Automatic Analyser of Satellite Pictures

G9: Chaden Ouertani, Linjing Zhang,
Pablo Sánchez, Artagan Öcal, Yang Zhang

July 2021

"Hint: In the **Readme-file** for the program is a user guide included. Please refer to it if you have questions."

1 Program Structure

As shown in the Figure 1.1 , the structure of the program consists of reading the image and then converting the RGB pixels into HSV pixels for colour recognition, which allows the user to see changes in the environment, such as the expansion of the city. The RGB image is converted to grey scale and the SURF algorithm is used to identify and match feature points. The image can then be calibrated using the transmission matrix obtained by the SURF algorithm. The difference between the two satellite images can be obtained by the SSIM algorithm and the mark changing bounding can be circled by using L1 Norm.

2 Algorithm

2.1 SURF Feature Detection and Matching

SURF (Speeded Up Robust Features) is a further optimisation of the SIFT features.It constructs the pyramid scale space based on the Hessian matrix and uses a "box Filter" to simplify the Gaussian filtering without downsampling. [2] Compared to the Gaussian pyramid process of the SIFT algorithm, the speed of the SURF algorithm is improved. In the Sift algorithm, the image size of each octave is not the same. The next octave is the downsampling (1/4 size) of the previous set of images; In each octave their size is the same, the difference is that they use different sigma scale. For Surf algorithm, the size of the image is always the same, only the size of the Gaussian blur template is changed and no downsampling process is required, so SURF algorithm can save time compared to Sift.[3]

For the feature description, SURF uses the "Wavelet Responses". A neighbourhood around the feature point is selected and divided into sub-regions, and then for each sub-region the "Wavelet Responses" are taken and plotted to obtain a SURF feature descriptor. The Laplace sign already calculated during

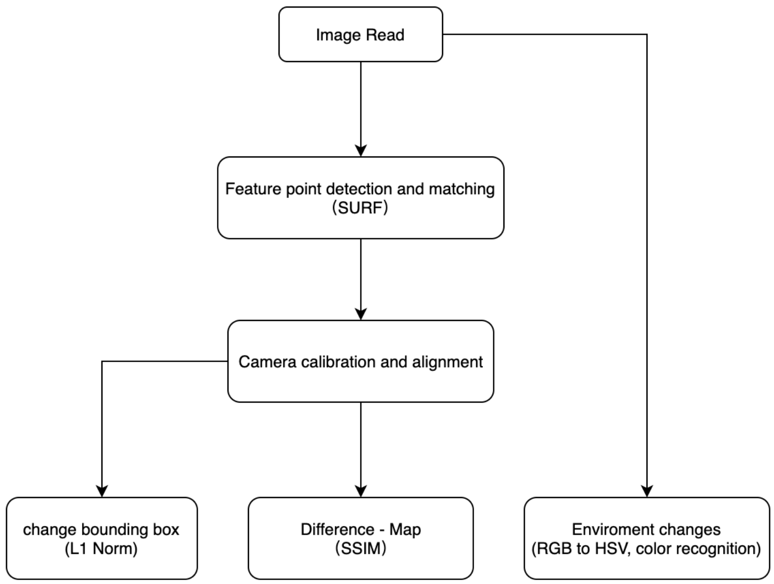

Figure 1.1: Flowchart for program "Automatic Analyser of Satellite Pictures"

detection is used for underlying points of interest. This sign distinguishes light blobs on a dark background from the reverse case. In matching, the features are only compared if they have the same contrast type, which enables faster matching.[2]

Figure 2.1 shows all matching feature points under SURF algorithm. Based on these feature point transformation matrices can be calculated and then image calibration can be achieved.

2.2 SSIM/L1 norm Change Map

The l1norm is a very common way to compare images, which is to take the absolute value of the subtraction of the image pixel values. But this way is too simple and does not consider the characteristics of the image: the correlation of neighboring pixel points. Therefore we also use the SSIM method.

The structural similarity index (SSIM) was developed to achieve greater matching with human visual perception. In SSIM, images are divided into patches and then compared. Rather than calculating absolute errors, structural information is considered. SSIM is defined for two patches $\vec{y}$ and $\hat{\vec{y}}$, which is $K \times K$ large:

$$\mathrm{SSIM}(\vec{y}, \hat{\vec{y}}) = [l(\vec{y}, \hat{\vec{y}})]^{\alpha}[c(\vec{y}, \hat{\vec{y}})]^{\beta}[s(\vec{y}, \hat{\vec{y}})]^{\gamma}. \quad (1)$$

Similarity is measured in luminance $l(\vec{y}, \hat{\vec{y}})$, contrast $c(\vec{y}, \hat{\vec{y}})$, structure $s(\vec{y}, \hat{\vec{y}})$, the three perspectives. $\alpha, \beta, \gamma \in \mathbb{R}$ determine the weight of the three components

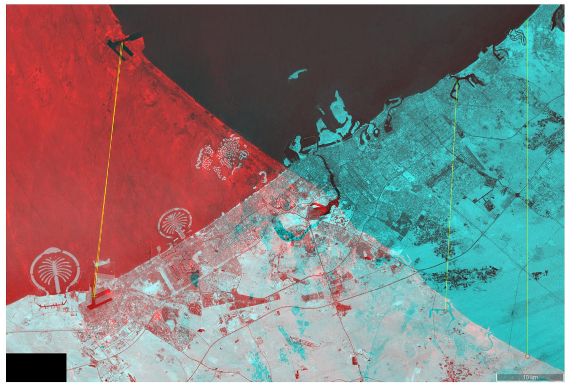

Figure 2.1: Feature point matching with SURF algorithm in Dubai in 12.2015 and 12.1990

for SSIM index, in this work $\alpha = \beta = \gamma = 1$ are defined the same as [4]

$$\mathrm{SSIM}(\vec{y}, \hat{\vec{y}}) = \frac{(2\mu_{\vec{y}}\mu_{\hat{\vec{y}}} + C_1)(2\sigma_{\vec{y}\hat{\vec{y}}} + C_2)}{(\mu_{\vec{y}}^2 + \mu_{\hat{\vec{y}}}^2 + C_1)(\sigma_{\vec{y}}^2 + \sigma_{\hat{\vec{y}}}^2 + C_2)} \quad \in [-1, 1]. \tag{2}$$

When choosing the method for image Change Map, we also compared these two methods with PCA-kmeans[1], as shown in Figure 2.2,The results of each method have their own advantages and disadvantages. k-means works well, but it is randomly divided into two classes and the labels of pixel points (1, 0) may be different, at the same time the computation time is long. l1 method is simple, the operation time is short, but the results are not so good. SSIM is more balanced in all criteria and performs well, but there are some boundary effects, because this algorithm will consider the neighbors.

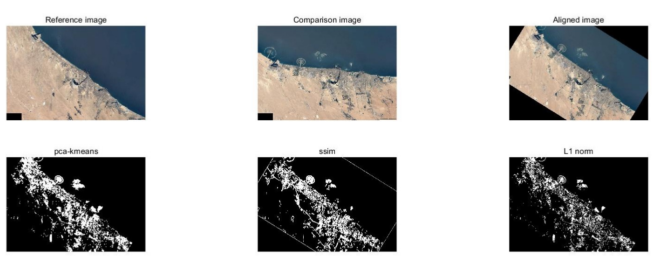

Figure 2.2: Change-Map use different Method

3 Features

3.1 Image Calibration

The image can be automatically calibrated by using the feature point matching obtained with the SURF algorithm and the calculated transformation matrix, as shown in the figure 3.1 showing the automatic calibration of the image Columbia Glacier in 12.2020 based on the reference image Columbia Glacier in 12.2000.

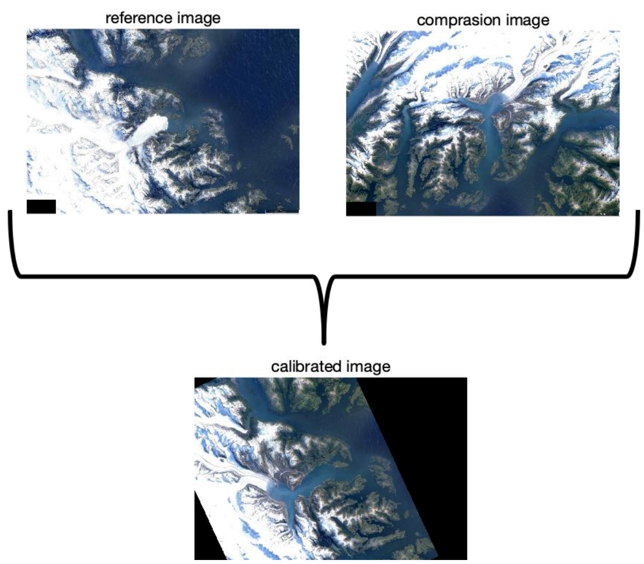

Figure 3.1: Automatic calibration for Columbia Glacier in 12.2020(reference image:Columbia Glacier in 12.2000)

However, for complex terrain such as rain forests in Brazil or images with different exposure and brightness or containing shadows of buildings at different angles such as Fraunkirche images. It is difficult to accurately match feature points in these images and therefore automatic calibration is not possible. In this case we use a manual selection method of feature points to calibrate the image. For this we use the Control Point Selection tool in matlab. In this control point selection tool the user is able to select feature points in two related images. When the control point selection tool is open, the user can add, move and delete similar features with the mouse. When the user has finished modifying the feature points, the selected feature points are exported by selecting "Close Control Point Selection Tool" from the File menu. By using the "fitgeotrans" method, the image is then rotated to the orientation of the reference image, thus enabling manual calibration of the image. This tool returns the coordinates corresponding to the validly selected moving and fixed control points in two numeric vectors. For example, in the figure 3.2 we manually selected seven matching feature points in two Frauenkirche images. In the upper left corner is the reference image (Frauenkirche in 08.2012) and in the upper right corner the image to be calibrated (Frauenkirche in 06.2021). By clicking on "add points", the user can add feature pairs to the image.

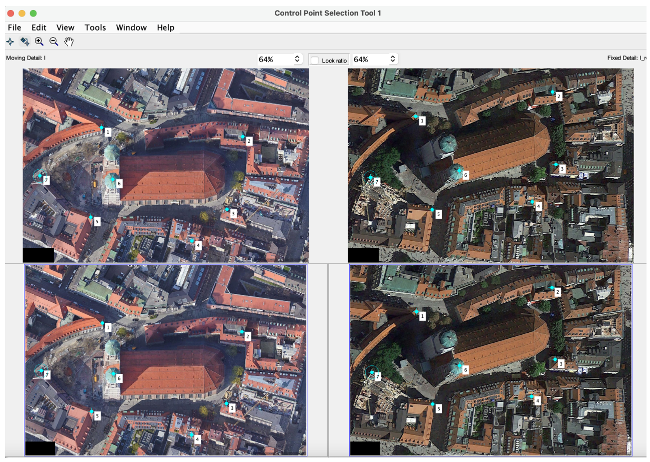

Figure 3.2: Manuel features points selection for Frauenkirche by using Control Point Selection tool

Here we recommend adding at least 4 pairs of feature pairs to ensure an accurate calibration image. The figure 3.3 shows the results of the calibration for the Frauenkirche.

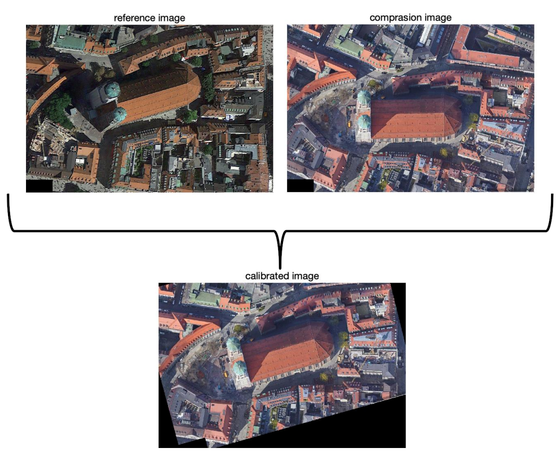

Figure 3.3: Manual calibration for Frauenkirche in 06.2021(reference image:Frauenkirche in 08.2012)

3.2 Environment Changes

For highlighting selected environments such as oceans (water), the pixels are sorted using a threshold. To do this, the images are first converted from RGB color space to HSV color space, as this allows easier selection by color. Then the pixels can be selected by values e.g. the pixels are blue if the pixel value is between 210 and 270. In addition, groups of pixels smaller than 50 pixels are ignored to avoid highlighting errors in the image. From these values, the blue pixels (water) can then be highlighted, as can be seen in the Figure 3.4.

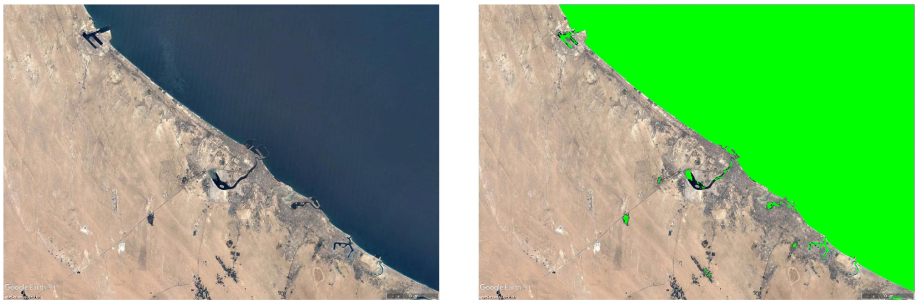

Figure 3.4: Highlighting the ocean

With this method, homogeneous surfaces such as water, snow or sand can easily be found and highlighted, but water is not always blue or buildings are not always gray, this then leads to incorrect results in the detections of these surfaces. A possible solution for this is that the user can select special profiles, such as colors, to examine an image by color since the user, knows for example what color the water in the image is.

In addition to this visualization option, the user is also shown the percentage of the selected environment in the image, such as the water on the image in Figure 3.4. Also, the user can select another image and see the change in the environment, such as the water, as a percentage.

3.3 Environment Classification

In this function, the image is divided into clusters using the k-means algorithm and these are marked and displayed, as shown in Figure 3.5.

In addition, a pie chart displays the size of each cluster as a percentage. With this function, users should be able to divide an image into clusters and find out the size of these clusters in relation to the other clusters, for example, to find out the area that plants fill in the image compared to the buildings. However, it can take a long time to calculate a large number of clusters, so we have limited it to 6 in our program.

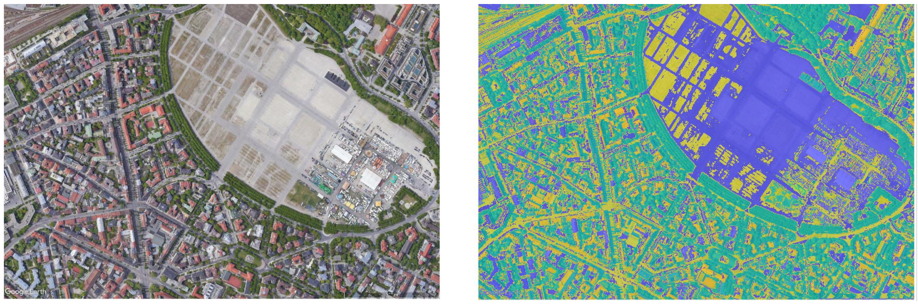

Figure 3.5: Clustering of the Wiesn

Figure 4.1: Difference map visualization(Brazilian Rainforest)

4 Mode of visualization

4.1 Difference map

As written in Chapter 2.2, we used ssim to create the difference map, and then filtered the small pixel groups. Final show the user a visualization of the binary map and the map overlaid on the comparison image. See figure 4.1 and figure 4.2.

4.2 Mark change bounding

We created the difference map using L1 norm and also performed the filtering of small pixel points. To take into account the effect of neighboring pixels, we used imdilate() in the image processing toolbox to expand the change pixel points and visualize the boundaries of different points on the two compared images. See figure 4.3 and figure 4.4.

Figure 4.2: Difference map visualization(Columbia Glacier)

Figure 4.3: Change bounding visualization(Kuwait)

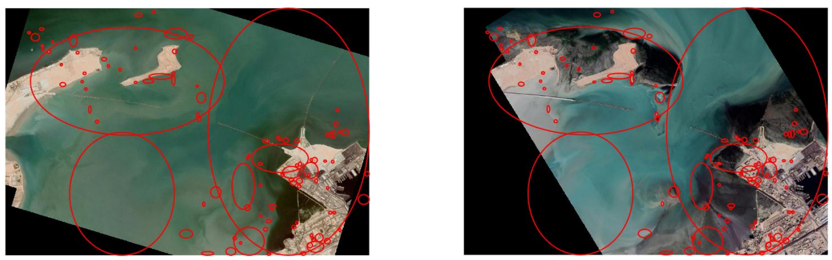

Figure 4.4: Change bounding visualization(Kuwait)

4.3 Time-lapse

With chapter 3.1 image calibration, all calibrated images are arranged in time sequence, which allows the user to see the changes in this location by selecting different dates. We show these changes in 'Time Lapse' mode, for example the figure 4.5 shows the change in Dubai over the last 30 years.

4.4 Statistic

To allow the user to more quickly find the time-points of large changes, we also provide a linear graph of the percentage change of the choose image compared to the reference image,as shown in Figure 4.6.

4.5 Classification

To simplify the analysis of single images, we allow the user to divide an image in up to six clusters, as shown in Figure 3.5.

5 Mode of change

5.1 large-little

To filter some changes, we provide some user-selectable thresholds, see figure 5.1. In the change map, ssim values below the threshold are counted as changes.

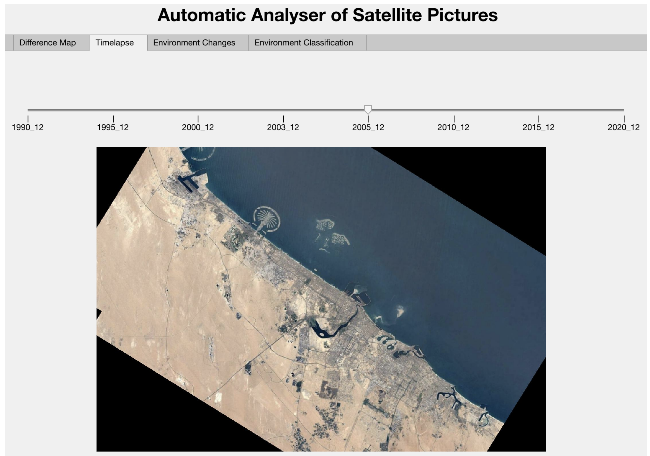

Figure 4.5: Time Lapse for Dubai

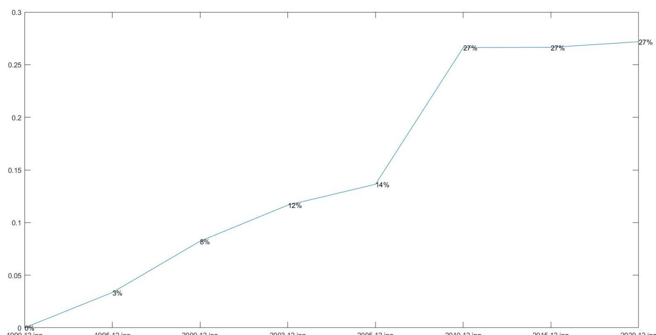

Figure 4.6: Graph of the Dubai Data set using the first image as a reference image

This threshold is set in the Detecting Strength slider, and the change size slider sets the size of the pixel groups that will be filtered out. In the boungdingblock, the chage size determines the size of the bounding.

5.2 quick-slow

We think that by relying on the data generated by chapter 4.4, and some other visualizations (such as time-lapse), users can find intervals with fast changes and have a fast or slow sense of the time point the picture is in.

Figure 5.1: Slider to control the size of the visualization changes

5.3 Changes in different environments

To illustrate the change explained in Chapter 3.2, we have plotted the change of the selected object in percent over the whole period of the available data as shown in Figure 5.2.

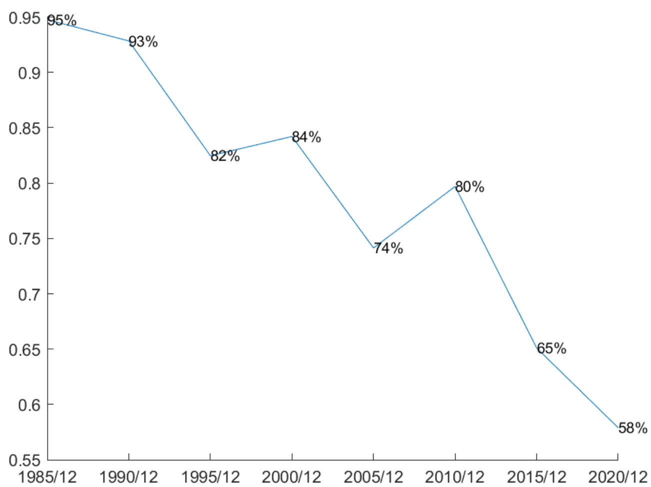

Figure 5.2: Change of forest over time

Figure 5.2 shows the change of the forest over time of the rainforest dataset.Especially the deforestation can be recognized by it

References

[1] Turgay Celik. Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE Geoscience and Remote Sensing Letters*, 6(4):772–776, 2009.

[2] Ebrahim Karami, Siva Prasad, and M. Shehata. Image matching using sift, surf, brief and orb: Performance comparison for distorted images. *ArXiv*, abs/1710.02726, 2017.

[3] S. Khan, Faheem Iftikhar, and Usman Akram. Geometry augmented surf with modified sobel for improved affine invariance in image matching. *2019 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–6, 2019.

[4] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.