Computer Vision

Report

Jiang, Xuefei 03745797 | Yan, Cheng 03745374 | Zhao, Ziqing 03739638 | Li, Runze 03740923 | Cao, Jimin 03738137

July 11, 2021

1 Motivation and Goals

Computer vision technology can be used in many fields, one of which is quite important is the field of satellite remote sensing technology.

Our goal of this project is to design an application with GUI based on MATLAB to visualize the changes between two or more satellite images of the same location. The detection of changes in the same location can be accomplished regardless of whether the images are rotated, the angle is switched or the brightness is different.

2 Project Structure

In this section, we will briefly introduce the structure and content of our project. The basic algorithm steps are shown in the following flowchart (Figure 1).

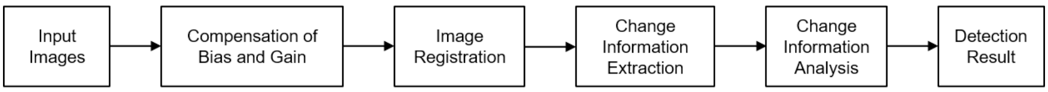


Figure 1: Project Structure

2.1 Input Images

As input, we choose datasets provided on Moodle including five location folders(Figure 2) which can be classified as urban area and natural area. Images in the same folder can have different rotated and translated points of view. Not only that, they can have different brightness and illuminations. In addition, we can also allow users to use their own satellite images datasets, provided that the images in the dataset have the same plotting scale. So we put a new location folder named "Aral sea" in the dataset for testing as well. In our Graphical User Interface(GUI), users can select any images as they want in our application. We need users to select one image as the reference image, and then select another as target image. At the same time, the whole dataset at a certain location is able to be made a GIF. These two images are used as input together, and we will compare their features and changes in the follow-up part.

(a) Frauenkirche (b) Columbia Glacier (c) Dubai (d) Kuwait (e) Wiesn

Figure 2: Input Dataset Examples

2.2 Images Preprocessing

After users have selected the input images, the first step our application needs to do is to preprocess the input images. Since we allow the input images in different brightness and illuminations as we said in subsection 2.1, the brightness and contrast between the input images could be very different, which means we need to normalize the scalling and shify of intensity values.

Therefore, our main task in the images preprocessing step is compensation of bias and gain. In this way, we can then deal with images with different brightness and contrast.

The first step of image intensity normalization is to transform the image from RGB color model to HSV model(hue, saturation and value), where the V-channel means the brightness of the image.

Then, perform normalization transformation on the V channel of input images. Suppose we have a n-dimensional reference V-channel image $I_{ref} : \{\mathbb{X} \in \mathbb{R}^n\}$ with mean value $\mathbb{E}_{ref}$, and a n-dimensional target V-channel image $I_{tar} : \{\mathbb{X} \in \mathbb{R}^n\}$ with mean value $\mathbb{E}_{tar}$.

The mathematical normalization model can be shown in the following formula:

$$I_{tar-norm} = I_{ref} - \mathbb{E}_{tar} + \mathbb{E}_{ref}$$

(a) Reference Image (b) Target Image (c) Illumination-normalized Image

Figure 3: Image Normalization

2.3 Image Registration

After image preprocessing, we eliminate the influence of different brightness and contrast on image processing. Another task we still need to do as we mentioned in subsection 2.1 is to eliminate the influence of different rotated and translated points of view on image processing.

In this subsection, we will briefly introduce the basic principles of image registration and the algorithm we choose to normalize the rotation angle and translated coordinate of selected images.

We first extract the features of the reference image and the target image separately. Then we use matched feature points to accomplish the images registration. In the process of finding feature points, we compared different algorithms such as: SURF, Kaze, FAST, Brisk, ORB and Harris.

Then we find that the Kaze algorithm[2] is scale and rotation invariant with a very high accuracy but relatively low computing efficiency. To be specific, Kaze algorithm can find sufficient matched feature points for image registration. By changing the parameters in the Kaze algorithm, the number of feature points and the quality of matched feature points can be alternated.

We set two different threshold values in the Kaze algorithm. During the registration of all images in on sub-dataset, the threshold value is 0.0008. In this way, some matching quality is reduced to get a shorter running time, because multiple images need to be processed. During the registration of one target image for further change detection, the threshold is set to 0.0004. Compared with Kaze and Surf algorithms, the remaining algorithms integrated in Matlab do not have the same satisfactory performances. After consideration, Kaze algorithm is chosen for registration of target images.

In the image registration process, we use help from computer vision tool box in Matlab. And we use 'imwrap' function and geometric transformation information to register images. Gif animination function is realised by combining series of registiered images. The performance of registration can be seen in the Figure 4.

(a) Reference Image (b) Target Image (c) Registered Image

Figure 4: Image Registration

2.4 Change Information Extraction

After aligning the images, we chose to form differential images using the unsupervised change detection method of PCA Kmeans[1]. Using this method avoids the need for multiple datasets to train the program under the supervised method, but instead directly detect change by making a direct comparison of two multi-temporal images considered without incorporating any additional information.

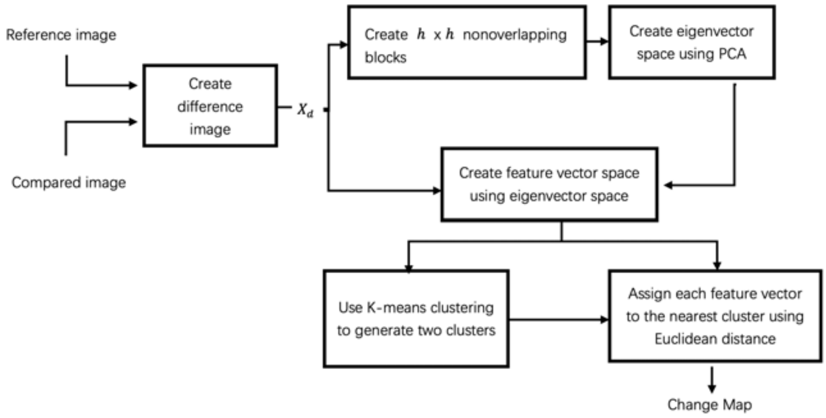


Figure 5: PCA K-means general scheme of the proposed approach[1]

When we want to use the PCA and K-means to detect the difference between the reference image and the target image,the proposed approach is made up of the following six main steps (see Fig. 5).

we first get two binary image of them and calculate the difference image. Next we partitioning difference image into hxh non-overlapping blocks, and that is used to create an eigenvector space using PCA. The feature vector space is created by projecting difference image matrix onto eigenvector space for each pixel at spatial location.

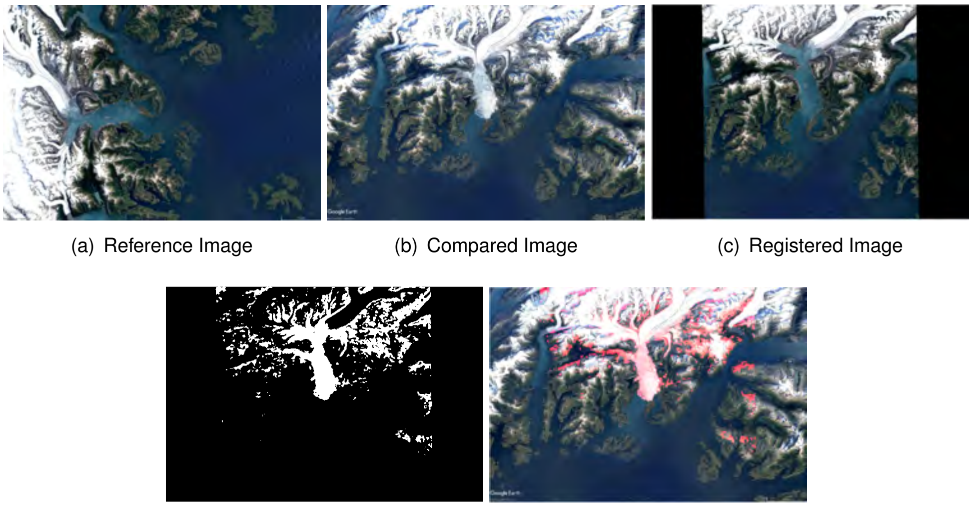
The next stage is the generation of two clusters by clustering the feature vector space using the k-means clustering algorithm. When there is a change between two images in a specific region, then it is expected that the values of the difference image pixels in that region are higher than the values of pixels in the regions where there is no change. Using this assumption, the cluster whose pixels have lower average value in the difference image is assigned as two classes (changed and unchanged). Finally we determine the classification and generate the difference image(Figure 6) by calculating the Euclidean distance[1].

A higher block size and the threshold are also chosen to filter the differential image results to eliminate most of the discrepant results due to differences in brightness and contrast.

The last step in this part is to visualize the changing regions in our application's GUI. We visualize the highlights of the detected change area on the calibrated image, and set a interactive slider that can change the threshold for users to freely try. The image changes can be visualized in real time on our designed GUI, and users can intuitively see the detected transformation area.

2.5 Change Information Analysis

In order to show the changes in satellite images more quantitatively, we plan to reflect the magnitude and speed of changes in the following metrics: change ratio, change rate, specific area, etc.



(a) Reference Image (b) Compared Image (c) Registered Image

(d) Differential Image (e) Highlight Change Information in Image

Figure 6: Change Information Detection

In addition, users can see the quantitative metrics of image changes in our application through the GUI.

2.5.1 Change Ratio

In order to compare the changes of the images as fair and reasonable as possible, we only select the overlapping part (the aligned area after rotation and translation) of the two images after registration as the new image. We call this area the registration area. In this part, the detected change is the change in the common area of the two images. In addition, We use the number of pixels to quantify the size of the above two areas.

Therefore, we can define a metric called the change ratio. Change ratio is the ratio of the changed area to the registered area. This can reflect the magnitude of change.

2.5.2 Change Speed

We have defined the rate of change to characterize the magnitude of the changes in the two images. Since we allow users to freely select images, the time interval between the two selected images may be completely different. That means that users cannot obtain relevant information about how fast the area shown by the satellite image changes over time.

Therefore, by adding the variable of time, we need to define a new metric called change Speed,which is the ratio of the change rate to the time period, and can quantify the speed of change.

2.5.3 Topological classification of the region

In addition, we can simply classify the image we choose, such as if it contains urban area or natural area. In this part, we use a function to detect the straight line[3][4][5] and color difference in the images. First, we convert the RGB image to HSV image and evaluate the size of its white/blue/green area. If the ratio of these three colors accounts for more than 70 percent of the overall image and there are not many straight lines, we consider it as a natural area, otherwise it is an urban area.

We do this because under normal circumstances, the very pure white areas are generally snow. Blue areas are related to water areas, such as oceans, lakes, and green areas are generally vegetation, etc. But occasionally there are water area with green (such as the dataset Kuwait), so we collectively refer to the white, blue and green sets as "natural factors".

2.6 GUI Development

The software interface we designed consists of three main sections:

- Show images: In this section, the comparison image can be normalized to the same brightness and contrast as the reference image, and all images from that location can be displayed in chronological order on the time lapse.
- Visualization: In this section, the following functions can be implemented: alignment of two images, production of the GIF for all images, display of differential images, highlight change areas.
- Quantitative indicators: In this section, we quantify the area of change and generate the analysis results, i.e.: time of change, speed of change, ratio of change, and topological classification of the region.

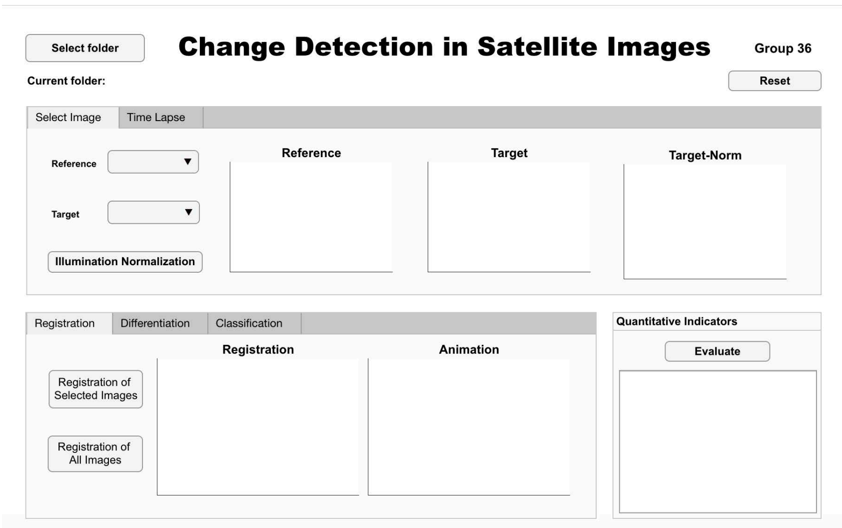


Figure 7: GUI interface

3 Results and Discussion

We selected four locations for testing and the results are as follows.

3.1 Location I - Wiesen

We set the image in 2015.07 as reference image and the image in 2018.04 as target image. The output image is as Figure 8. And the analysis results are as follows:

- Changing Time : 33 months
- Changing Ratio : 24.1714%
- Changing Speed : 0.7325% / month
- Area Classification : Urban

(a) Reference Image (b) Target Image (c) Normalized-Target Image (d) Registered Image

(e) Differential Image (f) Highlight Change Information (g) Highlight Nature Information

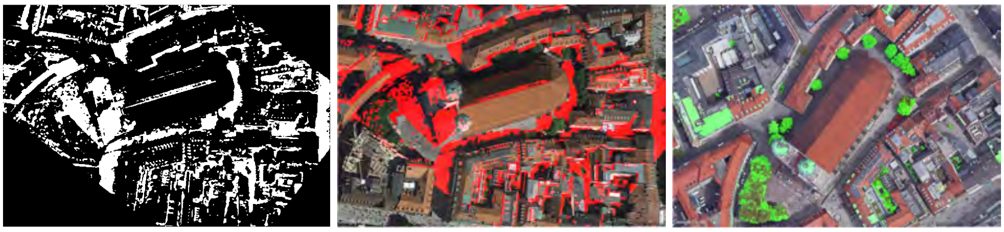
Figure 8: Location I - Wiesen

3.2 Location II - Frauenkirche

We set the image in 2012.08 as reference image and the image in 2018.04 as target image. The output image is as Figure 9. And the analysis results are as follows:

- Changing Time : 68 months
- Changing Ratio : 71.9349%
- Changing Speed : 1.0579% / month
- Area Classification : Urban

(a) Reference Image (b) Target Image (c) Normalized-Target Image (d) Registered Image



(e) Differential Image (f) Highlight Change Information (g) Highlight Nature Information

Figure 9: Location II - Frauenkirche

3.3 Location III - Columbia Glacier

We set the image in 2000.12 as reference image and the image in 2020.12 as target image. The output image is as Figure 10. And the analysis results are as follows:

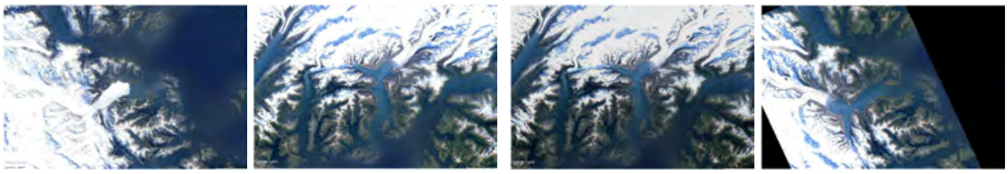
- Changing Time : 240 months
- Changing Ratio : 11.7237%
- Changing Speed : 0.0488% / month
- Area Classification : Nature

3.4 Location IV - Dubai

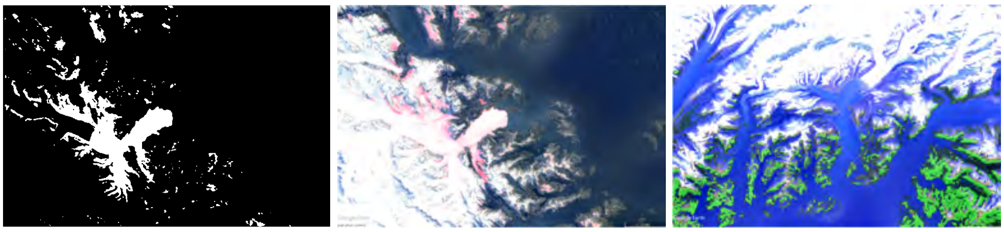
We set the image in 1990.12 as reference image and the image in 2003.12 as target image. The output image is as Figure 11. And the analysis results are as follows:

- Changing Time : 300 months
- Changing Ratio : 1.6363 %
- Changing Speed : 0.0055 %/ month
- Area Classification : Nature

What is interesting in this dataset Dubai is that if the image 2015.12 is set as the target image, the result of "This area is urban" will be obtained, which reflects the development of the area shown in the satellite image.

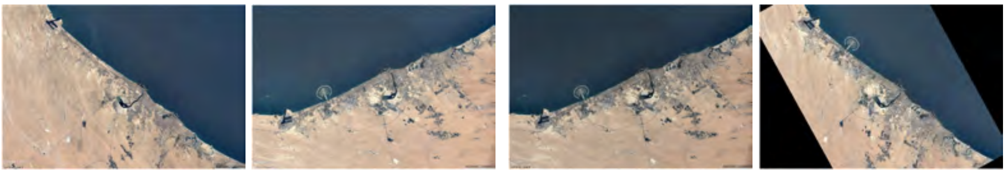


(a) Reference Image (b) Target Image (c) Normalized-Target Image (d) Registered Image



(e) Differential Image (f) Highlight Change Information (g) Highlight Nature Information

Figure 10: Location III - Columbia Glacier



(a) Reference Image (b) Target Image (c) Normalized-Target Image (d) Registered Image

(e) Differential Image (f) Highlight Change Information (g) Highlight Nature Information

Figure 11: Location IV - Dubai

4 Conclusion and Critical Reflection

According to the above statement of the project structure and content and the display of the final result, our code has generally achieved the following parts relatively well:

- Good visualization quality: outputs conclude reference image, target image, normalized-target image, registered image, differential image, image with highlight change information, image with highlight nature area, gif for all images in dataset, images in time lapse, etc.

- Complex Scenes and Different Locations: we can achieve good performance in images registration and change information detection in different scenes and locations.
- Run-time optimization: We have optimized our algorithm to run as fast as possible while ensuring accuracy
- Well-designed GUI: we have designed a feature-rich and interactive APP interface (MATLAB APP Designer)

Based on the above conclusions and reflections, we propose some improvements that can be used as references for future work.

- More sort of areas in images classification
- Reduced running time
- More accurate of region classification
- More accurate matching of some complex images

References

[1] T. Celik, "Unsupervised Change Detection in Satellite Images Using Principal Component Analysis and k-Means Clustering," in IEEE Geoscience and Remote Sensing Letters, vol. 6, no. 4, pp. 772-776, Oct. 2009.

[2] Pablo Fernandez Alcantarilla, Adrien Bartoli, and Andrew J. Davison, "KAZE Features" A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 214–227, 2012. c Springer-Verlag Berlin Heidelberg 2012.

[3] P. Kahn, L. Kitchen and E.M. Riseman, A fast line finder for vision-guided robot navigation, PAMI nov. 1990, pp 1098-1102.

[4] X. Lebegue and J.K. Aggarwal, Significant line segments for an indoor mobile robot. PAMI, Dec. 1993, pp 801-815.

[5] Jana Kosecka and Wei Zhang, Video Compass, IJCV 2002.