they represent - to a large probability (w) - the same image feature in both images. Likewise, B_1 and B_2 are matches. The process of coordinate transformation is divided into:

- Translation T_1. The A_2 point is translated into the coordinate origin, so $T_1 = T(-A_{2_x}; -A_{2_y})$.

- Rotation R. The line is rotated by an angle α so that the orientation is the same.

$$\alpha = \text{atan2}\left(B_{1_y} - A_{1_y}; B_{1_x} - A_{1_x}\right) - \text{atan2}\left(B_{2_y} - A_{2_y}; B_{2_x} - A_{2_x}\right)$$

- Scaling S. The coordinates are scaled so the two lines are the same length. As x- and y-axis are scaled the same, we can use a scaling factor s with $s = \left(\frac{|B_1 - A_1|}{|B_2 - A_2|}\right)$
- Translation T_2. The A_2 point is translated into the position of A_1, so $T_2 = T(A_{1_x}; A_{1_y})$.

The situation is shown in figure 1.

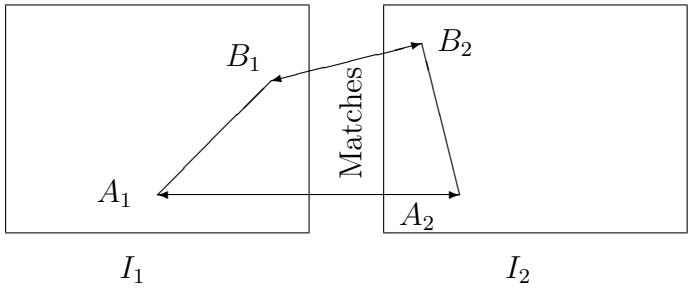

Figure 1: Two two lines used to fit the transformation from I_2 to I_1

Altogether we get the transformation matrix M with

$$M = \begin{bmatrix} s \cdot \cos(\alpha) & s \cdot (-\sin(\alpha)) & s \cdot \left(\cos(\alpha) \cdot (-A_{2_x}) - \sin(\alpha) \cdot (-A_{2_y})\right) + A_{1_x} \\ s \cdot \sin(\alpha) & s \cdot \cos(\alpha) & s \cdot \left(\sin(\alpha) \cdot (A_{2_x}) + \cos(\alpha) \cdot (-A_{2_y})\right) + A_{1_y} \\ 0 & 0 & 1 \end{bmatrix}$$

2.1 Distance to model fit

For every point $P = (x; y)$ in I_2 we can determine the expected position in I_1 by multiplying P's homegenous coordinates with our matrix:

$$P' = M \cdot P = M \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

We do this with every position from I_2 we have a keypoint match to I_1 for. Now we can compare the model-expected and real keypoint position from the matching keypoint in I_1. The distance is simply the geometric one. We yield a function d which takes the point P in image I_2, the matrix M and the matching keypoint K in image I_1:

$$d(P, M, K) = |K - M \cdot P|$$

The match can be in one of four categories:

- d is small, match is correct.

  This is of course, the most desired case. If the distance d is small, we can be reasonably sure that the keypoint fits into the geometric model.

- d is small, match is incorrect.

  This case is unlikely, but can occur if there is a keypoint match that also roughly fits into the geometric model. The only case I can think of would be repeating elements that are very near to each other, such as rows of windows next to each other. At least in my `autopano-sift` package, there are two mechanisms that make the occurance at the RANSAC stage unlikely:

  - The keypoint match quality term accounts for second-best ratio.

    The term that assigns a "quality of match" to each keypoint match includes the distance to the second-best match. If the second best match is a good match, too, then the first best match becomes a low-quality match, as it is likely to be a repeating or similar element. The idea of this is from the original SIFT paper [2].
  - Join-matches are removed.

    Whenever there are two features mapping to one feature in another image both matches are removed, because we can hardly tell which is the correct one. For repeating elements this is likely to remove most occurances of invalid matches.

- d is large, match is correct.

  This is something we do not want for panoramic imaging and is removed. This is the case for moving objects, such as people, cars, etcetera, where the matches correctly represent the same features in both images, but the object moved relatively to the background.

- d is large, match is incorrect.

  This match is removed. Most removals are due to this case.

2.2 Adjusting for model origin

The expected model position and the real position are most likely to differ only very small in areas around the two anchoring point pairs, A and B. Vice versa, they are likely to diverge more and more the more distant the point becomes. Also, the inherent distortion of the image projection will lead to increasing differences the farther we get from the anchors. As such, we modify the distance function to account for this. First, we define a term that reflects the relative distance from the anchors A and B for the point X:

$$d_a(A, B, X) = |X - A| + |X - B| - |A - B|$$

This gives a smooth eliptical weighting around the line, as shown in figure 2.

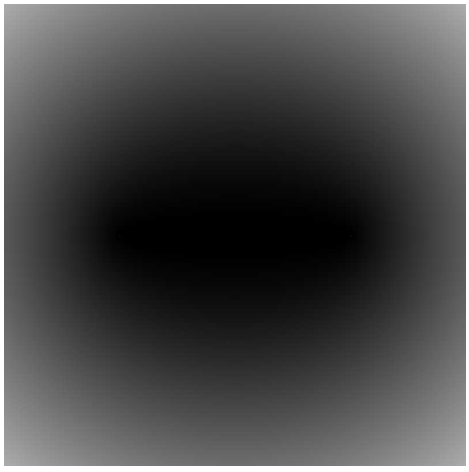

Figure 2: Distance weighting around the lines, black: zero, white: 1.0

The final term includes the original plus a factor f, which determines how much the distance from the line decreases the distance and the RANSAC acceptable fit threshhold t:

$$d(P, M, K, A, B) = |K - M \cdot P| - f \cdot d_a(A, B, P) \cdot t$$

In the current version of `autopano-sift` I use $t = 4.0$, $f = 16.0$.

3 Further improvements

- It may be possible to improve the distance-adjust term by fitting a more precise geometric model that accounts for more characteristics of the input image, such as barrel distortion, focal length, etcetera.
- It may be possible to combine the keypoint match quality with the geometric fit value the model to make a more unified decision. Currently the match quality is completely discarded before the RANSAC matching.
- It is possible to include the results of the SIFT matching better: SIFT keypoints also have an orientation assigned, not just a position. Include that into the model.

References

[1] David A. Forsyth, Jean Ponce, "Computer Vision - a modern approach", ISBN 7-302-07795-9

[2] David Lowe, "Distinctive image features from scale-invariant keypoints", `http://www.cs.ubc.ca/ lowe/papers/ijcv04-abs.html`