" Satler - A Satellite Image Stitching Webpage "

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

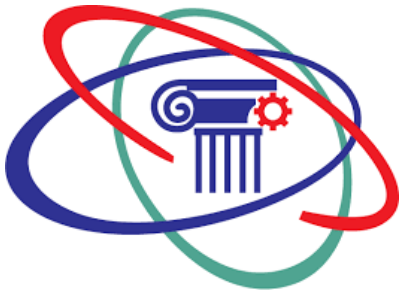
Bachelor of Technology in *Computer Science & Engineering*

Submitted by:

Aayushi Jain *0827CS201007*
Alokit Sharma *0827CS201023*
Aman Solanki *0827CS201027*

Guided by:

Mr. Narendra Pal Singh Rathore
Mrs. Priyanka Jangde



Acropolis Institute of Technology & Research, Indore

July - December 2022

EXAMINER APPROVAL

The Project entitled ***" Stitching Satellite Images "*** submitted by **Aayushi Jain (0827CS201007), Alokit Sharma (0827CS201023), Aman Solanki (0827CS201027)** has been examined and is hereby approved towards partial fulfillment for the award of Bachelor of Engineering degree in Computer Science & Engineering discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled **" *Stitching Satellite Images"*** submitted by **Aayushi Jain (0827CS201007), Alokit Sharma (0827CS201023), Aman Solanki (0827CS201027)** is a satisfactory account of the bonafide work done under the supervision of **Prof. Priyanka Jangde and Prof. Narendra Pal Singh** are recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science & Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that project entitled **“** ***Stitching Satellite Images*** **”** has been developed by us under the supervision of Prof. Priyanka Jangde and Prof. Narendra Pal Singh Rathore. The whole responsibility of work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

Aayushi Jain (0827CS201007)
Alokit Sharma (0827CS201023)
Aman Solanki (0827CS201027)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are a number of people without whom this project's work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentors **Prof. Priyanka Jangde and Prof. Narendra Pal Singh Rathore**, Associate Professor, AITR, for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing us all necessary facilities for the project to achieve our objectives.

We are grateful to **our parents** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say, “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Aayushi Jain (0827CS201007) Alokit Sharma (0827CS201023) Aman Solanki (0827CS201027)

Executive Summary

" Stitching Satellite Images "

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Computer Science & Engineering branch under the sagacious guidance and vigilant supervision of ***Prof. Priyanka Jangde and Prof. Narendra Pal Singh Rathore***.

Our project involves the making of a program which includes taking satellite images from the user and merging them together seamlessly to get a mosaic (or continuously merged image). This program will enable us to merge two images together only and only if the images overlap each other.

Key words : Image Processing, Neural Networks, Tensorflow

The whole purpose of education is to turn mirrors into windows.

-Sydney J. Harris

List of Tables

List of Figures

Table of Contents

CHAPTER 1 : INTRODUCTION

Introduction

The process in which images are integrated and combined in the form of single images is called Image Stitching. In Image Stitching common areas or overlapping of two or more images are used in domains like viewing critical areas,to identify the high resolution images into digital form,medical images and help in application related to 3D alliteration that's used in real world problems.counting those objects by detecting and then recognizing them in real-time.

1.1 Overview

The use of digital images is now increasing rapidly in many fields, and the use of digital images is widespread, such as solving large problems that occur exceptionally in all fields such as life, medicine, agriculture, industry, and the Internet. considered to be prevalent.

Information Sources on Technological Advances. One of the most important applications is image merging, also called image mosaic. Image stitching means grouping like-meaning images into a high-resolution, wide panoramic image of the overlapping regions. In recent years, modifications and further developments in the algorithms used in this field have made him an important area of image processing. Stitching has many uses, such as in maps, satellites, knowledge and positioning, etc. This summary article therefore presents a series of image stitching techniques and their use in terms of evidence and accuracy for each, along with a comparative study of several research papers in this field over a period of years (2017-2020). provide the law. Therefore, this article will help researchers in this field to use and develop Stitching algorithms to discover features and match them to create convenient, hassle-free, high-resolution panoramic images.

1.2 Background and Motivation

In this study, images are combined on the basis of the coordinates that intersect each other most frequently. The developed algorithm works on a single node. This algorithm was developed according to the Map/Reduce framework using the Python programming language. The application architecture consists of two mapper functions running in parallel. The output of the 1.mapper function is used as input to the 2.mapper function. The 1.map function is for creating a new space for an image, while the 2.mapper function is for computing the common intersection between images, the two image combining processes are most often computed. Reference the intersection coordinates.

1.3 Problem Statement and Objectives

In our project, there are multiple satellite images which we are merging using many algorithms and tools. There are multiple ways to merge satellite images

Thus, tools and sources which are used:

Objective : To identify tool names that have been used in satellite merging images and details about sources related to it:

Tool name	**Source**
PTGui	**close**
Hugin	**open**
Panoweaver 10	**close**
AutStitch	**close**
Panorama Studio	**open**
Panorama Stitcher	**Only for mac(close)**
PhotoStitcher	**open**
GigaPan Stitcher	**open**

PTGui is open source which originally started in Graphical User Interface for Panorama Tools in an industry leading photostitcher application.

Hugin is a close project that makes reasonable efforts to ensure that the released source code builds on a wide variety of platforms. We also provide information on how to build the source

code . We don't have the resources to go beyond that. Sometimes we also don't have access to specific platforms and can't guarantee that the code builds either.

Panoweaver 10 is a highly-intelligent and automatic panorama stitcher. The operation is very simple and friendly. You can create a perfect panorama with one click. It’s easy to remove the tripod and synchronize to EP-Sky, Facebook, Twitter, Google+ and etc., and can be recognized as panoramic viewing.

AutoStitch takes a step forward in panoramic image stitching by automatically recognising matching images. This allows entire panoramas to be constructed with no user input whatsoever. AutoStitch is incredibly simple to use! Just select a set of photos, and AutoStitch does the rest: digital photos in, panoramas out.

PanoramaStudio. Creation of seamless 360 degree and wide angle panoramic images.
This program combines simple creation of perfect panoramic images within a few steps with ambitious post processing features for advanced users.

Photoscticher Ensures that each photo has about 30-50% overlap with all other adjacent photographs.

1.4 Scope of the Project

The goal of this problem statement is to develop a fully automated algorithm to automatically stitch adjacent (multiple) satellite images without human intervention to create a seamless mosaic. The algorithm needs to be fast enough for on-the-fly stitching for web based visualization.

CHAPTER 2 : REVIEW OF LITERATURE

Review of Literature

Feature-based techniques aim to determine relationships between images based on the extraction of various features. The keys of some cost minutiae in the image pair are compared to the entire features on the multi image using one of their local descriptors. Image stitching properties focused on feature-based methods are:

Author/ Application	Parallax	Principal	Advantage	Disadvantage
Autostitch	Simple Sparse feature matching.		Auto method	Restricted to one plane.
Gao et al.	Dual-homograph		Two planes	one tomography
Lin et al.	Multi-affine	✔	determine small parallax	one affine.
Zaragoza et al.	Mesh-based		Multiple transformations	Local distortion
Liu and Chin	Insert more appropriate correspondence based on pixels.	✔	determine the images that have a specific degree of parallax, and wide-baseline.	Complex calculation.
Chang et al.	Local hybrid transformation model.		Various transformations between overlapping and non-overlapping regions.	Restricted to vertical or parallel images.
Li et al.	Quasi tomography		Reduce distortion	Limited to parallel images.
Zhang and Liu.	Hypothesis of warping	✔	Effective and robust stitching.	Not reusable knowledge and alignment
Lin et al.	Seam-driven	✔	Optimization seam, reduce ghosting	Iterative operations and complicated calculation
Herrmann et al.	Prior constraints		Single registration, and determine the images with	Limited to images contain

			moving objects.	
Chen and Chuang	**Coarse-to-fine-scheme**	✔	**Scale and rotation correction.**	**Local distortion.**
Zhang et al.	**Various prior constraints and optimization terms.**		**Address the wide-baseline images.**	**The complicated calculation, local distortion.**

The first application from the table, Autostitch, was based on Simple Sparse feature matching.It included the Auto method but was only restricted to one plane.

The next person Gao et al. focusing on Dual-homograph was for two planes but it had only one tomography.

Furthermore Lin et al. used this principle of Multi-affine and was able to determine small parallax. But it also did not have one affine.

Then came Zaragoza et al. with his Mesh-based system and was able to find out multiple transformations.

Liu and Chin, Inserted some correspondence based on pixels but it had complex calculations.

Chang et al. Local hybrid with his local hybrid transformation model was able to figure out various transformations between overlapping and non-overlapping regions.

Li et al. with Quasi tomography was able to reduce distortion but got limited to parallel images.

Zhang and Liu. with the Hypothesis of warping that was Effective and robust stitching.

Lin et al. with his Seam-driven approach being able to reduce ghosting but had iterative operations and created complex calculations.

Chen and Chuang had a Coarse-to-fine-scheme that was scale and rotation correction but still had local distortion.

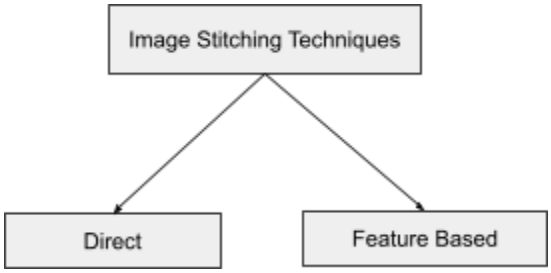
CHAPTER 3 : PROPOSED SYSTEM

Proposed System

3.1 The Proposal

- ➔ Higher reliability for each type of scene displayed in the image.
- ➔ This method is very fast and can recognize panoramas by automatically recognizing relationships between adjacent image sets.
- ➔ These functions are suitable for fully automatic panorama stitching.
- ➔ Feature-based methods rely on accurate detection of image features.

The feature-based techniques and direct are the two general methods that were used to characterize techniques used for Image stitching.



Type of Image stitching techniques

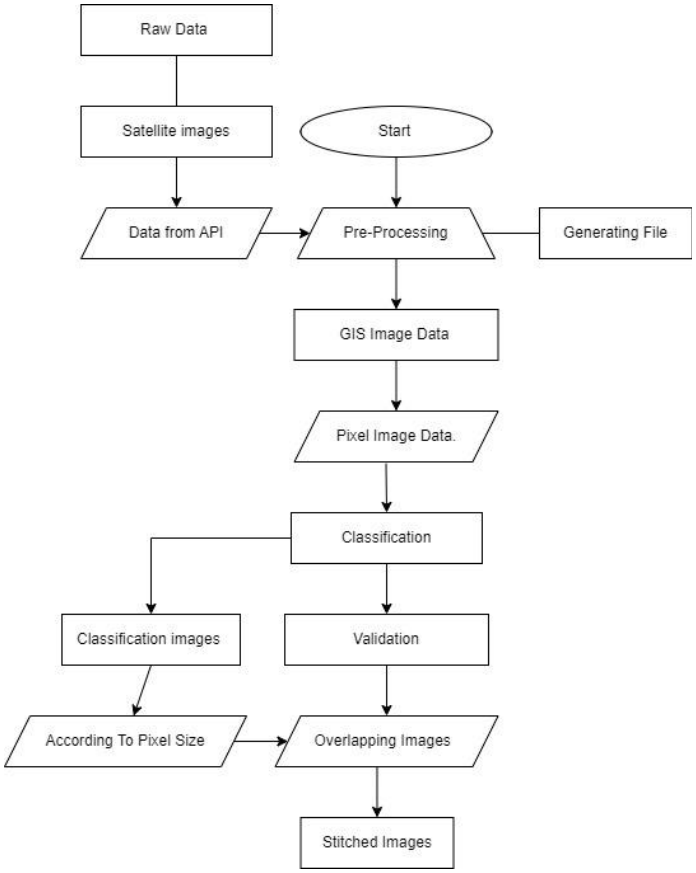
3.2 Direct Technique:

Direct Technique depends on the comparison of pixel intensity. It means how area coverage, overlapping of pixel intensity of the image is compared by another image this is known as direct technique

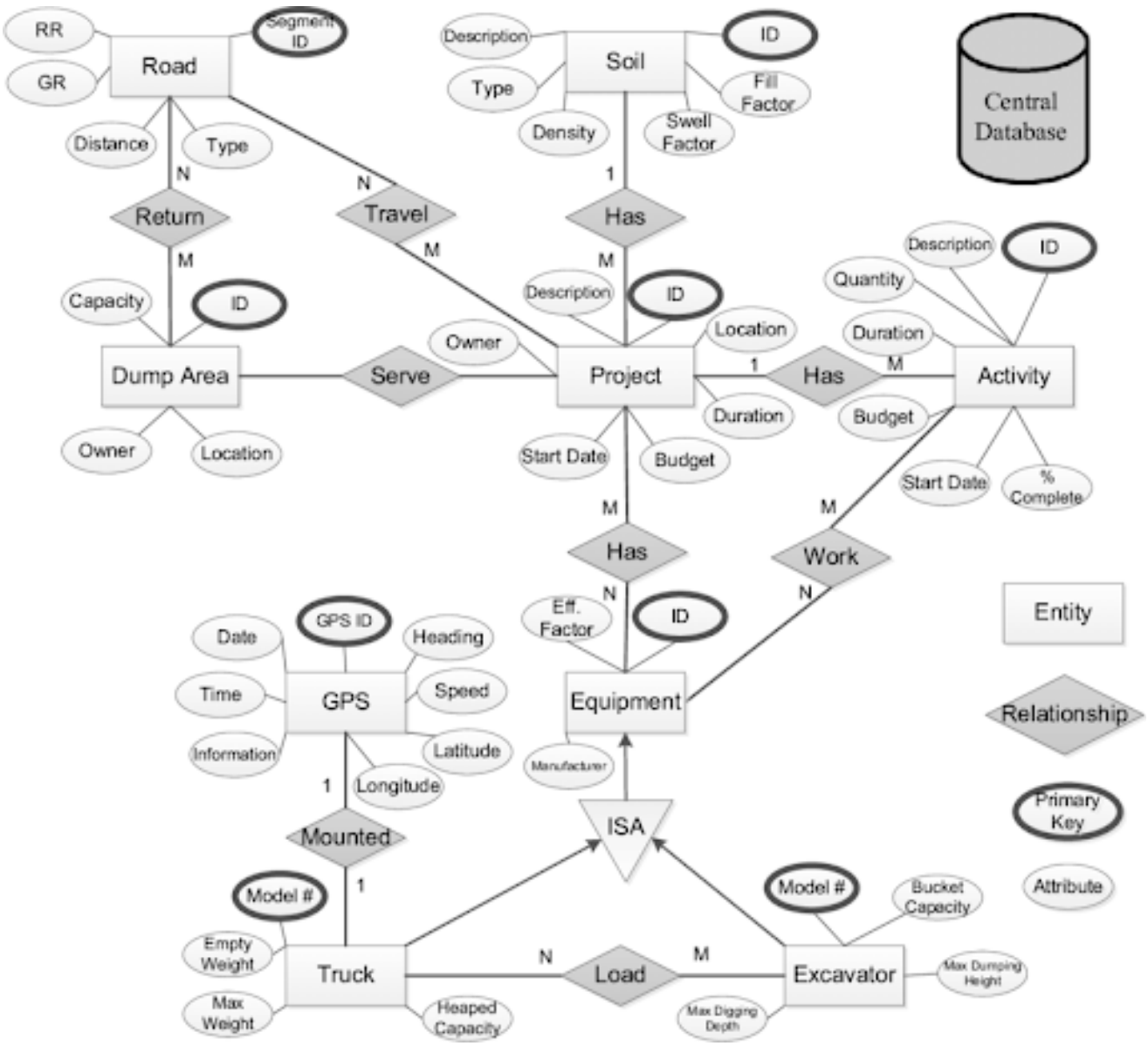
3.2.1 Advantage of Direct Technique

1. Minimum use of pixels in overlapping photos.
2. Best use of image content and in details.
3. In this technique input of each pixel into the image can be evolved.
4. Images should overlap up to 30 percent to 50 percent.

3.3 Diagrams



Flow Chart



Use Case Diagram

CHAPTER 4 : IMPLEMENTATION

4.1 Conversion Image to Bitmap and String Format

In computers, images are represented in the form of 8-bit matrices. Each matrix element, called a pixel, has a value ranging from 0 to 255, representing various pixel intensities. A threshold function can be used to convert an image into a bitmap where each pixel can have only two values, 0 or 255. All pixel intensities below the specified threshold are considered 0 and all pixels above the threshold are considered 255. The image is converted to binary format. , black pixels are '0' and white pixels are '255'.

image1 in a bitmap:

```
from IPython.display import Image
Image('/content/image_1_in_bitmap.png')
```

255	255	255	255	255
255	255	255	255	255
0	0	0	255	0
255	255	0	0	0
255	255	255	255	255

image2 in a bitmap:

```
from IPython.display import Image
Image('/content/image_2_in_bitmap.png')
```

255	255	255	255	255
255	255	255	255	255
0	0	255	255	255
0	0	255	255	255
0	0	0	0	0

Since the Mapper function accepts input stored in files, the bitmap images are required to be converted to String format, to supply input as text. String format of images are defined as each pixel column separated by a comma ’,’ character and each pixel row is separated by a semicolon ‘;’ character.

String format representation of image1:

“255,255,255,255,255;255,255,255,255,255;0,0,0,255,0;255,255,0,0,0;255,255,255,255,255"

String format representation of image2:

“255,255,255,255,255;255,255,255,255,255;0,0,255,255,255;0,0,255,255,255;0,0,0,0,0”

4.2 Creating new space for images:

The images are defined in a new space, so that all possible matching cases between the two images can be calculated. This is the first mapper function.

Input: Image1_Str Image2_Str
Output: Image1_Str Image2_New_Space

For each image pair, the width and height of image1 is added around the image2 as white pixels. Since image1 and image2 are 5X5 size images, thus the width of image1 (5 pixels) and height of image1 (5 pixels) is added to the width and height around image2, resulting in Image2_New_Space having a 13x13 size.

Image2_New_Space in a bitmap:

```
from IPython.display import Image
Image('/content/image_2_new_space.png')
```

255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	0	0	255	255	255	255	255	255	255
255	255	255	255	0	0	255	255	255	255	255	255	255
255	255	255	255	0	0	0	0	0	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255

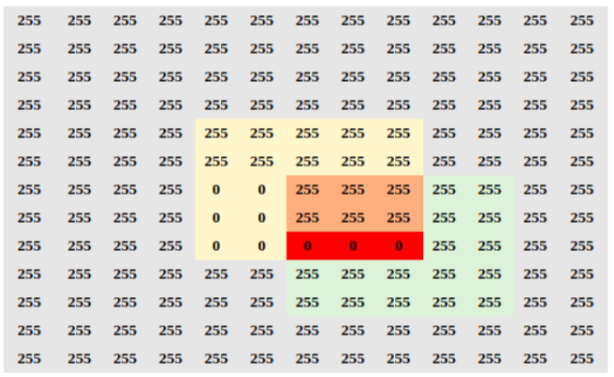
4.3 Calculation of Maximum Number of Black Pixel Overlaps Between The Images And The Largest Intersection Coordinate for Combining The Images:

The Image2_New_Space is defined such that it can test for all possible combinations of image1 and image2. The image1 matrix is thus traversed over the image2 matrix and the overlapping black pixels are counted. The elements of image1 are considered as a floating frame on image2(in new space). Black pixels in common indices are counted where one-to-one matching occurs between the first image and the second image.

As shown in the above figure, the elements of the image1 matrix are matched with 5X5 matrices each marked with a different color in the second image matrix. This operation is performed for all 5X5 matrices defined in the Image2_New_Space matrix and common black pixels are counted for each pairing state.

The output of the reduce function gives us the maximum number of black pixel overlaps. The maximum number of lack pixel overlaps are found to be 3. Using this information, the largest intersection coordinate is obtained. For the Image2_New_Space, the largest intersection coordinate with respect to image1 is calculated as (6,6)

```
from IPython.display import Image
Image('/content/overlapping_coordinates.png')
```



4.4 Merging

Using the obtained coordinate from the reduce functions, a merge operation is used to obtain the stitched image. The size of the merged image is defined as (3a-2) x (3b-2), where the size of the images is (a x b).

```
from IPython.display import Image
Image('/content/merged_image.png')
```

4.5 Calculation & Algorithm

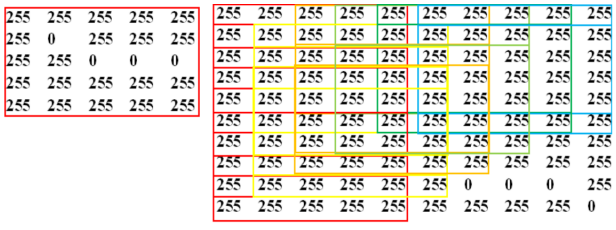
The first matrix is traversed over the second matrix using a block size generated by referencing the size of the first matrix, counting duplicate black pixels. Elements in Image1 are displayed as floating frames in Image2 (in the new space). If there is a one-to-one match between the first image and the second image, the black pixels at the common index are counted. FIG. 2 shows the state of the floating window. In Figure 3, the elements of the first image matrix match his 5x5 matrix, and each is marked with a different color in the second image matrix. This operation is performed for every 5x5 matrix defined in Image2 and common black pixels are counted for each pairing state. Save the intersection point coordinates, which is the maximum value of the obtained intersection number, and use it in the merging process in the next step. The formula for calculating the maximum number of matches between images is shown in equation (1).

$$\text{Max number of intersections} = \sum_{row=0}^{n+a-1} \sum_{col=0}^{m+b-1} select\ Max \left(\sum_{i=0}^{n} \sum_{i=0}^{m} 1 image1_{ij} = image2_{i+row, j+col} = 0 \right) \quad (1)$$

The operation is performed in the 2.map function defined in the algorithm. The input of the map function is defined as the output of the 1.map function.

Input: Image1 (string format) New space Image2 (string format)

Output: New image defined in string format by combination of image1 and image2



Algorithm1: A mapper function that computes the largest intersection between two images and performs two-image merging

```
1    class MergeImages
2      method Map(String Images)
3        for row = 0 to (n + a - 1)
4          for col = 0 to (m + b - 1)
5            countMatchings ← 0
6            for i = 0 to n - 1
7              for j = 0 to m - 1
8                if image1i,j = image2row+i,col+j = 0
9                  countMatchings ← countMatchings + 1
10
11       Select max match count and coordinate(x,y)
12
13       for i = 0 to n
14        for j = 0 to m
15         image2i+x, j+y ← image1i,j
16       emit(image2)
```

4.6 Tools Used

4.6.1 OpenCV

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand users and an estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mine inspection, stitching maps on the web or through advanced robotics.

4.2.2 Libraries Used

- ➔ Open CV
- ➔ Numpy
- ➔ Matplot
- ➔ Imageio

4.3 Language Used

Python language is used in the system due to the following Characteristics :

Simple :

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudocode nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source :

Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and know that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented :

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

Extensive Libraries :

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python.

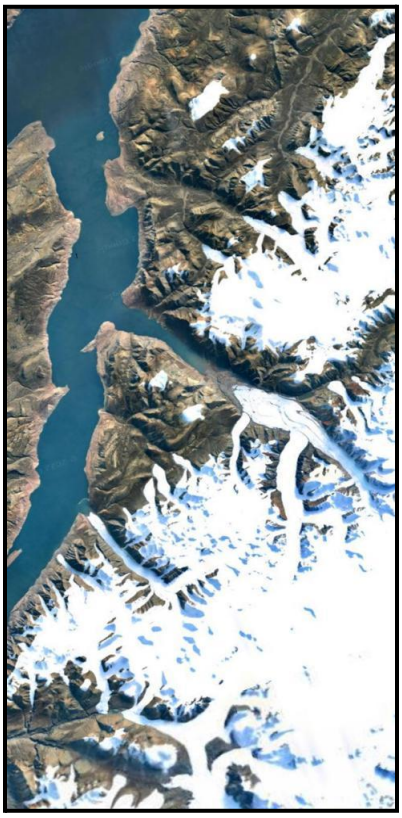
4.4 Result

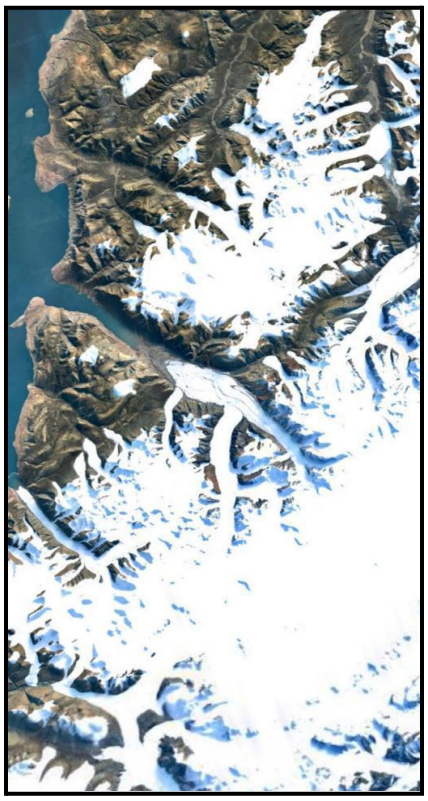
Usually the algorithm currently only works and focuses on two images. The result of the algorithm is: Show the database image and the image to be added, respectively. This image was found on Google Earth.

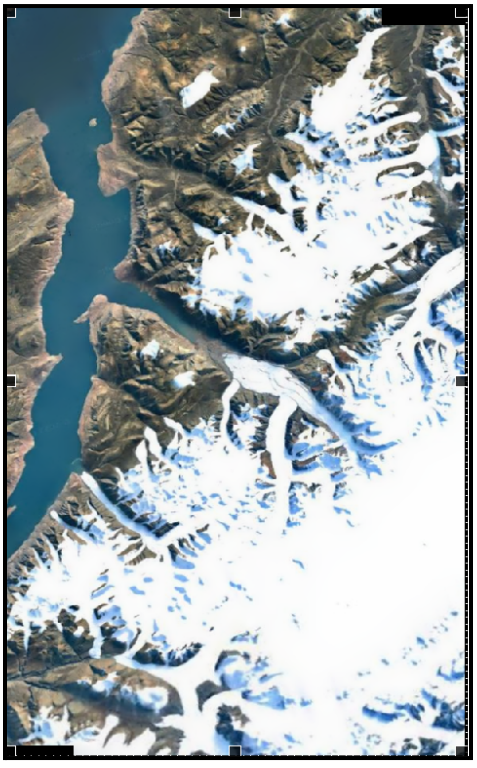
Image 1

Image 2

Result







CHAPTER 5 : CONCLUSION

Conclusion

5.1 Conclusion

Image stitching is still a hot spot in many image processing, computer vision, computer graphics, software development, and other journals. Several challenges face image stitching still unsolved so solved by using

Process images according to Hadoop Map/Reduce framework. It is intended to test the performance of a map function that will work by providing an image as input to the map function presented in text format. It reads images through the Hadoop file system and aims to create a scalable system for large images using a map/reduce programming approach. Based on the point where the two images are most similar, we tried to solve the fusion process of complexity n4.One of the improvement points of the developed algorithm is to use an algorithm that will determine the threshold value with the black-and-white pixel balance of the images as they are converted to the bl pixel.The conversion of images to according to the threshold value of "128" appears to be a challenge to find matches and a reduction in the accuracy of finding matches on the actual images with a high white density or black density.. The algorithm developed is based on the counting of pixels. It is a success when the black color intensity of the pictures is low. But when the white density of the image is too high, it is difficult to catch the intersections. The next step is to add a preprocessing step that subtracts the overall color intensity of the images and the intersection will be able to positively affect the performance of the calculated application with reference to this step. A preprocessing step that detects edges in an image can be effective in obtaining fast and accurate results when calculating the image-to-image match.

5.2 Limitations

- ➔ Limited coverage range in images.
- ➔ Scale and rotation of images does not change.
- ➔ The Technique is very complicated to fetch images.
- ➔ Only images that overlap each other can be merged right now.

REFERENCES

Languages used : HTML,CSS,Python

Application used: Photostitcher

https://www.researchgate.net/publication/321140605_An_Approach_For_Stitching_Satellite_Images_In_A_Bigdata_Mapreduce_Framework

https://iopscience.iop.org/article/10.1088/1742-6596/1999/1/012115/pdf

https://github.com/shubhi/satellite-image-stitching

https://www.researchgate.net/publication/342322385_Image_Processing_Techniques_for_Analysis_of_Satellite_Images_for_Historical_Maps_Classification-An_Overview

https://ieeexplore.ieee.org/document/9670954

https://towardsdatascience.com/how-to-download-high-resolution-satellite-data-for-anywhere-on-earth-5e6dddee2803