Ukrainian Catholic University

Faculty of Applied Sciences

Business Analytics & Computer Science Programmes

Linear Algebra
2nd interim report

Panorama Stitching

Team:
Daryna Kuzyshyn
Olha Havryliuk
Anastasiia Dynia

3rd of April 2024

1. General overview of the problem

In the field of computer vision, one area that stands out for its practical application and visual appeal is panorama stitching. The process involves taking multiple images from different angles and smoothly merging them into a single panoramic image. This technique allows for a broader view that is much wider than what can be captured by regular photography methods, offering a comprehensive perspective of landscapes, cityscapes, or any other scenery.

Even though many of today's digital devices can stitch panoramas, creating a perfect panoramic image still remains an issue because of technical challenges. The primary issues come from the need to accurately align and blend overlapping images without visible seams, distortions, or exposure differences. Each image in the series can vary in terms of angle, scale, lighting, and perspective, all these things make the stitching process complex. Additionally, the technical process of finding exact points that match up in different images, figuring out the best way to line them up, and seamlessly combining them into one picture without any flaws requires a strong and reliable method.

Our project aim is to develop a panorama stitching algorithm that addresses these challenges by using advanced techniques in Computer Vision and Linear Algebra. We want not just to develop the stitching process algorithm but to enhance the quality of panoramic photography.

2. A review of related work and possible approaches to solutions

Image panorama techniques are classified into two categories:

- Direct techniques
- Feature-based techniques

Let's take a closer look at each of them:

Direct Techniques:

The direct technique is dependent on the comparison of pixel intensities of the images and it decreases total differences among overlapping pixels. Each pixel intensity is contrasted with every other pixel intensity, making it an extremely complex method.

Feature-based techniques:

Feature-based techniques are aimed at determining a relationship between the images based on extracting different features. Feature-based methods are used to match different shapes between images such as points, lines, etc. The main steps for image stitching are **feature detection and description, feature matching and homography estimation**.

Step 1. Feature detection and description

Feature detection involves identifying specific points, regions, or structures in an image that are significant and can be used as references for further analysis. Once relevant features are detected, the next step is to describe these features in a way that allows for efficient matching and recognition. The goal is to create a representation of the features that captures their distinctive characteristics while being resistant to variations that might occur in real-world images. There are such techniques that allows us to detect and describe necessary features:

- *Harris Corner Detector.* Harris algorithm is a point extract algorithm to get all the corners in the image. The Harris detector looks at the average directional strength.
- *SURF Detector.* This algorithm is used based on multi-dimensional space theory and acceleration calculations using fast matrix approximation and Hessian's definition of "integrated images".
- *ORB Detector.* The ORB is a fast binary descriptor based on BRIEF key points and FAST detectors. BRIEF is a new feature descriptor that uses a smooth image patch binary test between pixels.
- *SIFT Detector.* SIFT calculates the local image descriptor of each key point based on the value of the image gradient and the direction of each point of the image sample in the area centered on the key point.

Step 2. Feature matching

After extracting features for pair of an image, the next step is to match the feature extracted. The matching process uses descriptor data to compare matching points. This step is intended to compare the best features of an image with the other image. Then if the features of the input images match, to achieve accurate feature matching, the locations of the identical features will be identified as matched pairs. Possible methods for feature matching are provided below:

- *Brute-Force Matcher.* The matches are evaluated on the basis of Euclidean Distance between the two key points. Euclidean distance of every keypoint in the first image is calculated with every other keypoint in the second image. The good matches are then separated by certain minimum distance criteria.
- *KNN matcher.* After identifying features in each image, the KNN matcher considers each feature point from one image and searches for the k closest feature points in the other image, based on a defined distance metric, for example, the Euclidean distance.

Step 3. Homography estimation

Homography estimation is a technique used to find the relationship between two images of the same scene, but captured from different viewpoints. It is used to align images, correct for perspective distortions, or perform image stitching. In order to transfer similar points from one image to another, we use 3x3 homography transformation matrix, which shows the correspondence between two images' corresponding points' pixel coordinates. Geometric changes including translation, rotation, scaling, and perspective distortion are possible using homography. There are such techniques that can be used to perform homography estimation:

- *RANSAC.* This algorithm estimates the homography matrix by picking a subset of the matched keypoints iteratively. The optimal homography matrix with the most inliers is chosen after assessing the number of inliers that suit the predicted model.

- *Hough transforms.* The Hough transform is a feature extraction technique that converts an image from Cartesian to polar coordinates. Any point within the image space is represented by a sinusoidal curve in the Hough space. In addition, two points in a line segment generate two curves, which are overlaid at a location that corresponds with a line through the image space.

3. A brief explanation of the algorithm chosen and its pros and cons

We have chosen the feature-based approach for our project, because it has such advantages:
• It can accurately match features across images that are taken from different angles or positions
• The features can be matched even if the images have been captured from different distances or orientations
• It helps reduce the impact of noise in the resulting picture

As it was already mentioned, there are **3 key steps for image stitching.**

- For feature detection and description **SIFT** was chosen. It is more accurate than any other descriptors and is rotation and scale invariant. But one of the drawbacks is that it is mathematically complicated and computationally heavy.

- For feature matching **KNN** is used. The pros of this method are that it is really simple to implement and it has relatively high accuracy. But, on the other hand, sometimes it can be computationally expensive and it has a high sensitivity to the choice of k and the distance metric.

- For homography estimation **the RANSAC** will be implemented. It is highly robust to outliers and is conceptually simple and straightforward to implement. But the cons are that sometimes too many iterations are required and this algorithm is highly sensitive to the threshold parameter (ϵ) that determines which points are considered inliers.

4. The theoretical part behind the algorithms

Here's a closer look at the theory and some of the formulas behind each of the stages of panorama stitching implementation:

4.1. Feature detection and description

In general, Scale-invariant feature transform(SIFT) algorithm can be decomposed into four steps:

- **Feature point (also called keypoint) detection:**

The first stage of SIFT is to identify keypoint candidates that are invariant to scale and orientation. To achieve scale invariance, the algorithm starts by constructing a scale space, which is typically done using the Gaussian function. The scale space of an image I is defined as a function $L(x, y, \sigma)$ that is produced by convolving $I(x, y)$ with a Gaussian kernel $G(x, y, \sigma)$ at different scales σ:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

SIFT uses the Difference of Gaussians (DoG) function to efficiently approximate the scale-normalized Laplacian of Gaussian (LoG), $\nabla_\sigma^2 G$, which is computationally more expensive. The DoG is obtained by subtracting two nearby scales:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3}$$

Extrema in the DoG images are identified as potential keypoints. These are points that are either maxima or minima of $D(x, y, \sigma)$ compared to their neighbors in both the same and adjacent scales.

- **Feature point localization:**

The second stage in the SIFT algorithm refines the location of these feature points to sub-pixel accuracy whilst simultaneously removing any poor features.

Consider D, which is the result of DoG method:

$$D(\mathbf{z_0} + \mathbf{z}) \approx D(\mathbf{z_0}) + \left(\left. \frac{\partial D}{\partial \mathbf{z}} \right|_{\mathbf{z_0}} \right)^{\mathrm{T}} \mathbf{z} + \frac{1}{2} \mathbf{z}^{\mathrm{T}} \left(\left. \frac{\partial^2 D}{\partial \mathbf{z}^2} \right|_{\mathbf{z_0}} \right) \mathbf{z},$$

where the derivatives are evaluated at the proposed point $\mathbf{z_0} = [x_0, y_0, \sigma_0]^T$ and $\mathbf{z} = [\delta_x, \delta_y, \delta_\sigma]^T$ is the offset from this point.

The Hessian matrix, used to reject features with low contrast or those located on edges, is defined as:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

Then we compute the ratio of the eigenvalues λ_1 and λ_2:

$$\frac{\mathrm{Tr}^2(\mathbf{H})}{\mathrm{Det}(\mathbf{H})} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} < \frac{(r+1)^2}{r},$$

where

$$r \text{ is a threshold ratio,}$$
$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy},$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx} D_{yy} - (D_{xy})^2.$$

Points with a high ratio of eigenvalues are identified as edge points and are excluded from consideration.

The last steps are **orientation assignment and keypoint description**.

4.2. Feature matching

Once features are extracted from each image, the next step is to find matching features in different images. For this we use KNN-algorithm. For each feature in one image, the algorithm searches for the closest k features in the other images based on a distance metric, for example Euclidean distance.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

The distance metric quantifies how similar two features are, with a smaller distance indicating a higher similarity.

4.3. Homography estimation

A homography is a 3×3 non-singular matrix that relates points in one plane to points in another plane, under projective transformations.

Homography is represented by the matrix **H**:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix},$$

where h_{ij} denotes the element in the i-th row and j-th column of **H**.

Lets assume $\mathbf{x} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$ and $\mathbf{x}' = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ in homogeneous coordinates.

The relationship between the two points under the homography transformation is given by:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = c \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

By eliminating the scale factor c, the relationship can be written in a linear form $A\mathbf{h} = \mathbf{0}$, where A is a 2×9 matrix constructed from the coordinates of the corresponding points, and **h** is a 9×1 column vector containing the flattened **H** matrix:

Homography has 8 degrees of freedom even though it contains 9 elements (3x3 matrix) i.e. the number of unknowns that need to be solved for is 8.

Computing a homography using **the RANSAC algorithm** involves mapping points from one plane to another with a perspective transformation. The RANSAC determines the complete homography matrix based on a threshold value.

Given a set of point correspondences between two images, we denote the points in the source image as $\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ and the corresponding points in the destination image as $\mathbf{x}'_i = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix}$:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \approx \mathbf{H} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

RANSAC iterates, each time selecting a new subset of points, estimating a new **H**. The best **H** is the one that results in the largest number of inliers.

5. The implementation pipeline

1. Literature Review (February 28)
2. Gathering necessary datasets for future algorithm testing (March 2)
3. Writing first interim report (March 6)
4. Learning the algorithms that will be used in the project (March 15)
5. Implementation of the main algorithms (March 30)
6. **Writing second interim report (Current step, April 3)**
7. Finishing the coding part and testing the implementation (April 20)
8. Preparing for the project presentation, writing and submitting the final report (April 24)
9. Presenting the work (April 24)

6. References

General overview of panorama stitching process:

- https://www.researchgate.net/publication/354746805_A_Review_Over_Panoramic_Image_Stitching_Techniques
- https://medium.com/@paulsonpremsingh7/image-stitching-using-opencv-a-step-by-step-
- https://www.researchgate.net/publication/348232877_A_Brief_Review_on_Image_Stitching_and_Panorama_Creation_Methods

SIFT:

- https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

KNN:

- https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorit

Homography:

- https://www.researchgate.net/publication/319662477_Unsupervised_Deep_Homography_A_Fast_and_Robust_Homography_Estimation_Model