Master's degree in Computer Engineering for Robotics and Smart Industry - a.y. 2022/2023
Computer Vision

Automatic 2D mosaicing

Lorenzo Busellato - VR472249 - lorenzo.busellato_02@studenti.univr.it

CONTENTS

Master's degree in Computer Engineering for Robotics and Smart Industry - a.y. 2022/2023
Computer Vision

Automatic 2D mosaicing

I. INTRODUCTION

Image mosaicing is a method of combining multiple images of the same scene into a single, larger image. The mosaicing process can be broadly divided in five steps:

- Feature point extraction
- Feature point matching
- Robust homography computation
- Image warping
- Image blending

Feature points are be found using Scale Invariant Feature Transform (SIFT). Their matching is obtained by comparing the descriptors resulting from SIFT. The homography computation is made statistically robust to outliers using RANdom SAmple Consensus (RANSAC). The images are correctly aligned using image warping, i.e. by using the homography between them. The quality of the resulting mosaic is improved using image blending, which makes the colors more uniform near the seams between the images.

II. OBJECTIVE

This project's objective is to develop a software application for the automatic generation of a 2D mosaic from a set of images of a planar scene.

III. METHODOLOGY

A. SIFT

Scale Invariant Feature Transformation (SIFT) is a method used to identify a set of feature points, or features, within an image.

Features are regions within an image that carry some information about the image's content. Features can often be associated to structures within the image, such as corners or edges.

To be able to match detected features in a set of images, the features should be invariant to:

- Scale
- Illumination
- Rotation

Given an input image I, the algorithm first finds candidate features as follows:

1) The image is downsampled four times, yielding a set of scaled images.
2) Each scaled image is convolved five times with a Gaussian kernel, yielding five sets of blurred images called **octaves**.
3) For each octave, four **difference-of-gaussians** (DoG) images are computed by taking the difference of adjacent images, yielding sixteen DoG images.
4) For each pixel in each DoG image, features are defined as the local maxima and minima in the 3x3x3 region surrounding the pixel in the previous, current and next DoG image in the octave.

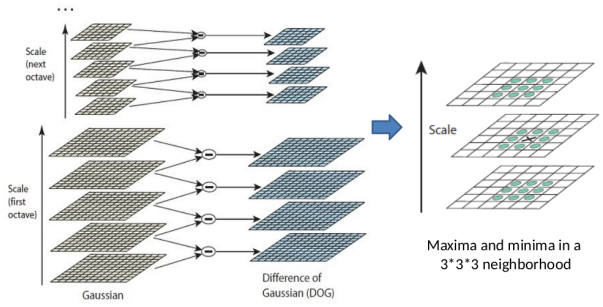

Fig. 1. From image pyramids to candidate feature points (image source [1])

The scaling and blurring introduce invariance to scale and illumination conditions. To introduce invariance to rotation, the orientation associated to the feature is estimated by computing the histogram of gradients for the 16x16 pixel region on the Gaussian pyramid scale the feature was found at. The orientation associated to the feature is the histogram bin corresponding to the highest peak in the histogram.

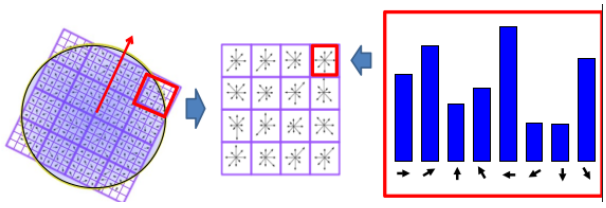

Fig. 2. SIFT descriptor (image source [1])

The feature descriptor finally is computed by considering the 16x16 region around the feature, which is then divided into 16 4x4 sub-regions for each of which a 8-bin histogram is computed. The feature descriptor is then the concatenation of the 16 resulting histograms, i.e. a 128 by 1 vector.

Feature matching is done with an heuristic based on distance. Given two images and the corresponding features extracted with SIFT, for each feature of the first image the distance to each feature of the second is computed. The candidate match is the feature pair that results in the smallest

distance. To reduce the influence of outliers, the candidate match is accepted only if the distance is smaller than an arbitrary threshold.

B. Homography

Any two images of the same planar scene can be related through a linear relationship called homography.

Let M be a 3D point in the reference frame centred on the left camera and let M' be the same point in the reference frame centred on the right camera. The perspective matrices that describe the cameras are then:

$$P = K[I \mid 0] = [K \mid 0] \qquad P' = K'[R \mid T] = K'G$$

The two points are linked by the rototranslation matrix G:

$$M' = GM = RM + T \tag{1}$$

Let $n^T \tilde{M} = d$ be the equation of the plane containing the scene, where n is the plane normal and d is some scalar representing the distance of the plane from the origin. The projections of M and M' on the image planes are:

$$m \simeq KM \qquad m' \simeq K'M'$$

Since M, by construction, belongs to the plane we have:

$$n^T M = d \implies \frac{n^T M}{d} = 1$$

Plugging the fraction into the equation 1 (treating T as $1 \cdot T$):

$$M' = RM + \frac{n^T M}{d} T$$

Therefore:

$$M' = K'^{-1} m' = \left(R + \frac{n^T}{d} T \right) M = \left(R + \frac{n^T}{d} T \right) K^{-1} m$$

Finally:

$$m' = K' \left(R + \frac{n^T}{d} T \right) K^{-1} m = H_\pi m$$

H_π is the homography, i.e. the linear relation between the pixels in the two images.

To compute the homography, we start from a set of n conjugate points (m_i, m'_i), and we want to estimate the matrix H such that:

$$m'_i = H m_i \quad i = 1, \ldots, n$$

Taking the cross-product of both sides with Hm_i yields:

$$m'_i \times H m_i = 0 \implies [m'_i]_x H m_i = 0$$

where $[m'_i]_x$ denotes the skew symmetric matrix of m'_i. To get a linear system, we need multiple instances of this equations:

$$vec([m'_i]_x H m_i) = (m_i^T \otimes [m'_i]_x) vec(H) = 0$$

where $\otimes$ denotes the Kronecker product and $vec()$ the vectorization transformation.

The system can be solved for H using singular value decomposition (SVD), for which at least four conjugate point pairs are needed, since each pair gives two linearly independent equations and H has nine unknowns.

C. RANSAC

RANdom SAmple Consensus (RANSAC) is an iterative algorithm which aims to improve the estimation of a model given a set of observations.

The algorithm randomly samples the observations and creates a model estimate on this subset. The amount of data points that are closely explained by the computed model is called its consensus. This procedure is repeated a number of times, and the resulting best-estimate for the model is the one with the highest consensus (i.e. with the least amount of outliers).

The algorithm is defined as follows:

- Input: a set of observations (y_i, x_i), $i = 1, \ldots, n$, a threshold ε and a number of iterations k.
- Algorithm:
  1) Repeat k times:
     a) Take a random sample of p elements from the observation set.
     b) Use the subset to estimate a probe model $\hat{\theta}_j$ (e.g. by regression).
     c) Compute the consensus set of the probe model:
     $$C = \{y \mid (y_i - f(x_i, \hat{\theta}_j))^2 < \varepsilon\}$$
     where f is the function that relates x to y given the model $\hat{\theta}_j$.
  2) Among all the consensus sets, pick the one with the most elements.
  3) The probe model corresponding to the most numerous set is the best estimation. Its consensus set is the set of inliers of the data set. The remaining observations are considered outliers.

RANSAC will be used to refine the set of detected features in each image. This means that features of a given image that are unlikely to correspond to features in a subsequent image will be treated as outliers and therefore removed.

The algorithm will also be used to improve the computation of the homography, resulting in the transformation that links as many feature matches as possible.

D. Image warping

To obtain the mosaic, the images are treated sequentially. Given two images, the first is treated as the reference image, while the second image undergoes a procedure of warping. Warping means that the homography that relates the two images is used to project the second image onto the plane of the reference.

E. Image blending

Once the second image has been warped, a simple superposition of it with the reference image yields the mosaic. The main issue is that the color blending is not uniform in the overlapping region and especially near the seams between the images. A first approach is to create a binary mask for the overlapping region, and use the averaged values of pixel intensities of the two images in the overlap, thus correcting the intensities in the stitch. This simple approach does not fix however the noticeable seams there are when the images are too misaligned.

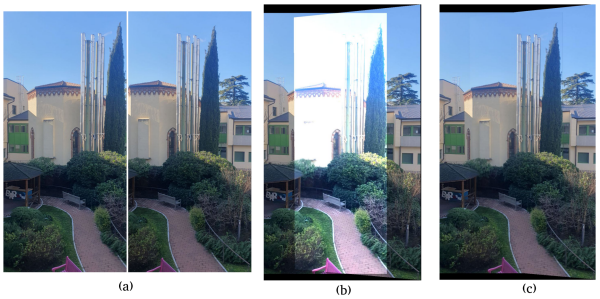

Fig. 3. Reference images (a) and the resulting mosaic without blending (b) and with blending (c).

To improve the quality of the blending, a linear alpha blending procedure was implemented. The binary mask that isolates the overlapping region was used to create a gradient. The gradient is used as an alpha mask, whose values are used to smoothly blend the images near the seams.

$$I(x,y) = \alpha(x,y)I_1(x,y) + (1-\alpha(x,y))I_2(x,y)$$

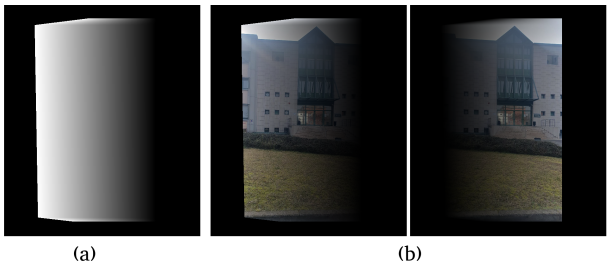

Fig. 4. Alpha mask (a) and its application to two source images (b).

IV. IMPLEMENTATION

The code for the project is available in a public GitHub repository [2].

In the main file (*main.m*), a struct of parameters is set to influence the execution. The main parameters are *ransac_thresh* and *pixel_tolerance*, that set the tolerance for inliers for the RANSAC procedure respectively on the features and on the homography, as well as *ransac_iter* that sets the maximum number of iterations of the RANSAC algorithm. Another parameter is *blending*, which sets which blending method is used in the image merging procedure. The implemented methods are *'none'*, with which no blending is applied, *'average'*, with which the averaging of pixel intensities is used, and *'linear'*, with which the linear alpha blending method is used.

The pipeline is as follows:

- Initialize the current mosaic as the first source image.
- For each remaining source image:
  1) The SIFT features and descriptors are computed for the current mosaic and the next source image.
  2) The computed features are matched based on their descriptors.
  3) The matches are refined using RANSAC.
  4) The resulting matches are used to robustly estimating the homography using RANSAC.
  5) The estimated homography is used to project the source image onto the plane of the current mosaic.
  6) The current mosaic and the projected source image are composed and then image blending is applied.

V. TESTS AND RESULTS

The following tests are performed on three sets of images taken with a smartphone camera.

The following tests were performed:

- Test 1: 5 pictures of set 1 (figure 5), 1000 RANSAC iterations, pixel tolerance of 5 pixels, no blending.
- Test 2: 5 pictures of set 1, 1000 RANSAC iterations, pixel tolerance of 5 pixels, average blending.
- Test 3: 5 pictures of set 1, 1000 RANSAC iterations, pixel tolerance of 5 pixel, linear blending.
- Test 4: 5 pictures of set 1, 1000 RANSAC iterations, pixel tolerance of 25 pixels, linear blending.
- Test 5: 5 pictures of set 1, no RANSAC, linear blending.
- Test 6: 5 pictures of set 2 (figure 6), 1000 RANSAC iterations, pixel tolerance of 5 pixel, linear blending.
- Test 7: 5 pictures of set 3 (figure 7), 1000 RANSAC iterations, pixel tolerance of 5 pixel, linear blending.

Fig. 5. Set 1

Fig. 6. Set 2

Fig. 7. Set 3

A. Test 1

The test (figure 8) shows that the produced mosaic is convincing, but the lack of color blending in the overlaps is very noticeable. Therefore the need for methods for image blending is apparent.

B. Test 2

The test (figure 9) shows the usage of the average color blending method. The resulting mosaic is better than the one produced by test 1, but there are some aberrations, for instance in the branches of the tree, and the seams between the images are noticeable.

C. Test 3

The test (figure 10) shows the usage of the linear alpha blending method. The produced mosaic is quite better than the one produced in test 2, with reduced aberrations and no noticeable seams.

D. Test 4

The test (figure 11) shows the effect of increasing the pixel tolerance for the RANSAC algorithm applied to the homography computation. As expected, the quality of the mosaic decreases, especially towards the right, where the errors accumulated in the computation of the homographies become quite noticeable.

E. Test 5

The test (figure 12) shows the importance of having a statistically robust homography estimation, by removing the usage of RANSAC to estimate the homography and instead computing from a random selection of 4 pairs of conjugate points. The algorithm had to be stopped at the first iteration, because the distortion is clearly too great.

F. Test 6

The test (figure 13) shows the mosaicing of set 2. The set consists of pictures of the same scene as set 1, but under different light conditions. The produced mosaic is still convincing.

G. Test 7

The test (figure 13) shows the mosaicing of set 3. The set consists of pictures of a different scene than set 1. The produced mosaic has some aberrations in the top part, due to the reflections of the sun.

VI. CONCLUSIONS

The automatic 2D mosaicing procedure has been successfully implemented.

Tests 1 through 3 showed the importance of having a good light blending step, with the linear alpha blending technique being the best option of the two tested.

Test 4 showed the limits in increasing the pixel tolerance for the RANSAC procedure applied to the homography estimation. Increasing the pixel tolerance results in a less convincing mosaic.

Test 5 showed the importance of having a statistically robust estimation of the homographies, showing that a random selection of conjugate points is not enough to reliably estimate the homography matrix.

Tests 6 and 7 showed the adaptability of the procedure to different light conditions and to different scenes.

REFERENCES

[1] U. Castellani. Lecture slides of the computer vision course, master's degree in computer engineering for robotics and smart industry, 2022.
[2] Automatic 2d mosaicing. https://github.com/lbusellato/cv_project.

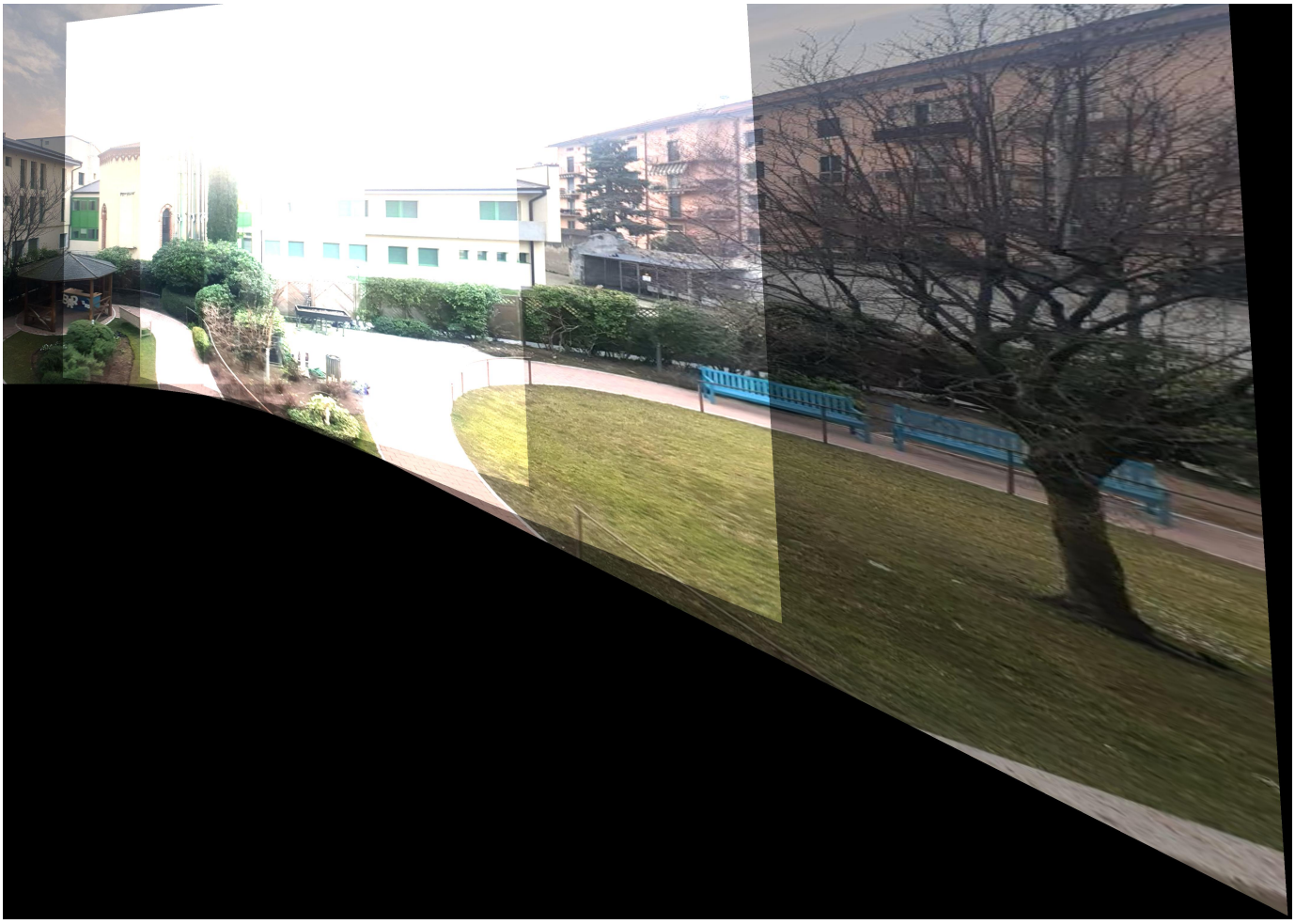

Fig. 8. Test 1: Mosaicing without color blending.

Fig. 9. Test 2: Mosaicing with average blending.

Fig. 10. Test 3: Mosaicing with linear alpha blending.

Fig. 11. Test 4: Mosaicing with linear alpha blending and increased pixel tolerance.

Fig. 12. Test 5: Mosaicing without RANSAC.

Fig. 13. Test 6: Mosaicing on the same scene with different light conditions.

Fig. 14. Test 7: Mosaicing on a different scene.