Robust Image Stitching with Superglue Keypoint Extraction and Blending

Fabian Perez, Paula Arguello , Mariana Robayo
Universidad Industrial de Santander
{nelson2200183, paula2191444, mariana2195091}@correo.uis.edu.co

Abstract

Image stitching is the process of creating panoramic images and seamless visual experiences. However, traditional image stitching algorithms, which rely on hand-designed heuristics, often struggle with handling varying lighting conditions and complex scenes. This paper presents a robust image stitching framework that leverages advanced feature extraction and matching techniques to achieve high-quality results. We utilize SuperGlue, a state-of-the-art neural network architecture, for feature extraction and matching, significantly enhancing the accuracy and robustness of keypoint correspondences between image pairs. Following feature extraction, we employ algorithms to find the homography matrix, ensuring the alignment of images despite the presence of outliers. Finally, the aligned images are blended using weighted masks to produce a seamless and visually appealing panoramic image. Our experimental results validate the effectiveness of integrating these stages for robust image stitching, making it a promising solution for various applications in computer vision and graphics. Additionally, we demonstrate that our approach is effective with real-world images. Our code is publicly available at https://github.com/Factral/image-stitching-supeglue

1. Introduction

Image stitching is a fundamental technique in computer vision, used to combine multiple images into a single panoramic image, enhancing the field of view beyond the capability of a single camera shot. This technology has a wide range of applications, including virtual reality [7], medical imaging [12], and geographic information systems [5]. Despite its utility, achieving seamless image stitching remains a challenging problem [11], especially in the presence of varying lighting conditions, complex scenes, and image distortions.

Traditional image stitching algorithms typically rely on hand-designed heuristics for feature extraction and matching. Commonly used feature extractors such as SIFT (Scale-Invariant Feature Transform [6]), SURF (Speeded-Up Robust Features [2]), and ORB (Oriented FAST and Rotated BRIEF [1]) identify keypoints and descriptors based on predefined criteria. While these methods have been effective, they often struggle with robustness and accuracy, particularly in scenes with repetitive patterns or significant viewpoint changes. Additionally, traditional approaches may not effectively handle outliers, leading to misalignment and visual artifacts in the final stitched image [10].

To address these limitations, this paper proposes a novel image stitching framework that leverages recent advancements in deep learning and robust estimation techniques. Specifically, the proposed method employ SuperGlue [9], a cutting-edge neural network architecture designed for feature matching, to enhance the accuracy and robustness of keypoint correspondences between image pairs. SuperGlue utilizes graph neural networks and attention mechanisms to dynamically weigh the importance of keypoints and their relationships, significantly improving feature matching performance over traditional methods.

Following feature extraction and matching, algorithms are employed to estimate the homography matrix, ensuring accurate alignment of images. The most commonly used algorithm is RANSAC [4], a robust method for model fitting even in the presence of significant outliers. The final step in our framework involves blending the aligned images using weighted masks to produce a seamless panoramic image. This is achieved by applying a linear or non-linear weighting scheme, which ensures a smooth transition between images without visible seams or abrupt changes in the image texture or color.

Our experimental results demonstrate that the proposed framework shows high accuracy and robustness in image stitching. Additionally, validations were performed with real-world scenarios using photos taken from a cellphone, a commonly used device for casual photography. This real-world validation underscores the practical applicability and reliability of the method, making it a versatile solution for various applications in computer vision and graphics.

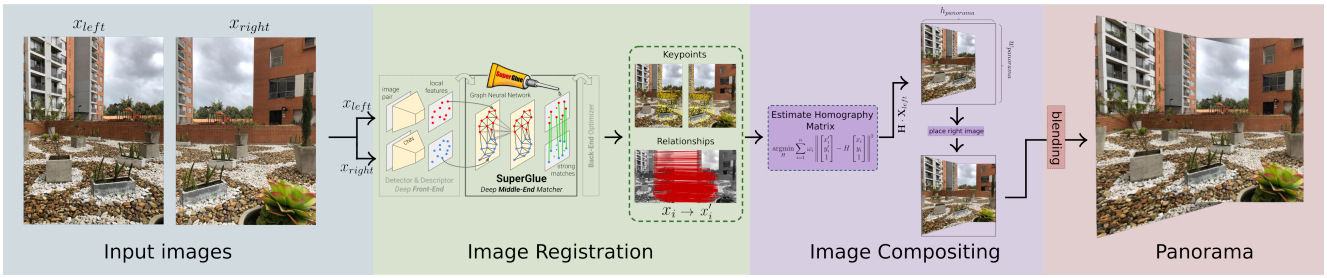


Figure 1. **Overview of the proposed image stitching framework.** The process begins with the input images x_{left} and x_{right}. During the Image Registration phase, keypoints are detected and described using deep features, followed by feature matching using the SuperGlue neural network. The homography matrix H is estimated to align the images accurately. In the Image Compositing phase, the right image is placed according to the estimated homography, and the images are blended to produce the final panoramic output.

2. Proposed Method

In the proposed image stitching method illustrated in Figure 1, the SuperGlue method is used, a cutting-edge neural network architecture, to perform feature extraction and matching. This approach enhances the precision and robustness of keypoint correspondences between image pairs. Using the SuperGlue-generated partial assignment matrix P, the method ensures that only the most reliable matches are considered. Subsequently, the filtered correspondences are used to compute the homography matrix. Once the images are aligned through the calculated homography, the final step involves a blending technique that uses a blending weighted masks approach. This technique ensures a smooth gradient at the seams between images, minimizing any visible lines or distortions, and thus creating a seamless and high-quality panoramic image.

2.1. Feature extraction: SuperGlue

The first step consists of the SuperGlue approach, which begins by encoding each keypoint's features through a combination of its visual descriptors and spatial coordinates using an MLP encoder. The initial feature representation for each keypoint in the images is formulated as $\mathbf{x}_i^{(0)} = \mathbf{d}_i + \text{MLP}_{\text{enc}}(\mathbf{p}_i)$. This encoding enriches the input features by incorporating both appearance and positional information, which is crucial for the subsequent steps of the process.

The method then utilizes an Attentional Graph Neural Network to perform feature aggregation through a mix of self-attention and cross-attention mechanisms.

Each keypoint i generates query, and each keypoint j provides a key and a value as following:

$$q_i = W_1(\ell)\mathbf{x}_i^{(\ell)} + b_1 \tag{1}$$

$$k_j = W_2(\ell)\mathbf{x}_j^{(\ell)} + b_2 \tag{2}$$

$$v_j = W_3(\ell)\mathbf{x}_j^{(\ell)} + b_3 \tag{3}$$

The attention mechanism computes the message $m_{\epsilon \to i} = \sum_{j:(i,j)\in\epsilon} \alpha_{ij} v_j$, where α_{ij} is the softmax over the dots of the queries and keys, effectively weighting the information based on their relevance.

After aggregating the features, the Optimal Matching Layer computes the partial assignment matrix P using a linear optimization framework. The pairwise scores S_{ij} are determined by the inner product of the feature vectors from images A and B,

$$S_{ij} = \langle f_i^A, f_j^B \rangle \tag{4}$$

This score matrix S forms the basis for optimizing the assignment matrix P through the Sinkhorn algorithm, which applies a series of normalization steps to make S doubly stochastic, adjusting the scores to manage both matched and unmatched keypoints effectively.

The Sinkhorn algorithm iteratively adjusts the scores by normalizing them across rows and columns, ensuring that the solution is feasible within the constraints of the assignment problem. This normalization process involves iteratively applying the exponential function to the score matrix S and normalizing it along rows and columns, similar to softmax operations of rows and columns. Specifically, the Sinkhorn algorithm, a differentiable version of the classically used Hungarian algorithm for bipartite matching, consists in iterative normalization $\exp(S)$ along rows and columns. After T iterations, the dustbins are dropped to recover the refined assignment matrix $P = P_{1:M,1:N}$. This matrix P is then utilized to establish robust correspondences between keypoints in the two images.

2.2. Homography Matrix

The second step of the proposed approach to image stitching involves estimating the homography matrix H, which is vital to seamlessly align images.

The homography matrix H is a 3x3 transformation matrix that maps the points in one image to the corresponding points in another image, as following:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

where (x', y', w') are the homogeneous coordinates in the destination image. And the general form of a homography matrix H is given by:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{6}$$

Here, $h_{11}, h_{12}, h_{21}, h_{22}$ contribute to rotation, scaling, and shear transformations. h_{13}, h_{23} represent translation in the x and y directions, respectively. They determine how much the image is shifted horizontally and vertically in the output image. h_{31}, h_{32} contribute to perspective distortion. And, h_{33} is often normalized to 1 for convenience.

The partial assignment matrix P mentioned earlier establishes initial correspondences between key points in two images. Each element P_{ij} represents the probability that the i-th keypoint in the first image corresponds to the j-th keypoint in the second image. This matrix is crucial because it filters out unreliable matches and highlights the most probable correspondences based on learned feature representations and spatial relationships.

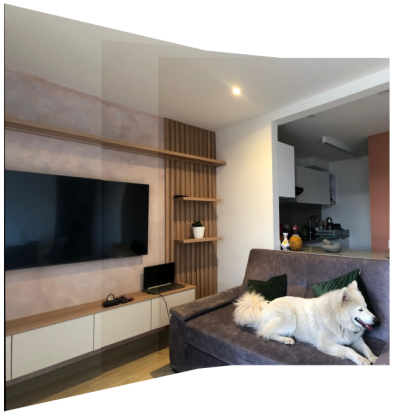
Then, the matrix P is used to select a set of keypoint pairs with high confidence scores. The homography H is computed in each iteration by solving the following optimization problem:

$$\underset{H}{\operatorname{argmin}} \sum_{i=1}^{n} \omega_i \left\| \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} - H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \right\|^2 \tag{7}$$

where $\{x_i, y_i\}$ and $\{x_i', y_i'\}$ are the corresponding points in the first and second images, respectively, and ω_i are weights derived from the matrix P, indicating the confidence in each keypoint match.

To solve this optimization problem, the function `cv2.findHomography` in *OpenCV* [3] was used, which returns the homography matrix H. The function solves the system using methods such as the Direct Linear Transformation (DLT) [8] algorithm. The DLT algorithm estimates H by minimizing the geometric error between the transformed points and their corresponding points in the second image. Specifically, it adjusts H to ensure that the transformation closely maps the points in the first image (left) to their counterparts in the second image (right). Additionally, the function can apply a robust estimation method like RANSAC (Random Sample Consensus) [4] to handle outliers. This ensures that the estimated homography is accurate even when some point correspondences are incorrect.

(a) Three distinct photographs from a single scene.



(b) Final scene following the warping process.

Figure 2. Intermediate result showcasing the stitched panorama after the warping stage and before the blending process.

The result is a 3×3 homography matrix H that best transforms the points from the first image (left) to their corresponding locations in the second image (right).

2.3. Image Warping

Furthermore, the matrix H is utilized to divide the two images into a single panoramic image. The process starts by transforming the corner points of the left image to determine where these corners will map onto the panorama canvas. The corners are defined as follows:

$$C = \begin{bmatrix} 0 & 0 & 1 \\ w_{\text{left}} - 1 & 0 & 1 \\ w_{\text{left}} - 1 & h_{\text{left}} - 1 & 1 \\ 0 & h_{\text{left}} - 1 & 1 \end{bmatrix}^\top \tag{8}$$

where w_{left} and h_{left} correspond to the width and height of the left image, respectively. Transforming these corners using the homography matrix H gives us the new positions:

$$C_t = H \cdot C \tag{9}$$

Subsequently, the bounding box that encompasses both the transformed right image and the original left image is determined by calculating the minimum and maximum coordinates along the x and y axes.

To ensure all parts of the panorama are visible, a translation matrix is applied:

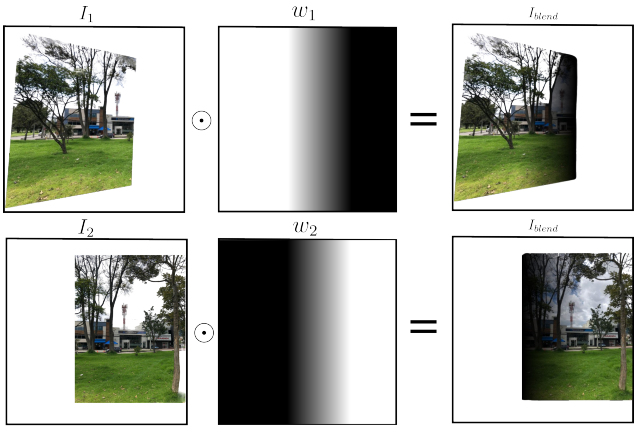


Figure 3. Illustration of the blending process in the proposed image stitching framework. The images I_1 and I_2 are the input images to be stitched. The masks $w1$ and $w2$ represent the blending masks applied to I_1 and I_2, respectively. The resulting images, I_{blend}, show the final blended outputs for each input image.

$$H_t = \begin{bmatrix} 1 & 0 & -x_{\min} \\ 0 & 1 & -y_{\min} \\ 0 & 0 & 1 \end{bmatrix} \cdot H \tag{10}$$

This matrix shifts the entire canvas, ensuring that all image content has non-negative coordinates.

Using H_t, the left image is warped onto the panorama canvas with the function `cv2.warpPerspective`. This function transforms the left image such that its coordinates align with those of the right image as dictated by the homography matrix. The function parameters adjust how the image is interpolated and handle pixel values outside the image boundaries.

After warping, the right image is placed onto the panorama canvas based on the calculated translations. This step ensures that the right image anchors the panorama and the left image seamlessly merges into it, resulting in a visually cohesive panoramic image. The process is executed one by one, adding each image sequentially to build the final composition. The intermediate results, as demonstrated in Figure 2, show the precise alignment of three images before the final blending process, highlighting the framework's ability to handle multiple images effectively.

2.4. Blending Images for Seamless Panoramas

The blending process is crucial to ensure a smooth transition between the warped images. Therefore, a weighted average blending technique was used for merging overlapping regions of the images. This approach consists of calculating each pixel in the overlapping region of two images by taking a weighted average of the corresponding pixels from each image:

$$I_{\text{blend}} = \frac{w_1 \cdot I_1 + w_2 \cdot I_2}{w_1 + w_2} \tag{11}$$

where I_1 and I_2 correspond to the left and right images, respectively, and w_1 and w_2 are the weights assigned to the corresponding pixels of the images. These weights are designed such that they smoothly transition from one image to the other as follows:

$$w_1(x) = 1 - \frac{x - x_{\text{start}}}{x_{\text{end}} - x_{\text{start}}}, \quad w_2(x) = \frac{x - x_{\text{start}}}{x_{\text{end}} - x_{\text{start}}} \tag{12}$$

Here, w_1 starts at 1 on the left of I_1 and linearly decreases to 0 at the far right, while w_2 does the opposite, starting at 0 and increasing to 1. In this formulation, x is the horizontal coordinate in the overlap region, and x_{start} and x_{end} define the boundaries of this region. Figure 3 shows an example of the blending weights used for this step.

2.5. Cropping the panorama

The final stage in panorama creation involves cropping the stitched image to focus on the region of interest and remove any unwanted areas or artifacts. This process is essential for achieving a clean and visually appealing panorama.

The cropping algorithm calculates the minimum x coordinate ($x_{\min}$) and the corresponding y coordinate ($y_{\min_{\text{est}}}$) from the corners provided to determine the cropping limits. The vertical boundaries for the crop, $y_{\min}$ and $y_{\max}$, are determined on the basis of the comparison between the vertical coordinates of the warped image corners and the top and bottom edges of the destination image's height. Specifically, $y_{\min}$ is set to the highest of the corner y coordinate or $y_{\min_{\text{est}}}$, and $y_{\max}$ is set based on whether the corner y coordinates exceed the sum of $y_{\min_{\text{est}}}$ and the height of the destination image (h_{dst}).

$$(x_{\min}, y_{\min_{\text{est}}}) = \min(\text{corners}) \tag{13}$$

$$y_{\min} = \max(\min(y_{\text{corner}_1}, y_{\text{corner}_2}), y_{\min_{\text{est}}}) \tag{14}$$

$$y_{\max} = \min(\max(y_{\text{corner}_3}, y_{\text{corner}_4}), y_{\min_{\text{est}}} + h_{\text{dst}}) \tag{15}$$

The horizontal boundary, $x_{\min}$, is adjusted to ensure that the panorama does not include any part of the image beyond the left edge of the destination image. The panorama is then cropped using these calculated boundaries:

$$x_{\min} = |x_{\text{corner}_0} - x_{\text{corner}_3}| \tag{16}$$

$$\text{pano} = \text{panorama}[y_{\min} : y_{\max}, x_{\min} :, :] \tag{17}$$

This method ensures that the resulting panorama is neatly trimmed to exclude any regions that fall outside the desired viewport, eliminating parts of the image that do not contain useful information or that contain artifacts from the stitching process.

Figure 4. **Visual results of the proposed image stitching framework.** The top row shows the acquired images for three different scenes, the bottom row presents the corresponding stitched images. *Scene 1* contains 2 photos of aerial shots, *Scene 2* contains 3 photos of a indoor space and *Scene 3* contains 4 photos of a tropical beach

3. Results

The proposed image stitching framework's effectiveness is vividly demonstrated across diverse scenes, as depicted in Figure 4. Following the implementation of the four stages described earlier, the resultant images clearly articulate the capabilities of the framework. Scene 1 features two aerial shots of an urban landscape captured with a cellphone, illustrating the framework's adeptness in seamlessly stitching complex architectural details without visible distortions. Scene 2 consists of three indoor images that feature a dog, also captured with a cellphone. In this scenario, the framework skillfully maintains visual continuity and adeptly manages varied lighting conditions, culminating in a coherent panoramic output. Scene 3, which merges four images sourced from https://github.com/AnhDuy26/Image-Stitching-Project/, exemplifies the framework's proficiency in handling natural environments, smoothly transitioning between consistent water textures and sky gradients. This scene highlights the precision of the SuperGlue method for feature matching and the blending technique's efficacy in minimizing visual artifacts. Together, these outcomes underscore the framework's robust capability to produce high-quality panoramas from both personal captures and internet-sourced images, demonstrating adaptability across a broad spectrum of environmental conditions and image qualities.

4. Conclusion

This paper has addressed several challenges inherent in traditional image stitching methods, particularly those related to robustness and accuracy in complex scenes with varying lighting conditions and significant viewpoint changes. By integrating the advanced capabilities of SuperGlue, a state-of-the-art neural network architecture for feature matching, our proposed framework significantly enhances the precision and reliability of keypoint correspondences. The robust estimation of the homography matrix further ensures that the images are accurately aligned, even in the presence of substantial outliers. Additionally, our blending technique, which applies weighted masks for image fusion, effectively eliminates visible seams, ensuring a smooth transition between concatenated images. The experimental results underscore the efficacy of the proposed framework, demonstrating high accuracy and robustness in stitching images from both controlled and real-world environments. In conclusion, the proposed image stitching framework represents a step forward in the quest for seamless panoramic imaging, addressing critical gaps in previous methodologies and setting a new standard for future research and application in this dynamic field.

References

[1] Prashant Aglave and Vijaykumar S Kolkure. Implementation of high performance feature extraction method using oriented fast and rotated brief algorithm. *Int. J. Res. Eng. Technol*, 4:394–397, 2015. 1

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006. 1

[3] Gary Bradski, Adrian Kaehler, et al. Opencv. *Dr. Dobb's journal of software tools*, 3(2), 2000. 3

[4] H Cantzler. Random sample consensus (ransac). *Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh*, 3, 1981. 1, 3

[5] Julian Colorado, Ivan Mondragon, Juan Rodriguez, and Carolina Castiblanco. Geo-mapping and visual stitching to sup-

port landmine detection using a low-cost uav. *International Journal of Advanced Robotic Systems*, 12(9):125, 2015. 1

[6] Tony Lindeberg. Scale invariant feature transform. 2012. 1

[7] Pavan Chennagiri Madhusudana and Rajiv Soundararajan. Subjective and objective quality assessment of stitched images for virtual reality. *IEEE Transactions on Image Processing*, 28(11):5620–5635, 2019. 1

[8] Bronislav Přibyl, Pavel Zemčík, and Martin Čadík. Absolute pose estimation from line correspondences using direct linear transformation. *Computer Vision and Image Understanding*, 161:130–144, 2017. 3

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1

[10] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007. 1

[11] LYU Wei, Zhou Zhong, Chen Lang, and ZHOU Yi. A survey on image and video stitching. *Virtual Reality & Intelligent Hardware*, 1(1):55–83, 2019. 1

[12] Kyi Pyar Win, Yuttana Kitjaidure, and Kazuhiko Hamamoto. Automatic stitching of medical images using feature based approach. *Advances in Science, Technology and Eng. Systems*, 4(2):127–133, 2019. 1