Image Stiching Based on Greedy and Kuhn-Munkres Algorithm

Mochammad Fariz Rifqi Rizqulloh - 13523069
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: faris361707@gmail.com , 13523069@std.stei.itb.ac.id

***Abstract*— Abstract—Image stitching is the process of merging multiple overlapping images into a unified panoramic or high-resolution composite. A critical stage in this process is feature matching, which significantly impacts the quality and accuracy of the final stitched result. This study investigates and compares three feature matching strategies: a commonly used FLANN-based matcher, a Greedy matcher, and the Kuhn-Munkres (Hungarian) algorithm. The FLANN matcher offers speed and efficiency through approximate nearest neighbor search, while the Greedy matcher provides faster performance at the cost of global consistency. The Kuhn-Munkres algorithm, although computationally more expensive, delivers globally optimal one-to-one matches. Experiments on various image configurations demonstrate that while the Greedy method is fastest, it may produce suboptimal matches. The Kuhn-Munkres algorithm, albeit slower, yields the most accurate and consistent results, making it suitable for applications where precision is critical. This paper highlights the trade-offs between speed and accuracy in feature matching, offering insights into selecting the appropriate algorithm based on application requirements.**

Keywords—Image Stitching, Feature Matching, Greedy Algorithm, Kuhn-Munkres, Computer Vision, Homography

I. INTRODUCTION

Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fi elds of view to produce a segmented panorama or high-resolution image. Commonly performed through the use of computer software, most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results. It is also known as mosaicing. “Stitching” refers to the technique of using a computer to merge images together to create a large image, preferably without it being at all noticeable that the generated image has been created by computer [1]. The goal of image stitching is to seamlessly combine multiple overlapping images into a single, unified composite. This process typically involves detecting salient features in each image, matching corresponding features between image pairs, estimating transformations, and blending the results to produce a visually coherent output.

The image registration process typically consists of five main stages: feature detection and description, feature matching, outlier rejection, derivation of the transformation function, and image reconstruction. Feature detection involves identifying salient points (also known as key points or interest points) within an image. These points often correspond to distinctive structures such as corners, edges, blobs, junctions, or line intersections [2]. In the context of image stitching, commonly used feature detectors include SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. Each of these algorithms is paired with its own feature descriptor and presents unique advantages and trade-offs in terms of accuracy, computational cost, scale invariance, and robustness to illumination or viewpoint changes. In this study, the Scale-Invariant Feature Transform (SIFT) algorithm is selected due to its proven robustness in detecting repeatable key points across varying scales, orientations, and lighting conditions, making it highly suitable for high-quality image stitching applications.

A critical step in the image stitching pipeline is feature matching, also referred to as keypoint pairing, where key points from one image are paired with those from another. This pairing process serves as the backbone of the entire image stitching task. If key points from image A can be accurately matched to those in image B, the images can generally be stitched together with high precision. However, this process is inherently challenging due to ambiguous correspondences, false matches, and non-uniform keypoint distributions, particularly in scenes with low texture, repetitive patterns, or inconsistent lighting conditions.

One of the most basic methods for keypoint pairing is nearest-neighbor matching, which often uses the brute-force algorithm to compare each keypoint descriptor from one image to every descriptor in the other. While straightforward, this approach becomes computationally expensive as the number of key points increases. To improve efficiency, more sophisticated yet approximate methods such as k-nearest neighbors (KNN) using FLANN-based matchers or KD-tree data structures are commonly used. These heuristic-based approaches significantly reduce computational time but do not guarantee globally optimal pairings between the two sets of key points. As a result, they may sacrifice match quality in exchange for speed, which can affect the final stitching accuracy.

To address these limitations, this study will introduce two methods for feature matching strategy: greedy based feature matching, and Kuhn-Munkres (Hungarian) based feature matching algorithm. Both of this strategy will have its own

trade-off that will be discussed later. Main purpose of this study is to introducing alternative method while also give comparative study for each introduced method, while comparing the result to commonly used algorithm.

II. THEORY

A. Image

In the digital devices, there are several ways on how an image could be stored. First, there is a pixel-based representation, and also there is a vector-based representation. A pixel-based representation partitions the image into a grid of a certain dimension (width × height), and each pixel has one value which correspond to the light intensity at that position. On the other hand, vector-based representation takes into account that an image is composed of primitives such as lines, circles, or curves, so that the components of the images is specified by those information. One advantage of this representation is that when zooming an image, the image could be reconstructed using the equations, so that there is no loss of sharpness. In the pixel-based representation, however, when zooming out the picture the pixel size relative to the image stays the same, and to increase sharpness we need to interpolate the image, which could not be done automatically and/or easily like what the vector-based representation does with the equations of the lines and circles.

In this study, we will be more focused on pixel-based representation, since in this study, the focus is placed on pixel-based representation, as it aligns with the requirements of image stitching techniques. Image stitching operates directly on raster image data, where operations such as feature detection, keypoint matching, and image warping are performed on pixel-level information extracted from the grid.

In pixel-based representation, image will be represented as 2D-Array, where f(x, y) is represent intensity value, commonly represented as RGB (Red, Green, Blue) value, within that specific spatial coordinate. In RGB color system, there are three channels, one for each color red, green, and blue. Each channel is an M x N matrix, so in an RGB image, the size of the matrix is M x N x 3, i.e. each position has three values for each channel. These three values are then used to obtain the true colors of the pixel, shown in the figure below. Each value can be from 0 to 255 (with data type integer) or norm. Each value can be from 0 to 255 (with data type integer) or normalized to between 0 to 1 (real-valued). When the components of red, green, and blue are all the same, the resulting color will be grayscale

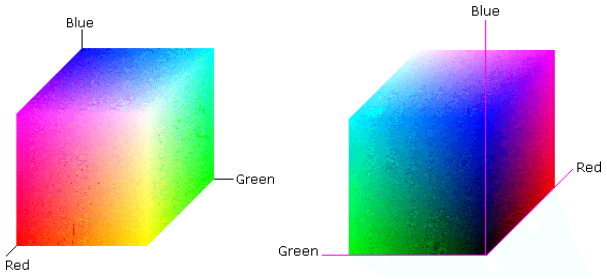


Fig 1. RGB Color space

B. Feature and Feature Extraction

A feature in computer vision is an identifiable part of an image or video that conveys meaningful information for tasks like object detection, tracking, or classification. Features are often distinct patterns, such as edges, corners, textures, or specific shapes, that algorithms use to understand and analyze visual data. For example, in a photo of a car, features might include the edges of the windshield, the corners of the license plate, or the texture of the tires. These features help reduce the complexity of raw pixel data by focusing on key elements that are relevant to solving a problem. By extracting and comparing features, algorithms can recognize objects, match images, or detect changes across frames in a video.

The detected features are subsequently described in logically different ways on the basis of unique patterns possessed by their neighboring pixels. This process is called feature description as it describes each feature by assigning it a distinctive identity which enables their effective recognition for matching. Some featuredetectors are readily available with their designated feature description algorithm while others exist individually [2].

In this study, we will more focused on a specific feature extraction method called SIFT (Scale Invariant Feature Transform). SIFT stands for Scale-Invariant Feature Transform and was first presented in 2004, by D. Lowe, University of British Columbia. SIFT is invariance to image scale and rotation [4]. Compared to other feature extraction process, SIFT offer a lot of advantages, including :

- Locality: features are local, so robust to occlusion and clutter (no prior segmentation)
- Distinctiveness: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Efficiency: close to real-time performance
- Extensibility: can easily be extended to a wide range of different feature types, with each adding robustness
- Resistance : SIFT is designed to identify and describe local features in images that are robust to changes in scale, rotation, illumination, and minor affine transformations.

The Scale-Invariant Feature Transform (SIFT) algorithm operates through a series of well-defined steps, each designed to ensure that the detected key points are distinctive, repeatable, and invariant to scale, rotation, and illumination changes. The main steps are described as follows:

- Scale-space peak Selection

  One of the fundamental strengths of SIFT is its ability to detect features that are invariant to scale. This is accomplished by constructing a scale-space representation of the image through Gaussian blurring

at multiple scales. The Difference-of-Gaussians (DoG) is then used to identify local extrema (minima or maxima) across spatial and scale dimensions. The primary goal of this step is to detect candidate key points that may correspond to potential features at various scales.

- Key points Localization

  While the scale-space step generates a large number of key points, not all of them are reliable. Some may lie along edges or in low-contrast regions, making them unstable. This step involves accurately refining the location of each keypoint by fitting a 3D quadratic function to the DoG response, and discarding those that do not meet specific contrast or edge-response thresholds. This improves the stability and distinctiveness of the remaining key points

- Orientation Assignment

  To achieve rotation invariance, each keypoint is assigned one or more orientations based on the local image gradients within a neighborhood around the keypoint. The dominant gradient orientation is used to align the keypoint descriptor, ensuring that the resulting feature vector remains stable even if the image is rotated.

- Keypoint Descriptor

  With location, scale, and orientation assigned, the next step is to compute a robust descriptor for each keypoint. A window of size 16×16 is taken around the keypoint and divided into 4×4 subregions. In each subregion, an 8-bin histogram of gradient orientations is computed, resulting in a 128-dimensional feature vector. This descriptor is designed to be highly distinctive and resilient to changes in illumination, viewpoint, and minor geometric distortions

At the end of SIFT process, we will expect result in form of set of features or key points, where each key points represented by high dimensional vector (128-dimensional vector) and the position (spatial coordinate) of the location of each key points.

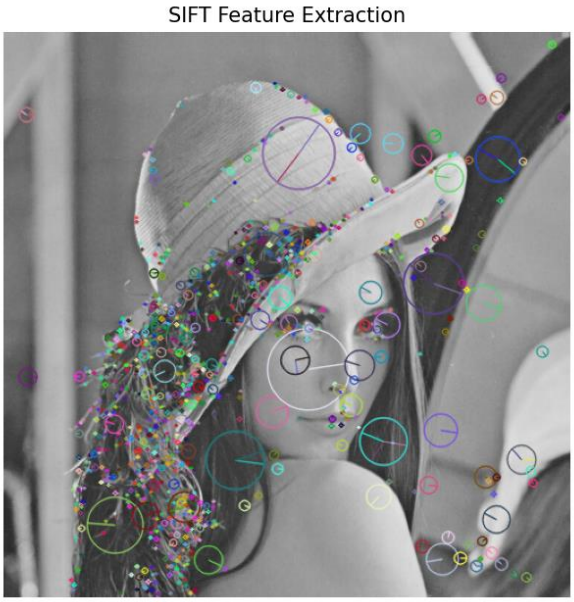


Fig 2. SIFT Key points Visualization

C. *Cosine Similarity*

Cosine similarity is a metric commonly used to measure the similarity between two non-zero vectors in a high-dimensional space. In the context of computer vision and image processing, it is often employed to compare feature descriptors, such as SIFT vectors, by assessing their directional alignment rather than their absolute magnitudes. Given two vectors A and B, the cosine similarity is defined as:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

D. *Greedy Algorithm*

A greedy algorithm is an algorithm that solves a problem step by step in such a way that, at each step:

1. It selects the best possible choice available at that moment, without considering future consequences (following the principle of "take what you can get now");
2. and hopes that by consistently choosing the local optimum at each step, it will eventually arrive at a global optimum. [3]

Greedy algorithm is a most popular and simple method for solving optimization problems (maximation and minimization problems). However, greedy algorithms do not guarantee an optimal solution for all problems. Their effectiveness depends on whether the problem exhibits the greedy-choice property (i.e., a globally optimal solution can be reached by choosing local optima) and optimal substructure (i.e., the optimal

solution to the problem contains the optimal solution to its subproblems).

In the context of feature matching in computer vision, a greedy approach can be used to quickly pair descriptors from two sets by always choosing the nearest unmatched feature according to a distance metric (e.g., cosine or Euclidean distance). While fast, this method may lead to many-to-one matches and suboptimal pairings, which can degrade the quality of tasks such as image stitching.

E. Kuhn-Munkres Algorithm

Kuhn-Munkres Algorithm (Hungarian Algorithm) was originally proposed to solve the Optimal Assignment Problem, which involves finding a one-to-one mapping (or matching) between two sets of elements, typically formulated as nodes in a bipartite graph, such that the total cost associated with the selected pairings is minimized. Each possible assignment has an associated cost, and the goal is to choose a subset of assignments that forms a perfect matching with the lowest total cost.

In the context of feature matching for image stitching, the Kuhn-Munkres algorithm can be applied by modeling the matching process as a bipartite graph, where one set of nodes represents the key points from image A and the other set represents the key points from image B. The cost matrix is constructed based on a distance metric (e.g., cosine distance or Euclidean distance) between the feature descriptors of key points from both images. Each entry $C_{i,j}$ in the matrix represents the "cost" of matching key point i from image A with key point j from image B.

By applying the Kuhn-Munkres algorithm to this matrix, the goal is to find the minimum-cost perfect matching, where each keypoint in the smaller set is matched with a unique keypoint in the other set. This approach ensures that the overall matching is globally optimal, unlike greedy or approximate methods which may result in duplicate pairings, missing assignments, or suboptimal total cost. Although the algorithm has a higher computational complexity, typically $O(N^3)$ for an NxN matrix, its accuracy and deterministic nature make it well-suited for applications where precise and reliable feature correspondences are essential

III. METHODOLOGY

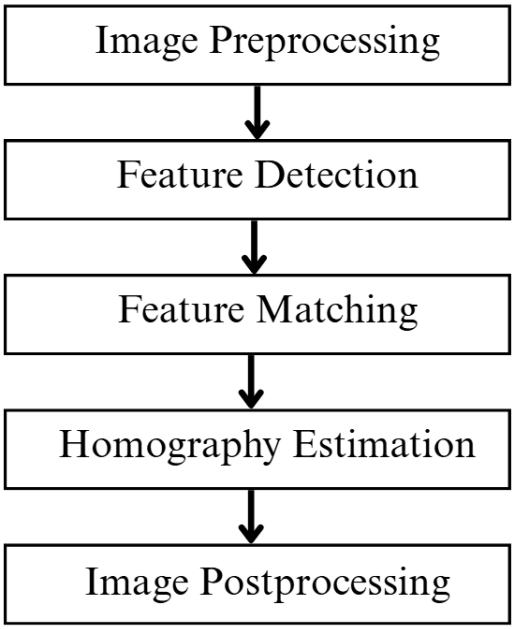


Fig 3. Process pipelining

The proposed image stitching pipeline is composed of several sequential stages: image preprocessing, feature detection, feature matching, homography estimation, and image postprocessing. Each stage plays a critical role in ensuring accurate alignment and seamless blending between input images.

A. Image Preprocessing

Before performing feature extraction, input images undergo several preprocessing steps to normalize their size and enhance consistency:

- Resizing: Large images are scaled down proportionally to ensure that the algorithm remains efficient and memory usage is controlled, especially when working with high-resolution datasets.
- Grayscale Conversion: Since SIFT operates on intensity gradients rather than color information, each image is converted to grayscale, reducing the data complexity while retaining the structural features necessary for detection.

These steps help reduce computational cost and improve robustness across diverse inputs.

B. Feature Detection

After preprocessing, the algorithm detects distinctive key points in each image using the Scale-Invariant Feature Transform (SIFT). As we already discussed in previous part, SIFT identifies points of interest that are invariant to scale, rotation, and illumination changes by constructing a scale-space and detecting local extrema. Each detected key point is then described using a 128-dimensional feature vector, which

encodes local gradient information around the point. Main purpose of this feature detection and feature extraction is to reduce working on all pixels and only focused to meaningful one. The number of features will usually be a lot less than number of pixels in an image. After this step, other process will be centered around this features / key points.

C. Feature Matching

In this study, three distinct feature matching methods are implemented and compared: the FLANN-based method, a greedy-based approach, and the Kuhn-Munkres (Hungarian) algorithm. Each method provides a different trade-off between computational efficiency and matching accuracy, and their effectiveness is evaluated in the context of key point pairing for image stitching.

The FLANN-based matcher (Fast Library for Approximate Nearest Neighbors) is widely used due to its efficiency in handling high-dimensional descriptor spaces. It utilizes approximate nearest neighbor search structures such as KD-trees or hierarchical clustering trees, making it significantly faster than brute-force methods. However, this speed comes at the cost of potentially suboptimal matches, as it relies on heuristics and may not always return the globally best pairings. To improve match quality, the FLANN-based matcher will ultilize the k-nearest neighbor (k-NN) search strategy with k = 2, followed by a ratio test introduced by David Lowe in the original SIFT paper. In this study, a ratio threshold of 0.75 is used, meaning that for each pair of closest (m, n) the match mmm is accepted only if:

$$Distance(m) < 0.75 * Distance(n)$$

The greedy-based matcher operates by selecting, for each keypoint in the first image, the closest unmatched keypoint in the second image based on a similarity metric (e.g., cosine distance). While simple and computationally lightweight, this method is also prone to mismatches, especially in cluttered scenes or when descriptor distributions are uneven.

In contrast, the Kuhn-Munkres algorithm (also known as the Hungarian algorithm) frames the matching process as a bipartite graph assignment problem and seeks a globally optimal one-to-one correspondence that minimizes total matching cost. Although computationally more intensive, it is deterministic and tends to yield higher-quality matches by avoiding duplicate assignments and minimizing overall matching error. The assignment will rely heavily on cosine similarity when comparing cost of the assignment. More specifically, a cost matrix is constructed where each element represents the cosine distance (1 − cosine similarity) between a descriptor in the first image and a descriptor in the second image. The Kuhn-Munkres algorithm then solves the linear assignment problem on this cost matrix, ensuring that each keypoint in one image is matched to a unique keypoint in the other, while minimizing the total cost across all assignments. In this study, a cosine distance threshold of *0.20* is used to filter out poor matches when using the Kuhn-Munkres algorithm. In other words, only keypoint pairs with a cosine distance less than or equal to *0.20* are considered valid matches, ensuring that the resulting assignments reflect a high degree of descriptor similarity and reducing the likelihood of false correspondences.

D. Homography Estimation

Once keypoint correspondences are established using one of the matching methods, the next step is to estimate the geometric transformation between images. This step will follow widely used pipeline : Homography estimation using RANSAC (Random Sampling Consencus) Approach. RANSAC is particularly well-suited for handling noisy or imperfect data, as it robustly estimates the transformation by repeatedly selecting random subsets of keypoint matches, computing candidate homographies, and evaluating how many of the remaining correspondences agree (i.e., lie within a certain reprojection threshold). The homography with the highest inlier count is selected as the final transformation matrix.

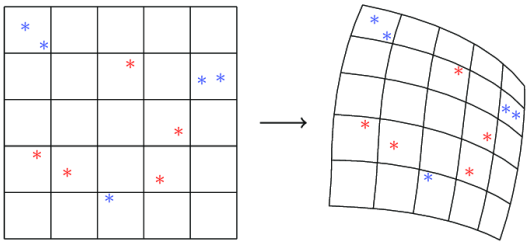


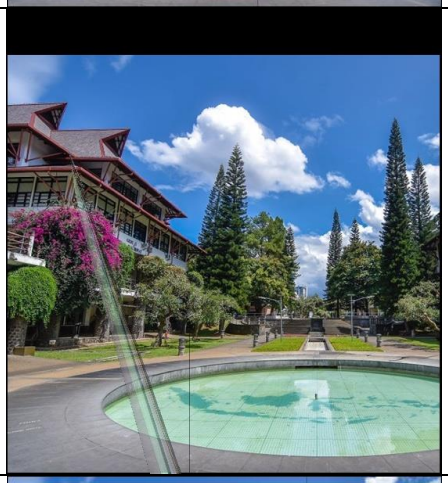
Fig 4. Homography

The expected outcome at the end of the homography estimation step is a 3×3 transformation matrix H that defines a projective mapping from the coordinate space of one image to another. This matrix encapsulates rotation, translation, scaling, and perspective distortion, allowing keypoints and features from the source image to be aligned accurately with their corresponding locations in the destination image. A successful homography estimation enables precise image warping, such that the overlapping regions between images can be correctly aligned and blended, minimizing visible seams and structural inconsistencies in the final stitched image.

IV. Result and Discussion

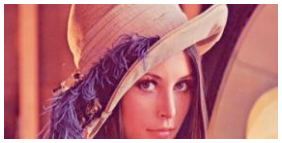
A. Case I – Horizontal Images Stiching

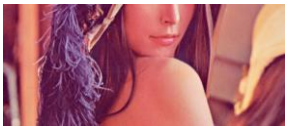
Image Input :

Method	Stitch Result	Time
FLANN		0.62s
Greedy		0.41s
KMA		0.73s

B. Case II – Vertical Images Stiching

Image Input :





Method	Stitch Result	Time
FLANN		0.23s
Greedy		0.22s
KMA		0.31s

C. Case III – Combination of Horizontal and Vertical

Image Input :

Method	Stitch Result	Time
FLANN		1.38
Greedy		0.96
KMA		1.69s

V. Conclussion

The Greedy matching method, while being the fastest, might produced faulty or suboptimal results due to its lack of global optimization leading to ambiguous or duplicate matches that affect homography estimation and the visual quality of the stitched images. In contrast, the FLANN-based matcher demonstrated consistent performance across most scenarios, effectively balancing speed and accuracy, offering a reliable trade-off between efficiency and matching quality. On the other hand, the Kuhn-Munkres algorithm—though computationally the slowest, but as consistent as FLANN, and theoretically better than FLANN-based matcher.

Github Repository

https://github.com/Fariz36/Makalah_STIMA_ImageStitching

References

[1] Mehta, J.D., Bhirud, S.G. (2011). Image stitching techniques. In: Pise, S.J. (eds) Thinkquest~2010. Springer, New Delhi. https://doi.org/10.1007/978-81-8489-989-4_13

[2] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2018, pp. 1-10, doi: 10.1109/ICOMET.2018.8346440. keywords: {SIFT;SURF;KAZE;AKAZE;ORB;BRISK;RANSAC;nearest neighbor distance ratio;feature detection;feature matching;image registration;scale invariance;rotation invariance;affine invariance;image matching;mosaicing},

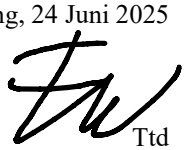
[3] R. Munir, “Algoritma Greedy (Bagian 1),” Institut Teknologi Bandung, Bandung, Indonesia, Apr. 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/04-Algoritma-Greedy-(2025)-Bag1.pdf. [Accessed: Jun. 24, 2025].

[4] D. Tiwari, “Introduction to SIFT — Scale Invariant Feature Transform,” Medium, Aug. 22, 2020. [Online]. Available: https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40. [Accessed: Jun. 24, 2025].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025



Ttd

Nama dan NIM