Technische Universität Berlin

Verkehrs- und Maschinensysteme, Fakultät V,
Institut für Werkzeugmaschinen und Fabrikbetrieb,
Fachgebiet Industrielle Automatisierungstechnik,
Pascalstraße 8-9,
10587 Berlin

Design and Investigation of an Automatic Recognition and Positioning System for Real-Time Reconstruction of Fragmented Documents

---

Bachelor Thesis

by

Diana Leo

Matriculation Number: 414183

First Supervisor:	Prof. Dr.-Ing. Jörg Krüger
Second Supervisor:	M. Sc. Oliver Krumpek

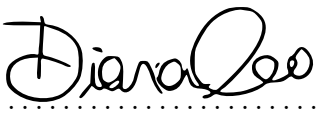
Berlin, January 31., 2024

Eidesstattliche Erklärung

Name:	Diana Leo
Matrikelnummer:	414183
Studiengang:	B.Sc. Computational Engineering Science (CES)

Hiermit versichere ich eidesstattlich, dass ich diese Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe. Die wörtlich und inhaltlich entnommenen Stellen aus den Quellen sind als solche kenntlich gemacht.

Berlin, 31.01.2024



.....

(Unterschrift)

Abstract

This thesis addresses the challenge of real-time reconstruction of fragmented documents, a traditionally laborious and time-consuming task. Central to this work is the enhancement of the FalKe system, developed by Fraunhofer IPK, which assists in the real-time reconstruction of fragmented documents by showing the user the exact location and orientation of the given fragments in the complete document image. A comprehensive analysis of the system, whose hardware consists of a camera, a projector and a newly integrated adjustable light, led to significant improvements in three key areas. A novel Camera-Projector Calibration process was developed to enhance the user interface's precision, essential for guiding users in the accurate placement of fragments during document reconstruction. To efficiently handle multiple reference documents, a Digitization and Storage System was developed, streamlining the integration and retrieval of reference document data. A significant part of the thesis was devoted to improving the accuracy of the reconstruction process in recognizing and positioning document fragments. To this end, four local feature matching methods were selected and rigorously tested to assess their robustness against noise, rotation, and scale when matching fragments with varying degrees of damage. The key findings from the experimental analysis revealed that SE(2)-LoFTR, a deep learning based method, and the traditional SIFT can significantly enhance the system's ability to deal with these conditions. SE(2)-LoFTR excels at finding matches in highly damaged scenarios, while SIFT is more effective on less damaged fragments. These developments and findings not only improve the FalKe system's reconstruction functionality in cultural heritage applications but also open up avenues for its use in industrial settings and lay the groundwork for future research and optimization.

Kurzfassung

Diese Bachelorarbeit beschäftigt sich mit der Herausforderung der Echtzeit-Rekonstruktion von fragmentierten Dokumenten, einer traditionell mühsamen und zeitaufwändigen Aufgabe. Im Mittelpunkt dieser Arbeit steht die Weiterentwicklung des am Fraunhofer IPK entwickelten FalKe Systems, das bei der Echtzeit-Rekonstruktion fragmentierter Dokumente hilft, indem es dem Benutzer die genaue Position und Ausrichtung der gegebenen Fragmente im gesamten Dokumentenbild anzeigt. Eine umfassende Analyse des Systems, dessen Hardware aus einer Kamera, einem Projektor und einer neuen regulierbaren Beleuchtung besteht, führte zu signifikanten Verbesserungen in drei Kernbereichen. Ein neuartiger Kamera-Projektor-Kalibrierungsprozess wurde entwickelt, um die Präzision der Benutzeroberfläche zu verbessern, die für die genaue Platzierung der Fragmente während der Dokumentenrekonstruktion unerlässlich ist. Um mehrere Referenzdokumente effizient zu handhaben, wurde ein Digitalisierungs- und Speichersystem entwickelt, das die Integration und den Abruf von Referenzdokumentdaten rationalisiert. Ein wesentlicher Teil der Arbeit wurde der Verbesserung der Genauigkeit des Rekonstruktionsprozesses bei der Erkennung und Positionierung von Dokumentenfragmenten gewidmet. Zu diesem Zweck wurden vier Local-Feature-Matching Methoden ausgewählt und experimentell untersucht, um ihre Robustheit gegenüber Rauschen, Rotation und Skalierung beim Matching von Fragmenten mit unterschiedlichem Beschädigungsgrad zu bewerten. Die experimentelle Analyse ergab, dass SE(2)-LoFTR, eine auf Deep Learning basierende Methode, und das traditionelle SIFT die Fähigkeit des Systems, diese Bedingungen zu bewältigen, erheblich verbessern können. SE(2)-LoFTR zeigt hervorragende Ergebnisse bei der Suche nach Matches in stark beschädigten Szenarien, während SIFT bei weniger beschädigten Fragmenten effektiver ist. Diese Entwicklungen und Erkenntnisse verbessern nicht nur die Rekonstruktionsfunktionalität des FalKe Systems für Anwendungen im Bereich des kulturellen Erbes, sondern eröffnen auch Möglichkeiten für den Einsatz in industriellen Umgebungen und legen den Grundstein für zukünftige Forschung und Optimierung.

Contents

List of Figures

List of Tables

List of Acronyms

ACNe Attentive Context Networks

BF Brute-Force

BRIEF Binary Robust Independent Elementary Features

BRISK Binary Robust Invariant Scalable Keypoints

CNN Convolutional Neural Network

DLT Direct Linear Transform

DoG Difference-of-Gaussian

FalKe Forschungsallianz Kulturerbe

FAST Features from Accelerated Segment Test

FCN Fully Convolutional Network

FLANN Fast Library for Approximate Nearest Neighbors

GAT Graph Attention Network

G-CNN Group Equivariant Convolutional Neural Networks

GMS Grid-based Motion Statistics

GNN Graph Neural Network

IOU Intersection Over Union

IPK Fraunhofer Institute for Production Systems and Design Technology

LIFT Learned Invariant Feature Transform

LoFTR Local Feature TRansformer

MAE Mean Absolute Error

NN Nearest Neighbor matching

OpenCV Open Source Computer Vision Library

ORB Oriented FAST and Rotated BRIEF

px pixels

RANSAC Random Sample Consensus

SG SuperGlue

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localization and Mapping

SP SuperPoint

SURF Speed Up Robust Features

UI User Interface

1 Introduction

1.1 Motivation

The deterioration of paper documents and archival materials is an inevitable process, threatened not only by natural aging but also by deliberate acts of vandalism and natural disasters. This reality highlights the critical need for their preservation and reconstruction.

In various applied disciplines, such as forensics, archaeology, and art restoration, the reconstruction of fragmented objects and shredded documents stands as a formidable challenge. The imperative to restore these objects to their original states for functional and cultural preservation is hindered by the laborious and often impractical nature of manual reconstruction attempts. In scenarios where multiple shredded documents or fragmented objects require reassembly, manual inspection becomes time-consuming, unfeasible, and risk-laden. The danger of further damaging fragile artifacts intensifies with each handling. Thus, there is a pressing need for methods that can automate, at least partially, the reconstruction process, which can be seen as an intricate form of jigsaw-puzzle-solving.

Over the past decades, the scientific community has extensively explored solutions for automating fragment reassembly problems. This exploration encompasses not only theoretical exercises in robotics and computer vision [BW89, BB93, GMB02] but also practical challenges in reconstructing real-world 3D and 2D-items, like wall paintings [MK03, SF16, PEP+08] or shredded documents used in forensic investigations [DS09, JOF06].

To address a specific use case of this complex reassembling problem, Fraunhofer Institute for Production Systems and Design Technology's (IPK), as part of the Research Alliance for Cultural Heritage (FalKe) - comprised of the Fraunhofer-Gesellschaft, the Leibniz Association, and the Prussian Cultural Heritage Foundation - developed an assistance system that enables real-time reconstruction of damaged documents by providing users with the location of single fragments on the complete picture. This system specifically deals with a scenario where a digital reference image exists for a physical document that has been torn apart in pieces with varying shapes and forms.

The system, referred to in subsequent chapters as the *FalKe system*, employs a camera and projector setup for a fast and interactive reconstruction process. The User Interface (UI), projected onto a plane surface, intuitively guides the user through the reconstruction, as shown in Figure 1.1. This UI is divided into two main sections: the detection area, which lies within the camera's view, for placing document fragments - also called snippets - and a reconstruction area for visualizing the virtually puzzled document. Users place snippets in the detection area. The system then employs AI-based computer vision techniques to analyze each snippet, matching its features with the reference image. The identified matches are then used to determine the position and orientation of each fragment within the original document, displaying this information in the reconstruction area. This allows the user to quickly manually assemble the document.

While primarily designed for damaged paper documents, this system also has the potential to adapt to other flat 3D objects, like tiles, offering a broader spectrum of usability. Beyond the realm of cultural heritage, this system holds significant potential for industrial applications. It could be crucial in assembling fragmented industrial components, assisting in quality control processes. Expanding into industrial applications not only showcases the

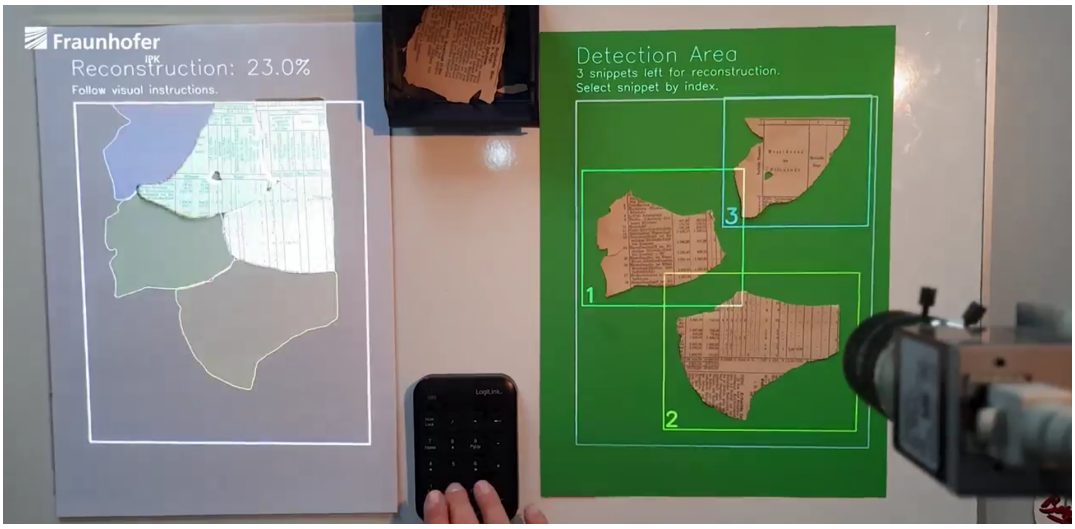


Figure 1.1: **Guided fragments reassembly through the FalKe system.** The user has placed 3 snippets in the *Detection Area* on the right and starts the reconstruction process. The system has recognised the 3 snippets and shows to the user their right placement within the complete document in the *Reconstruction Area* on the left, facilitating an efficient manual reassembly process.

system's versatility but also paves the way for future enhancements, emphasizing its potential and the scope for further optimization and research yet to be explored across various sectors.

1.2 Objective and Scope

The goal of this thesis is to enhance and refine the existing FalKe system, ensuring it operates cohesively to efficiently assist in the real-time reconstruction of fragmented documents. Addressing the system's limitations in both hardware and software components, this thesis aims to improve the system from the initial digitization of reference documents and fragments to the final reassembly. Figure 1.2 delineates a clear visualization of the system's architecture, with the specific areas of focus for this research highlighted in green. While the focus on hardware improvements is limited to the integration of a light with adjustable intensity, this thesis primarily explores possible solutions for the software components. Among these, the main effort has been put into an extensive exploration of the recognition and positioning methods to ensure the most accurate document reconstruction in various challenging real world conditions, like paper damage.

The software components are delineated into three macro areas, each playing a pivotal role in the document reconstruction process:

1. **Acquisition and Digitisation**. This area is responsible for capturing and interpreting images of reference documents and fragments. While this thesis does not delve into the *Image Acquisition* methods, it focuses on the development of a robust *Digitization and Storage system* for reference images. This new addition to the system would enable users to easily add images of undamaged reference documents to the system, which interprets and stores them with some key information. By pre-storing key data about reference images, the system's performance during reconstruction is optimized by reducing redundant computations.

2. **Reconstruction**. The core of the entire reconstruction process lies within this macro area. It consists of the *Segmentation* method that is crucial for isolating individual

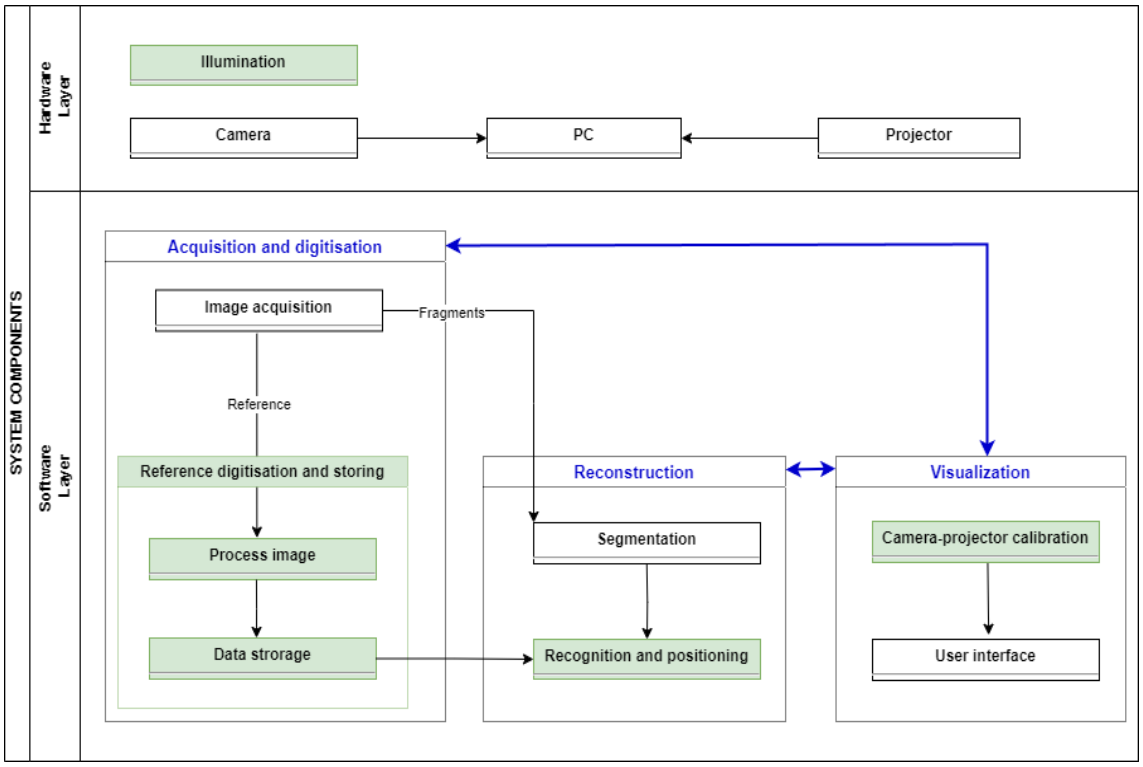


Figure 1.2: **Simplified diagram of the FalKe system.** This simplified representation outlines the FalKe system's major components, both hardware and software. Areas of particular relevance to this study are emphasized in green, illustrating the system's functional aspects critical to this research.

fragments from the captured images and the *Recognition and Positioning* method, which uses local feature matching algorithms to determine the correct placement of a fragment within the corresponding reference document. Although segmentation methods are not explored in this thesis, multiple local feature matching methods have been implemented and extensively investigated to ensure the most accurate reconstruction. The challenge here lies in achieving accurate and reliable document reassembly, independently on the orientation, scale or damage level of the given fragments.

3. **Visualisation**. Central to user interaction, the visualization area comprises the *Camera-Projector Calibration* and the *User Interface (UI)* components. The UI is a constant presence throughout the reconstruction process, acting as a visual guide for the user. While this thesis does not investigate the UI design, it emphasizes the importance of an accurate camera-projector calibration, by developing and implementing a three steps calibration process based on homography. A well-calibrated system is essential for ensuring precise translation between camera and projector coordinates, which is pivotal for the UI to provide accurate, real-time visual feedback and guidance during the reconstruction process.

The following key constituents encapsulate the overarching goals and summarize the core efforts that underpin this research:

- Literature research on state of the art matching methods and presentation of project-specific limitations of the individual algorithms
- Design and creation of a suitable digitization system for the acquisition of reference images and fragments, including suitable illumination

- Prototypical implementation of the concept for the recognition and positioning system based on the results of the literature review and the design parameters
- Critical consideration and evaluation of the used techniques and methods
- Outlook elaboration for follow-up work

1.3 Outline

The rest of the thesis is organized as follows:

Chapter 2 introduces the essential theoretical foundations necessary for understanding the methods and solutions implemented in this thesis.

Chapter 3 begins with a review of prior work in computer-aided reassembly of fragmented objects, showcasing the different approaches, then delves into local feature matching methods. This chapter provides an overview of both detector-based and detector-free methods, with a specific focus on the methods implemented and examined in this study.

Chapter 4 begins by examining the FalKe system's limitations and outlines the specific requirements identified for improving the addressed components. The chapter then delves into the specific solutions implemented to address these requirements, offering an in-depth look at the enhancements made across different system components. Furthermore, it describes the methodologies adopted for evaluating the effectiveness of the implemented local feature matching methods in the context of document reassembly, focusing on both their theoretical foundations and practical application.

Chapter 5 is dedicated to presenting the outcomes of the evaluations conducted on the implemented feature matching methods. This chapter details the performance of these methods in accurately reconstructing a variety of document types, particularly under challenging conditions. It aims to shed light on the practical effectiveness and adaptability of the developed solutions in real-life document reassembly scenarios.

Chapter 6 summarizes the main achievements of the thesis and provides an outlook on future work and potential areas for further research.

2 Fundamentals

This chapter lays the foundation for understanding the theoretical underpinnings crucial to the methodologies discussed later in Chapter 4. It covers essential concepts in projective transformations and neural networks in computer vision, providing the necessary background for comprehending the complex techniques employed in the system, specifically for the calibration process and the learning-based local feature matching methods detailed in Chapter 3. The prime source used for Section 2.1 is [HZ03].

2.1 Projective transformations

Projective transformations are key in image processing and computer vision for mapping points between different imaging planes. They form the core of the FalKe system's ability to calibrate the camera-projector setup and accurately recover the original place and orientation of the fragments within the complete reference document in the reconstruction process.

This section lays the groundwork for understanding these transformations, focusing on homography and affine transformations and their computation using the Random Sample Consensus (RANSAC) algorithm. First the homogeneous coordinates and the projective plane are explained, as well as the pinhole camera model. This framework is crucial for understanding transformations like homography and affine transformations, which will be outlined subsequently.

2.1.1 Homogeneous coordinates and the projective space

In projective geometry, points are represented using homogeneous coordinates, which are a generalized form of Cartesian coordinates. The advantage over the Cartesian coordinates is that homogeneous coordinates enable the unification of rotate, scale, and translate image transformations into a single matrix multiplication, enhancing computational efficiency.

A 2D Euclidean point (x, y) is represented as a triple $(x, y, 1)$ in homogeneous coordinates. For instance, $(x, y, 1)$ and (kx, ky, k) for any non-zero k represent the same point, meaning

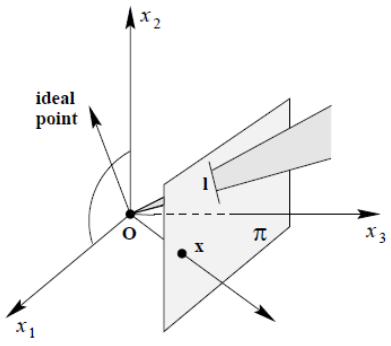


Figure 2.1: **Projective plane model**, from [HZ03]. In the projective plane $\mathbb{P}^2$, points are depicted as rays and lines as planes originating from the $\mathbb{R}^3$ space's origin. The $x_1 - x_2$ plane represents lines at infinity and rays lying there represent ideal points.

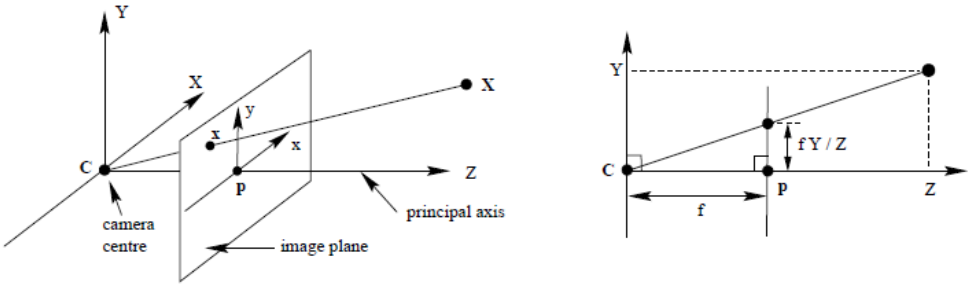


Figure 2.2: **Pinhole camera geometry**, from [HZ03]. The camera center C is situated at the coordinate origin. The image plane, placed in front of C, intersects the principal axis in the principal point p.

that homogeneous coordinates are equivalent by scale. The homogeneous coordinates of a point (kx, ky, k) can be normalized by dividing each coordinate by the third coordinate.

This system allows for the representation of points at infinity, also called ideal points. Points at infinity arise when the last coordinate is zero. This approach extends Euclidean space $\mathbb{R}^n$ to projective space $\mathbb{P}^n$, where n is the dimension of the space. The projective plane $\mathbb{P}^2$, illustrated in Figure 2.1, is modeled as a space of rays emanating from the origin in $\mathbb{R}^3$. Each ray represents a point in projective space, with planes through the origin representing lines in $\mathbb{P}^2$. This model ensures that two distinct rays intersect in exactly one plane, mirroring how two points determine a line in Euclidean space. In projective space, points at infinity form significant geometric entities like the line at infinity in two dimensions or the plane at infinity in three dimensions. Points at infinity are parallel to the plane $x_3 = 1$ and do not intersect it, illustrating the reach of projective geometry to capture concepts beyond traditional Euclidean limits.

2.1.2 Pinhole camera model

The pinhole camera model, illustrated in Figure 2.2, describes the central projection where points in 3D space are mapped onto a 2D image plane through a single point, the camera center. In homogeneous coordinates, a point $X = (x, y, z, 1)^T$ in $\mathbb{R}^3$ maps to $(fx, fy, z)^T$ on the image plane in $\mathbb{R}^2$, where f is the distance from the camera center to the image plane, known as the focal length. The line perpendicular to the image plane from the camera center is the *principal axis*, and the intersection of this axis with the image plane is the *principal point* p.

For more complex and realistic camera setups, the model is expanded using camera intrinsic and extrinsic matrices:

$$K = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}; M_{ex} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

The intrinsic matrix K represents the internal parameters of the camera, which include the focal lengths f_x, f_y and principal point coordinates c_x, c_y. It transforms 3D coordinates into 2D pixel coordinates. The extrinsic matrix M_{ex} describes the camera's orientation and position in the 3D space. It consists of both a 3x3 rotation matrix and a 3x1 translation vector. The rotation matrix is an orthogonal matrix, which handles the rotational transformation of 3D points from world to camera coordinates, involving rotations around the x, y, and z axes. These rotations define the camera's angular position in the world's 3D space. The translation

vector, on the other hand, indicates the camera center's position in the world's 3D space. This vector delineates the camera center's shift from the world coordinate system's origin.

The multiplication of the intrinsic matrix with the extrinsic matrix forms the 3x4 *camera matrix*, also called *projection matrix* denoted as P. This matrix is a cornerstone in developing a more comprehensive camera model. The following equation describes how a 3D point in homogeneous coordinates $(x, y, z, 1)^T$ can be projected to 2D image coordinates $(x', y', 1)^T$ in homogeneous coordinates, up to a scale factor λ:

$$\begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = P \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = KM_{ex} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.2}$$

The extrinsic matrix applies rotation and translation to transform the world coordinates into camera coordinates. Then, the intrinsic matrix further projects these coordinates onto the sensor plane, ultimately resulting in 2D image coordinates.

2.1.3 Homography

Homography arises as a special case in the pinhole camera model when projecting planar scenes. It applies when the scene being imaged is essentially flat. Under this scenario, the world scene's depth is constant, and the Z coordinate of the world points on the plane can be considered zero. This simplifies the projection matrix $P = KM_{ex}$ of the pinhole model into a planar homography H:

$$\begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & 0 & t_1 \\ r_{21} & r_{22} & 0 & t_2 \\ r_{31} & r_{32} & 0 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} \tag{2.3}$$

$$\Leftrightarrow \begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.4}$$

$$\Leftrightarrow \begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.5}$$

$$\Leftrightarrow \begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.6}$$

Here, H is a full-rank 3x3 matrix that describes the 2D to 2D projective transformation between the plane in the 3D world and the 2D image plane. Homography correlates the points, expressed in homogeneous coordinates, in the world plane $(x, y, 1)^T$ to their counterparts $(\lambda x', \lambda y', \lambda)^T$ in the image plane.

The homography matrix is especially useful when dealing with two images of the same planar scene taken from different viewpoints, allowing for the computation of the second image's point locations based on the points from the first image. The geometry of this situation is equivalent to the geometry of the FalKe system and can be visualized in Figure 2.3. The cameras **c** and **c'** represent two distinct image planes - the camera and the projector plane in the FalKe system - and π denotes the planar scene - the projecting surface in the

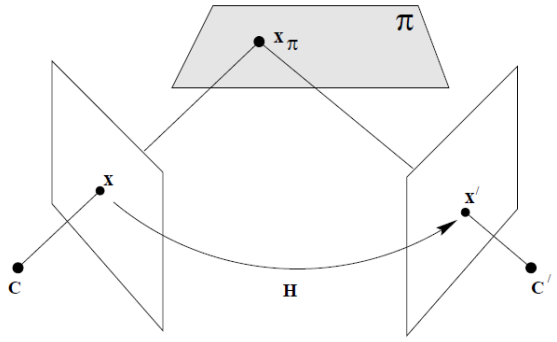


Figure 2.3: **The homography induced by a plane**, from [HZ03]. The plane π induces a homography H between the points in the plane **c** and the points in the plane **c'**.

FalKe system. A point x_π on the plane π is projected onto the image planes resulting in x and x'. The mapping from x to x' is the homography induced by plane π. The homography results from the combination of the two perspectivities: one from the world plane to the first image plane $x = H_{1\pi}x_\pi$ and another from the world plane to the second image plane $x' = H_{2\pi}x_\pi$. Their combination results in the homography matrix $x' = H_{2\pi}H_{1\pi}^{-1}x = Hx$. Essentially, the homography matrix maps points from one image to their counterparts in another, assuming both capture the same planar surface. This transformation adjusts for variations in viewpoint and orientation, correlating the images as different perspectives of that plane.

In practice, H can be computed from four point correspondences, with no three collinear on either plane. This can be done by using robust estimation techniques like the RANSAC algorithm [FB81] (see Section 2.1.5). In Python programming language, the Open Source Computer Vision Library (OpenCV) function findHomography() computes the homography matrix using RANSAC from at least four point correspondences, and warpPerspective() then applies this transformation matrix to images.

Homography is a type of projective transformation that maintains collinearity and incidence, meaning that points lying on a line before the transformation will still lie on a line after the transformation. However, homography does not necessarily preserve parallelism or the ratios of lengths along lines. In essence, while it keeps straight lines straight and maps lines to lines, it can change the appearance of shapes and angles, and the relative distances between points on a line may not be consistent after the transformation.

By applying the homography matrix, one can project the entire plane from one image to another, facilitating tasks such as image stitching, perspective correction, and object recognition, where planar surfaces are involved. Figure 2.4 shows an example of correcting perspective distortions with homography.

In the FalKe system, homography is utilized in the camera-calibration process (see Section 4.3) and in the reconstruction process to map the fragments onto the complete reference image (see Section 4.5).

2.1.4 Affine transformations

Affine transformations are a subgroup of projective transformations, distinguished by their capacity to map points from one 2D plane to another while preserving parallelism, ratios of lengths of parallel line segments and ratios of areas. While projective transformations can handle a wide range of changes, including perspective distortions, affine transformations are limited to transformations that preserve lines and parallelism (though not necessarily angles

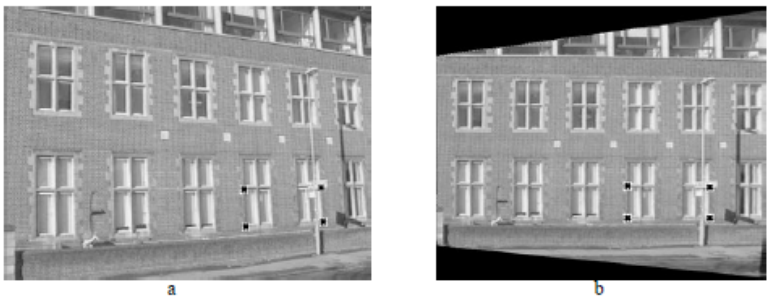


Figure 2.4: **Removing perspective distortion with homography**, from [HZ03]. (a) The original image with perspective distortion. (b) The image after applying homography from the four point correspondences.

and distances).

Mathematically, an affine transformation in homogeneous coordinates is expressed as:

$$\begin{pmatrix} \lambda x' \\ \lambda y' \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \vec{t} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.7}$$

In this representation, the 2x3 matrix represents the affine transformation, which consists of a 2x2 part A for linear transformations (like scaling and rotation) and a translation vector $\vec{t}$.

The linear transformation component A can always be decomposed into a combination of rotations and non-isotropic scalings, where scaling occurs in orthogonal directions at a particular angle. It can be expressed as:

$$A = R(\theta)R(-\phi)DR(\phi) \text{ with } D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{2.8}$$

where $R(\theta)$ and $R(\phi)$ are rotation matrices by angles θ and ϕ respectively. D is a diagonal matrix representing non-isotropic scaling by factors λ_1 along the x-axis and λ_2 along y-axis, allowing for non-uniform scaling. This decomposition illustrates that affine transformations are a sequence of rotating by angle ϕ, scaling in orthogonal directions by factors λ_1 and λ_2, rotating back by ϕ, and then rotating by a final angle θ. In the context of this decomposition, it is important to note that parallel lines remain parallel after an affine transformation. This means that the orientation of lines after transformation is determined solely by their initial orientation and not their position. Furthermore, affinities are classified as orientation-preserving or reversing based on the sign of the determinant of A. A positive determinant signifies that the transformation preserves the orientation of figures, while a negative determinant indicates that the transformation reverses their orientation, akin to a reflection.

The 2x3 affine transformation matrix can be computed from three point correspondences. In Python programming language, OpenCV functions like estimateAffinePartial2D() and warpAffine() are used to compute and apply affine transformations. The former computes the transformation matrix given at least three point correspondences, utilizing RANSAC or Least Median of Squares for robust estimation, while the latter applies the transformation to images.

Affine transformations, unlike homographies, don't account for perspective but are ideal for preserving parallelism and proportions in flat or slightly curved scenes, where the object or scene of interest lies approximately in a single plane with minimal perspective distortion. They are computationally simpler and often used in image registration tasks to align two images [LDZ+10].

In the FalKe system affine transformations are used as an alternative to homography in the reconstruction process to map the fragments onto the complete reference image (see Section 4.5).

2.1.5 RANSAC estimation

RANSAC is a robust estimation algorithm commonly used in computer vision and image processing to estimate model parameters from a set of observed data points containing outliers, firstly introduced by Fischler and Bolles [FB81] for 2D detection. RANSAC can be used to robustly estimate a projective transformation from a set of point correspondences and it typically requires a minimum of four point correspondences to compute a homography and three to compute an affine transformation [HZ03]. Notably, while this minimum is required, adding more correspondences can significantly enhance the robustness of the estimation.

The steps for using RANSAC to compute the transformation matrix typically involve the following [HZ03]:

1. **Random sample selection**. RANSAC selects a minimal subset of point correspondences (usually three or four depending on the transformation) from the given set of correspondences.

2. **Matrix estimation**. A homography or affine transformation matrix is computed using the selected subset of correspondences. This can be done using methods such as Direct Linear Transform (DLT) or other robust estimation techniques.

3. **Inlier counting**. The computed homography or affine transformation is then used to test the remaining correspondences, and those that are consistent with the estimated model within a certain threshold are considered as inliers.

4. **Iterative refinement**. The above steps are repeated for a number of iterations, and the homography with the largest number of inliers is selected as the best estimate.

RANSAC is particularly useful in the presence of outliers and noise in the data, as it focuses on finding a model that is consistent with the largest subset of the data while ignoring outliers. In the case of computing a homography matrix or affine transformation matrix, RANSAC helps to robustly estimate the transformation despite the presence of noise and outliers in the point correspondences. For this reason, its role in the FalKe system is crucial, enabling the rejection of outliers and accurate estimation of transformation matrices from corresponding feature points.

2.2 Neural Networks in Computer Vision

The field of computer vision has experienced remarkable advancements through the integration of various deep learning architectures. This section explores a specific range of neural network designs that are pivotal in the learning-based local feature matching methods implemented and investigated in this thesis. It covers Convolutional Neural Networks (CNNs) — including Fully Convolutional Networks (FCNs) and Group-equivariant Convolutional Neural Networks (G-CNNs) — as well as Graph Neural Networks (GNNs). Each of these architectures offers a unique approach and significantly contributes to advancing the processing and understanding of visual data. The following discussion aims to provide an insightful perspective on how these diverse models enhance computer vision tasks, addressing complex challenges with innovative solutions.

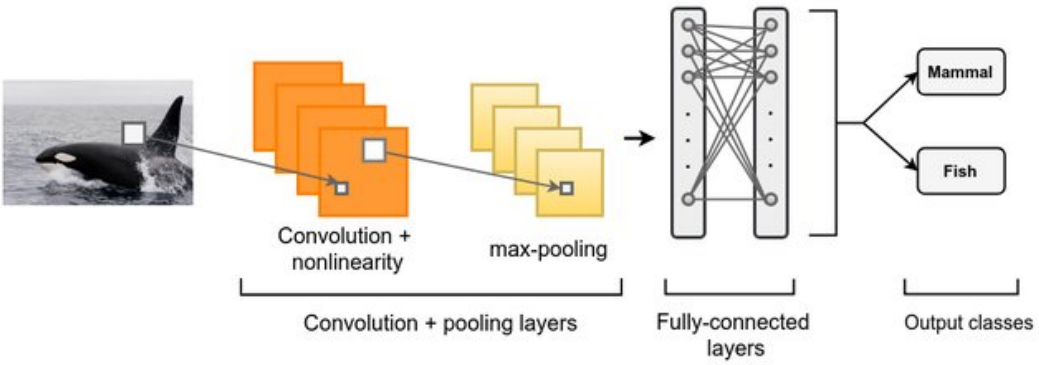


Figure 2.5: **Example of a CNN for image classification**, from [LEM+19].

2.2.1 Convolutional Neural Networks (CNNs)

CNNs are a pivotal category of deep learning models, exceptionally suited for various image processing tasks due to their capability to automatically learn spatial hierarchies of features. They are versatile and effective in a wide range of applications, from image classification [YXG+20, KSH12] and object detection [LAE+16, GZL+19, GDDM16] to more complex tasks like image segmentation [YY21, BKC17, ZSQ+17]. Their key characteristics include:

- **Layered architecture**. CNNs consist of an input layer, multiple hidden layers with neurons, and an output layer, each layer designed to process different aspects of the input data. The hidden layers are usually a combination of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 2.5. This structure ensures a flow of data from one layer to the next, where each layer's output becomes the input for the subsequent layer. The core building block of a CNN is the convolutional layer, which applies a series of filters to the input. These filters help the network focus on specific features in the image, like edges, textures, or shapes. Each filter produces a feature map that represents the presence of specific features in the input image. After computing the feature maps, a nonlinear activation function, like ReLu, sigmoid, or tanh, is applied. Following convolutional layers, pooling layers are often used to reduce the spatial dimensions (width and height) of the input volume for the next convolutional layer. Pooling helps in reducing the computational load, memory usage, and the number of parameters. It also provides a form of translation invariance. Towards the end of the network, CNNs typically have one or more fully connected layers. These layers use the high-level features learned by the convolutional and pooling layers to make a final classification decision. [GBC16, AZH+21, LEM+19]

- **Hierarchical feature representation**. CNNs are renowned for their ability to learn hierarchical representations of features. Starting with simple features such as edges and textures in the initial layers, they progressively move towards identifying more complex patterns. This hierarchical learning is crucial for interpreting both the finer details and the broader context within images [ZF14].

- **Local connectivity and shared weights**. CNNs exploit spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers. The shared weights reduce the number of parameters, enabling the network to be deeper with fewer parameters. [GBC16]

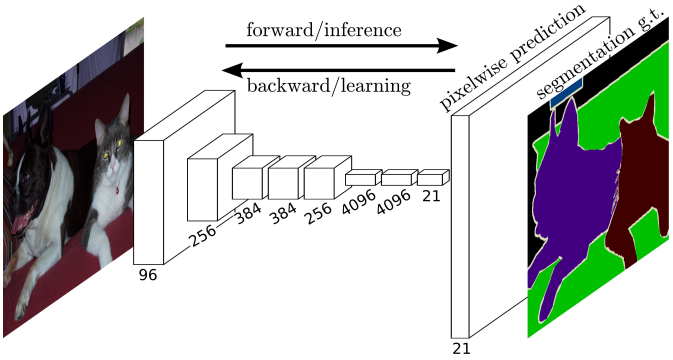


Figure 2.6: **Example of a FCN that performs semantic segmentation**, from [LSD15].

- **Learning through backpropagation**. Training a CNN involves adjusting the connections (weights) between neurons to minimize prediction error. This process is achieved through backpropagation and various optimization techniques, enabling the network to effectively learn from the input data and improve its accuracy. [GBC16, AZH+21]

Fully Convolutional Networks (FCNs)

A Fully Convolutional Network (FCN) is a type of neural network architecture that consists entirely of convolutional layers without any fully connected layers at the end [DV18]. FCNs are widely used in complex tasks such as image segmentation [BKC17, ZSQ+17], object detection [WSL+21], and other tasks, like keypoints detection and description [DMR18], where the output is a spatial map rather than a single label or prediction for the entire input. Figure 2.6 depicts a FCN applying semantic segmentation, where each pixel or small pixel area in the input image is methodically labeled.

The key characteristics of a FCN include:

- **Spatial preservation**. FCNs preserve the spatial dimensions of the input throughout the network by using convolutional and pooling operations without reducing the spatial resolution excessively. This is essential for accurately mapping the features back to the original image space. [LSD15]
- **End-to-end processing**. FCNs can accept input images of arbitrary size and produce output feature maps of corresponding spatial dimensions, enabling them to process full-sized images in a single forward pass [LSD15]. These maps are crucial for tasks requiring pixel-wise predictions such as dense and precise keypoint detection [DMR18].
- **Semantic information**. By utilizing multiple convolutional layers with varying receptive fields, FCNs capture semantic information at different spatial scales. This allows them to learn hierarchical representations of the input data, essential for understanding complex scenes [LSD15, BKC17].
- **Transposed convolutions**. To upsample feature maps and generate output maps with the same spatial dimensions as the input, FCNs often employ transposed convolutions (also known as deconvolutions or upsampling) [LSD15]. This technique is pivotal in reconstructing detailed information from compressed feature representations [DV18].

SuperPoint [DMR18], detailed in Section 3.2.1, is an example of using FCNs for local feature detection and description.

Group Equivariant Convolutional Networks (G-CNNs)

G-CNNs are a class of CNNs, designed to be equivariant to transformations from a specific group. Equivariance here means that if the input data is transformed according to a group operation (like rotation, translation, etc.), the output of the network transforms in a predictable way according to the same group operation. G-CNNs are an extension of traditional CNNs, which are inherently equivariant to translations but not necessarily to other transformations like rotations or scaling. [CW16a]

Steerable CNNs are a specialized form of G-CNNs designed to handle data with inherent symmetries, particularly rotational [CW16b, WGW+18, WHS18]. The term "steerable" refers to the property that the network's filters can be "steered" or adjusted according to the symmetries (like rotation) in the input data. This is achieved through the use of representation theory of groups, allowing feature vectors within feature fields to correspond to specific group representations [FH04]. This makes them capable of processing features aligned with the input data's symmetries. Their key characteristics can be summarised in:

- **Equivariance to rotation**. Steerable CNNs are designed to be equivariant to transformations from a specific group, such as the SO(2) Group, the Special Orthogonal Group in 2 dimensions, which represents rotations in a plane. This means that the network's responses to input data transform in a predictable way when the input data itself undergoes specific transformations. This property is crucial for tasks where orientation and position of objects vary. By being sensitive to such transformations, equivariant CNNs can learn more generalizable and robust feature representations. By incorporating equivariance to rotations, steerable CNNs can effectively handle images with varying orientations and extract rotation-invariant features. [WHS18, CW16b]
- **Feature fields and channels**. In Steerable CNNs, an image or input data is represented as a collection of feature fields. Each field consists of several channels, with each channel representing different orientations or features. For instance, a channel might be designed to respond maximally to edges in the image oriented at a specific angle. [CW16b]
- **Use of representation theory**. The network employs representation theory of groups to interpret feature vectors within these feature fields. This allows feature vectors to correspond to specific group representations, aligning with the symmetries in the input data [FH04].
- **Enhanced weight sharing**. Steerable CNNs exhibit enhanced weight sharing compared to ordinary convolutions. This feature allows the network to use the same filter for recognizing features across different orientations, reducing computational complexity and improving learning efficiency in processing rotationally varied data. [CW16b]

E(2) Equivariant Steerable CNNs extend the capabilities of Steerable CNNs to specifically address the Euclidean group E(2), making them equivariant to combinations of translations and rotations in 2D spaces [WC21]. The network's architecture and operations are designed to preserve the symmetries and invariances associated with translations, rotations and reflections in 2D space, ensuring that the network's responses to input data are consistent and predictable under these transformations. Their ability to detect and represent translation and rotation invariant features renders them highly effective for tasks in fields such as aerial imagery, biomedical imaging, and robotics, where processing images and planar representations that exhibit diverse spatial aspects is crucial [WC21]. A subgroup of

E(2) equivariant steerable CNNs are the ones equivariant specifically to subgroups of the Special Euclidean Group SE(2), the group that includes translations and rotations but not reflections in 2D space [WC21]. Unlike E(2) equivariant models, they specifically emphasize rotational and translational invariances, optimizing feature detection and representation in scenarios where these specific transformations are prevalent. These networks excel in maintaining consistency in feature detection and representation despite rotational variations in the input [BK22], enhancing the model's performance and generalizability in rotationally diverse environments. SE2-LoFTR [BK22], detailed in Section 3.2.2 is an example of using SE(2) Equivariant Steerable CNNs for local feature matching.

2.2.2 Graph Neural Networks (GNNs)

GNNs are specialized neural network models adept at processing graph-structured data, where data points and their relationships are represented as nodes and edges in a graph [SGT+09]. Unlike traditional neural networks that handle grid-like data (such as images or sequences), GNNs are particularly effective with complex, non-grid structures found in social networks, molecular structures and knowledge graphs [WPC+21]. Their proficiency lies in interpreting relational data and understanding the intricate connections within it [SGT+09]. This ability is vital for tasks like high-precision feature matching in images, where discerning the spatial relationships among keypoints is essential [SDMR20, JSTL19].

Key characteristics of GNNs include:

- **Neighborhood aggregation**. GNNs aggregate and process information from a node's immediate neighbors. This aggregation function, which can be a sum, mean, or a more complex operation, allows GNNs to capture the local structure around each node, enabling the network to understand and utilize the connectivity patterns within the graph.[SGT+09]
- **Recursive feature propagation**. GNNs iteratively update node representations by combining their own features with aggregated features from their neighbors. This recursive process, often realized through layers in the network, allows for the integration of information from larger graph neighborhoods over successive iterations. This allows to capture the relational dependencies within the graph. [HYL18]
- **Weight sharing across nodes**. Unlike traditional neural networks, where different parts of the input (e.g., pixels in an image) can have different weights, GNNs share weights across all nodes in the graph. This characteristic makes GNNs particularly efficient in parameter usage and helps in generalizing across different graph structures. [WPC+21]
- **Inductive learning abilities**. GNNs are capable of inductive learning, meaning they can generalize to unseen nodes or entirely new graphs after training. This is crucial for tasks where the model encounters new, unseen data after its initial training phase. [HYL18]

Attention-based GNNs are specifically designed to leverage attention mechanisms for reasoning about the relationships between elements in a graph, allowing for more complex and flexible reasoning about the data. Attention mechanisms have been widely used in GNNs to perform both global and data-dependent local aggregation by focusing on specific elements and attributes, making them more flexible. SuperGlue [SDMR20], detailed in Section 3.2.1, is an example of using attentional GNNs for local feature matching.

3 State of the art

The FalKe system's approach to reconstructing documents is fundamentally based on local feature matching, a process essential for determining the fragments' location within the complete reference document. This chapter commences with the research done on related work on the automatic fragment reassembly problem. It then delves into the state of the art local feature matching methods, from traditional detector-based ones to the newest detector-free ones, focusing on the methods that have been implemented in this thesis and investigated in Section 4.5.2 of the next chapter.

3.1 Computer-aided fragment reassembly

The task of reconstructing fragmented objects is comparable to jigsaw puzzle assembly. Automatic solutions in this field have been extensively researched, particularly for scenarios where the original image is unknown. These solutions generally fall into three categories: methods that rely solely on the contours of fragments, methods utilizing color information, and hybrid approaches that combine both. These complex techniques range from contour-based matching in archaeological artifacts to color analysis in document reconstruction. The diversity in strategies underscores the complexity of the reassembly problem when the original images are absent. For a more comprehensive understanding, a detailed overview of these methods is available in the Annex 1.

In scenarios where the original image is known, as in the FalKe system examined in this thesis, solving the puzzle becomes simpler. This approach involves comparing the features of the puzzle pieces directly with the original image, rather than the more complex task of calculating similarities between two puzzle pieces. For instance, Li et al. [LZZC14] used the Scale Invariant Feature Transform (SIFT)[Low04] algorithm to match puzzle pieces with their corresponding areas in the original image. This method involved aligning fragments using RANSAC, then grouping them through agglomerative clustering. This technique successfully reconstructed banknotes from many fragments. In [MLS23] the authors developed a method for assembling 2D pictorial jigsaw puzzles using a robotic arm. Also this approach uses SIFT to detect feature points in both the puzzle pieces and the original image and from the found matches computes a transformation matrix that indicates rotation angles between original image and pieces. To remove mismatched pairs the RANSAC algorithm is applied. This approach is robust but its effectiveness diminishes in images with extensive homogeneous areas. The FalKe system adopts a similar approach for document reconstruction, utilizing local feature matching to pinpoint each fragment's specific location in the original image and applying RANSAC for precise alignment in these identified areas.

3.2 Local feature matching

The primary objective of matching (sub-)pixels in two images is to use these correspondences to estimate a transformation between the images [HZ03]. This principle, addressed in Section 2.1.5, is fundamental for the FalKe system, which relies on the found matches to estimate a transformation that serves to accurately map fragments to their precise location in the

complete reference image. The reliability of these matching methods, especially in terms of rotation and scale invariance as well as robustness to variations in texture and lighting conditions, is thus of paramount importance for accurate reconstruction.

The local feature matching process typically follows a structured, four-step pipeline [Sze11]:

1. **Detection**. This involves identifying locations of keypoints in both images. Keypoints are distinctive features in the images, such as edges, corners, or specific texture patterns.

2. **Description**. This phase focuses on encoding the image content surrounding these keypoints into descriptors - also called features - forming the basis for comparison.

3. **Matching**. In this stage, the previously obtained descriptors are compared to determine corresponding locations across the images.

4. **Filtering**. The final step entails refining these matches by eliminating outliers or incorrect matches, ensuring the reliability of the correspondences. Here, RANSAC [FB81] is the most commonly employed method.

While detection and description have traditionally been closely intertwined, matching and filtering have been more independent. Nevertheless, the advent of neural network-based methods has led to a trend of integrating these steps into a cohesive, end-to-end process. Another research trajectory – detector-free methods – forgoes the detection stage altogether. Instead, it relies on densely describing every pixel (or their downsampled versions) in both images.

In the following sections, key detector-based and detector-free local feature matching methods are concisely overviewed, with an emphasis on those implemented and thoroughly examined in this thesis.

3.2.1 Detector-based local feature matching methods

Traditional methods

Building upon the foundational pipeline previously illustrated, detector-based methods have been central in local feature matching, especially before the advent of deep learning.

SIFT [Low04], a highly successful hand-crafted local feature detector and descriptor, emerged as a cornerstone in various computer vision tasks due to its robustness in handling scale and rotation changes. Following SIFT, Oriented FAST and Rotated BRIEF (ORB) [RRKB11] was developed as an efficient alternative. ORB utilizes Features from Accelerated Segment Test (FAST) [RD06] for speedy keypoint detection and a modified, rotation-aware version of Binary Robust Independent Elementary Features (BRIEF) [CLSF10] for its descriptors. While more computationally efficient than SIFT, ORB compromises slightly on scale, rotation invariance and is more sensitive to the noise in an image. Several other traditional features are widely recognized for their distinct strengths: SURF [BETV08] is valued for its computational efficiency; ASIFT [YM11] is known for its affine invariance; and KAZE [ABD12] is notable for its ability to preserve object boundaries. A comparative analysis of SIFT, SURF, KAZE, AKAZE [ANB13], ORB, and BRISK [LCS11] has found that SIFT and BRISK are the algorithms most invariant to scale, rotation and viewpoint changes [TS18].

Owing to its exceptional robustness to scale and rotation, surpassing not only traditional methods but also many deep-learning-based approaches, SIFT was selected for implementation and investigation in this thesis and its functionality is detailed subsequently.

Deep learning-based methods

Deep learning has revolutionized the field of local feature matching, offering significant advancements in finding correspondences under difficult imagining conditions such as poor lighting, varied textures, scale and perspective differences. The following methods contributed to the field, highlighting a trend towards integrating deep learning with detector-based designs.

Learned Invariant Feature Transform (LIFT) [YTLF16], introduced as the first end-to-end network, implemented a comprehensive pipeline that includes detection, orientation estimation, and feature description. Its rotation invariant features make it adept at matching features between images captured from different angles or distances. Following LIFT the field witnessed a surge in the development of learning-based methods tailored for feature matching. Building on the groundwork laid by the MagicPoint detector [DMR17], SuperPoint [DMR18] represents a significant advancement, combining keypoint detection and feature description within a single neural network. Its real-time processing capability and self-supervised training method enable effective handling of scale and rotational changes. Another relevant method is Grid-based Motion Statistics (GMS) [BLM+17], which focuses on motion smoothness to assess match quality. By translating high match numbers in a region into a measure of match quality, GMS effectively handles low-texture and blurred scenarios, often associated with poor lighting. This focus on match quality over quantity signifies a major shift in matching strategies. Following this line R2D2 [RWS+19] predicts the distinctiveness of local descriptors, thus avoiding ambiguous areas and improving keypoint detection and description reliability. ASLFeat [LZB+20] takes advantage of intrinsic feature hierarchies to improve the spatial resolution and detail of keypoints, leading to more precise keypoints localization.

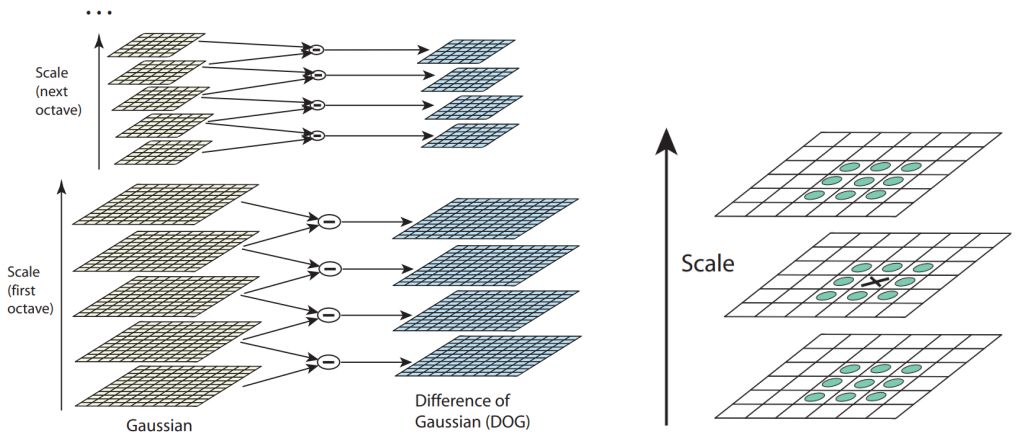
In this thesis, SuperPoint was the only method implemented in the initial state of the FalKe system. The decision to retain and enhance it was motivated by its real-time processing capabilities, efficiency, and the robustness of its feature descriptors. Further details on its functionality are provided in the following sections.

Matching and outlier filtering

In feature matching, traditional methods like Nearest Neighbor Matching (NN) (Brute-Force (BF) or Fast Library for Approximate Nearest Neighbors (FLANN)) are still very used due to their simplicity, lower computational requirements and broad compatibility with various feature descriptors [Sze11]. BF matching examines each descriptor against all others to find the best match [Sze11], while FLANN [ML09] uses an approximate nearest neighbor search to speed up the matching process. These methods typically pair with RANSAC [FB81] for outlier detection and filtering.

A significant leap forward are learning-based methods, like SuperGlue [SDMR20], which concurrently matches and filters outliers. SuperGlue excels in establishing highly accurate and robust feature correspondences, even in challenging scenarios with significant changes in viewpoint or illumination, which is why it was implemented alongside SuperPoint in this thesis. Its functionality is explored subsequently.

Furthermore, deep learning has also refined the efficiency and accuracy in the last step of the feature matching pipeline: outlier filtering and estimating transformations, like homographies. Innovations such as Attentive Context Networks (ACNe) [SJT+21], T-Net [ZXZ+21], and the rotation invariant ZZ-Net [BKF22] exemplify this progress, showcasing a remarkable evolution in feature matching and outlier detection.



(a) Difference of Gaussians (DoG), from [Low04] (b) Feature detection, from [Low04]

Figure 3.1: **SIFT detector using the DoG**. (a)Within each scale space octave, the initial image is convolved with Gaussians to produce scale space images (*Left*), and adjacent ones are subtracted to form DoG images (*Right*). The image is then halved in size for the next octave and the process is repeated. (b) Each pixel (marked with X) is compared with its 8 surrounding pixels and the 9 pixels in the scale above and below it (marked with circles). If the recorded value is greater than the values of the surrounding pixels, the point is recognised as a keypoint.

SIFT

SIFT [Low04] is a widely used algorithm in computer vision for detecting and describing local features in images. It works by identifying keypoints that are invariant to scale and rotation, then generating a descriptor based on the gradients around these keypoints. It operates in four main stages:

1. **Scale-space extrema detection**. To identify keypoints, the SIFT algorithm uses a scale-space representation of the image, where the image is convolved with a series of Gaussian filters (blur filters) at different scales. This process results in a pyramid of images at different levels of blur and resolution. It can be mathematically represented as:
$$L(x, y, \sigma) = (G(x, y, \sigma) * I(x, y) \tag{3.1}$$
where $L(x, y, \sigma)$ is the scale-space representation of the image, $G(x, y, \sigma)$ is the Gaussian function at scale σ, and $I(x, y)$ is the original image. Potential keypoints are then identified as local extrema at each level in the Difference-of-Gaussian (DoG) scale space (see Figure 3.1b), which is obtained by subtracting adjacent levels of the Gaussian pyramid, as shown in Figure 3.1a. Mathematically, the DoG function can be defined as:
$$D(x, y, \sigma) = (G(x, y, k\sigma) * I(x, y)) - (G(x, y, \sigma) * I(x, y)) \tag{3.2}$$
where $D(x, y, \sigma)$ is the DoG function, k is the scale factor, and $G(x, y, \sigma)$ is the Gaussian function at scale σ.

2. **Keypoint localization**. Following the detection of potential keypoints, the algorithm refines their positions and scales by fitting a 3D quadratic function around each keypoint in the DoG scale space. This enhances the precision in localizing extrema and contributes to scale invariance.

3. **Orientation assignment**. To ensure rotation invariance, each keypoint is assigned a dominant orientation. This is computed by creating a histogram of gradient orientations

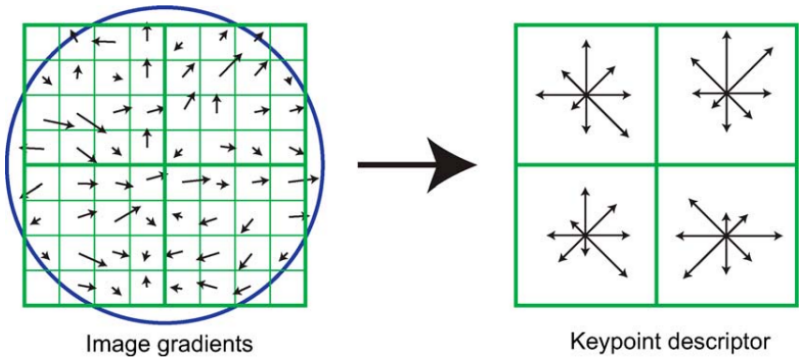


Figure 3.2: **SIFT descriptor**, from [Low04]. *Left*: Gradient magnitudes and orientations at sample points around the keypoint are computed and Gaussian-weighted, as the circle overlay shows. *Right*: These gradients are accumulated into orientation histograms over 4x4 subregions, with arrow lengths indicating the sum of nearby gradient magnitudes.

within the keypoint's local neighborhood, based on the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$. The peak of this histogram determines the keypoint's orientation.

4. **Descriptor generation**. The final stage involves generating a unique descriptor for each keypoint. This descriptor captures the gradient information (magnitude and orientation) at each pixel in a region around the keypoint. These gradients are then weighted by a Gaussian window to give more weight to pixels closer to the keypoint. The weighted gradients are then accumulated into orientation histograms over 4x4 subregions, resulting in a 128-dimensional vector that represents the keypoint descriptor, as shown in Figure 3.2. This descriptor is crafted to be robust against changes in illumination, rotation, and scale.

The SIFT algorithm employs Nearest Neighbor matching (NN) for matching descriptors between images, using the Euclidean distance as the metric. To mitigate the risk of ambiguous matches, a *Ratio test* is applied, comparing the distance between the closest and second-closest matches. Descriptors failing this test are discarded. Subsequent refinement, possibly using RANSAC, eliminates outliers and determines the transformation between image pairs.

SuperPoint (SP)

The SuperPoint [DMR18] algorithm is a self-supervised framework designed for training keypoint detectors and descriptors. It operates on a full-sized image and computes pixel-level keypoint locations and associated descriptors in a single forward pass thanks to its Fully Convolutional Network (FCN) structure. This enables real-time performance and efficient computation of interest points and descriptors. FCNs are detailed in section 2.2.1. Here's a technical explanation of how SuperPoint works:

- **Interest point pre-training with Synthetic Shapes**. SuperPoint's initial training occurs on a dataset of simple geometric shapes, called *Synthetic Shapes*, where it uses simple geometric shapes to develop a base detector called MagicPoint [DMR17]. During this phase, the network learns to extract hierarchical features from the image, enabling it to recognize interest points. While MagicPoint performs well on Synthetic Shapes, it doesn't generalize too well on real images.

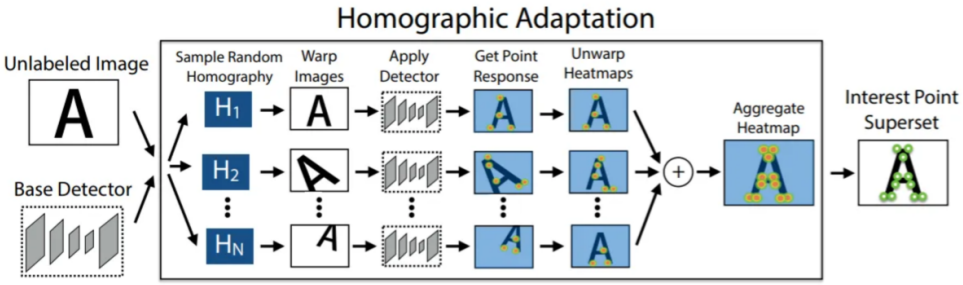


Figure 3.3: **Homographic Adaptation process**, from [DMR18]. The original image undergoes multiple warping transformations through random homographies. The MagicPoint detector is then applied to these warped images to identify interest points. After detection, these points are mapped back to their positions in the original image.

- **Interest point self-labeling with Homographic Adaptation**. SuperPoint's performance on real images is enhanced by a technique known as *Homographic Adaptation*. This multi-scale, multi-transform method, shown in Figure 3.3, alters the image's perspective and scale. It involves detecting interest points on these modified images and then mapping these points back to the original image. This self-labeling mechanism is a cornerstone of SuperPoint's self-supervised learning, enabling the model to autonomously generate training data, thereby reducing reliance on manually annotated datasets. It significantly improves MagicPoint's detection of interest points in different scales, viewpoints and varied real-world textures and patterns, enhancing its ability to create descriptive representations of these points.

- **FCN architecture**. SuperPoint's FCN architecture involves a shared encoder to simplify the image, then splits into two parts for detection and description of points. This unique design allows for efficient processing and sharing of computational tasks: during joint training the model is trained to perform both interest point detection and descriptor generation at the same time.

- **Efficient detection and description**. In detecting points, SuperPoint calculates the probability of each pixel being an interest point and concurrently generates a grid of high-dimensional descriptors. These descriptors, optimized for both efficiency and practicality, capture the local texture and appearance around each point. This process, facilitated by the neural network's convolutional layers, effectively combines point detection with detailed description, ensuring a streamlined operation.

SuperPoint is effective in various computer vision tasks, particularly in geometric applications like Simultaneous Localization and Mapping (SLAM) and object tracking. While it is proficient at keypoint detection, its detected features are not inherently rotation or scale-invariant. As depicted in Figure 3.4, SuperPoint demonstrates strong performance under illumination changes, outperforming LIFT, ORB, and SIFT. However, its performance significantly declines with rotations exceeding 45 degrees, where SIFT and ORB perform better. In fact SuperPoint's descriptors are inherently invariant to rotations up to 45 degrees, and while training can extend this range, complete invariance might still require additional methods beyond the training process. It's also crucial to recognize the inherent trade-off between invariance and discriminativeness. As noted in [PLOP20], a descriptor optimized for rotation invariance may sacrifice some level of illumination invariance, particularly within a fixed network capacity.

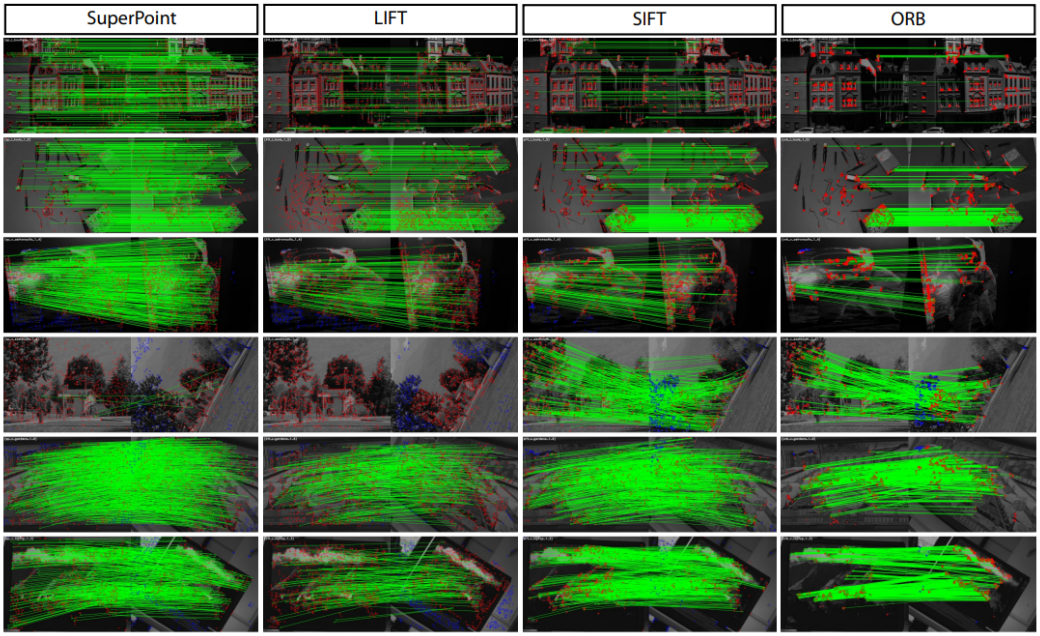


Figure 3.4: **Matching results from SuperPoint, LIFT, SIFT and ORB**, from [DMR18]. The green lines show correct correspondences. Notably LIFT and SuperPoint fail in handling extreme in-plane rotations (4th row), while SIFT and ORB excel.

SuperGlue (SG)

SuperGlue [SDMR20] is an advanced real-time local feature matcher leveraging Graph Neural Networks (GNNs) to conduct context aggregation, matching, and filtering in a cohesive, end-to-end architecture. For a foundational understanding of GNNs, refer to 2.2.2. The SuperGlue's architecture, shown in Figure 3.5, has two main components: the *Attentional Graph Neural Network* and the *Optimal Matching Layer*. The Attentional Graph Neural Network is responsible for the feature encoding and the context aggregation conducted through self-attention and cross-attention mechanisms; while the Optimal Matching layer solves the optimal transport problem to estimate the best matches (assignments) between the two sets of features.

- **Attentional Graph Neural Network**. This component is crucial for processing the input local features from two images. It constructs fully connected graphs for each set of features where nodes represent keypoints and edges represent potential correspondences between the features in the two sets. Each keypoint is encoded with its visual descriptor into a unified vector. The encoding captures both the position and appearance information. Within this network, both self-attention and cross-attention mechanisms are applied. Self-attention enhances the representation of each keypoint by aggregating contextual information from other keypoints within the same image. Cross-attention, on the other hand, allows keypoints from one image to interact with those from another, facilitating the identification of potential matches. The cross-attention layer is inspired by the way humans look back-and-forth when matching images. The output of this component is a set of feature representations that capture both appearance and keypoint location. Mathematically, the attentional GNN can be represented as follows:

$$f_i = g(h(p_i, d_i)) \tag{3.3}$$

  where p_i is the position of the i-th keypoint, d_i is its visual descriptor, h is the keypoint

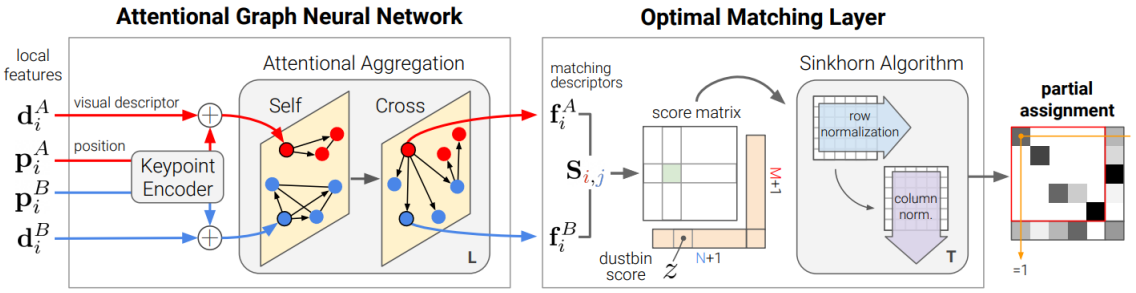


Figure 3.5: **SuperGlue's architecture**, from [SDMR20]. The Attentional Graph Neural Network (*Left*) utilizes a keypoint encoder to integrate keypoint positions p and visual descriptors d into a composite vector. This vector undergoes refinement through alternating self- and cross-attention layers, applied repeatedly for L iterations, to develop advanced representations f. The Optimal Matching Layer (*Right*) generates an MxN score matrix. This matrix is augmented with dustbins and the Sinkhorn algorithm is applied over T iterations to achieve the optimal partial assignment.

encoder that maps p_i and d_i into a single vector, g is the alternating self- and cross-attention layers that create more powerful representations, and f_i is the output feature representation of the i-th keypoint.

- **Optimal Matching Layer**. This layer computes a similarity score matrix, where each element represents the similarity between keypoints from the two images:

$$S_{ij} = f_i^T f_j \tag{3.4}$$

where S_{ij} is the score between the i-th and j-th keypoints, and f_i and f_j are their feature representations. It incorporates *dustbins* to account for keypoints without corresponding matches, addressing the challenge of non-matchable points. The layer then applies the Sinkhorn algorithm, a differentiable method for solving optimal transport problems, to iteratively refine the score matrix. This process results in a partial assignment matrix that indicates the most probable matches between keypoints across the two images. Based on the partial assignment matrix, SuperGlue identifies the final matches between keypoints across the two images.

SuperGlue refines its parameters, including those in the attention mechanism and feature encoding process, during training to improve keypoint matching accuracy in diverse scenarios. Although robust to lighting changes, occlusions, and textures, it struggles in handling significant rotations (over 45 degrees) and scalings. These limitations stem from the initial feature descriptors' quality, which ideally should be rotation and scale-invariant, and the training data diversity. Extensive training across various orientations and scales can enhance its ability to generalize to new images with similar variations.

SuperGlue's real-time performance and versatility with both classical and learned features make it ideal for computer vision tasks like 3D reconstruction, SLAM and pose estimation, outperforming traditional methods and excelling in complex, real-world environments [SDMR20]. Figure 3.6 demonstrates that SuperGlue, when combined with SuperPoint as a keypoint detector, surpasses the performance of SuperPoint with Nearest Neighbor Matcher in both outdoor and indoor settings.

Despite its innovations, SuperGlue shares the common limitation of detector dependency, especially in identifying repeatable keypoints in non-distinct areas, characterized by low textures such as smooth surfaces or uniform backgrounds.

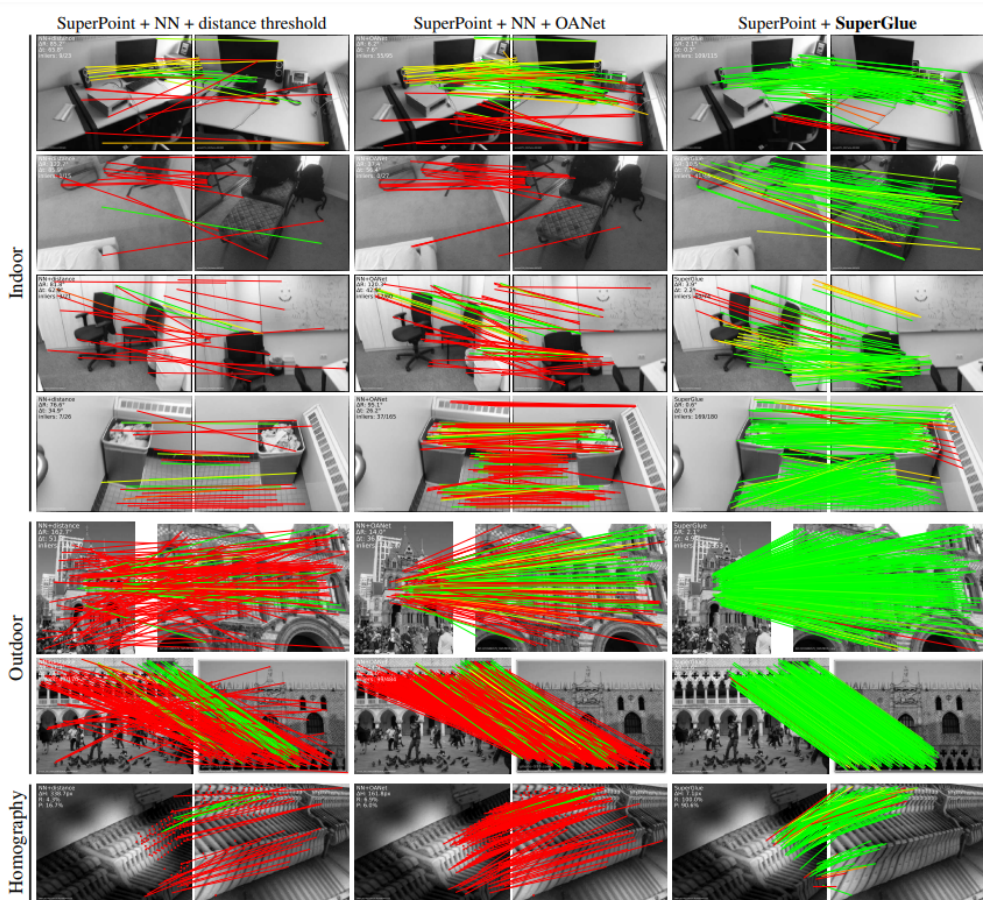


Figure 3.6: **Performance of SuperPoint+NN and SuperPoint+SuperGlue in varied conditions**, from [SDMR20]. SuperPoint+SuperGlue consistently outperforms SuperPoint+Nearest Neighbor (NN) matcher, with either handcrafted or learned outlier detectors, in accuracy (green lines) and fewer mismatches (red lines). Its superior performance, handling repeated textures, major viewpoint shifts, and diverse illumination, is clear in both indoor and outdoor settings.

3.2.2 Detector-free local feature matching methods

Detector-free local feature matching methods have revolutionized computer vision by offering a unified approach for feature description and matching, eliminating the need for separate detection. Unlike detector-based methods that often rely on sparse, keypoint-based features, these methods utilize dense image descriptions, providing a comprehensive understanding of the entire image space. This approach is crucial in capturing complex image contexts, making these methods highly effective in challenging scenarios such as non-distinct low-texture areas and repetitive patterns. Initially, detector-free matching faced challenges in robust feature matching, leading to lesser popularity compared to detector-based methods. However, the advent of advanced deep neural networks has significantly boosted their performance.

Presently, these methods can be roughly categorized into cost volume-based and transformer-based approaches. Cost volume-based methods [RCA+18, LHLP20, MTS+19, TDT20] utilize correlation layers to assess feature similarities, while transformer-based methods [SSW+21, JTH+21, CLZ+22] employ cross attention to direct the search and refinement of feature correspondences.

Cost volume-based methods, such as NCNet [RCA+18], use end-to-end learning to create and regularize 4D cost volumes. This allows for the efficient enumeration and refinement of potential matches between images using advanced convolution techniques, making them ideal for scenarios with dense correspondences. DRC-Net [LHLP20], another example of this category, employs a coarse-to-fine strategy, starting with broad correlations from coarser feature maps and progressively refining these to detailed, pixel-level correspondences.

Transformer-based methods have made notable advancements in feature matching through innovations in attention mechanisms. Methods such as LoFTR [SSW+21] and COTR [JTH+21] utilize self- and cross-attention mechanisms for refined feature matching, achieving sub-pixel accuracy. LoFTR uses a transformer for initial broad pixel-wise dense matches, later refining them for higher precision, particularly effective in weakly textured areas. Conversely, COTR begins with image downsampling via a CNN, followed by transformer-based match refinement. SE2-LoFTR [BK22] addresses the challenge of achieving rotation invariant features with CNNs by replacing the CNN backbone of LoFTR with steerable CNNs. The robustness and versatility of SE2-LoFTR across a wide spectrum of imaging conditions render it particularly well-suited for the challenges addressed in the FalKe system. Consequently, it was implemented in this thesis and plays a pivotal role in the investigation detailed in Section 4.5.2. Its functionality is subsequently detailed.

SE2-LoFTR

SE2-LoFTR [BK22], a sophisticated variant of the Local Feature TRansformer (LoFTR) [SSW+21] model, is specifically engineered to boost the accuracy and robustness of matching point correspondences between two images of the same scene, particularly under large rotational variances. Its defining characteristic is the rotation invariance of the extracted dense local features, achieved by replacing LoFTR's conventional CNN backbone with SE(2) Equivariant Steerable CNNs. As outlined in Section 2.2.1, these steerable CNNs exhibit equivariance under subgroups of SE(2) - the group encompassing rotations and translations. This attribute renders SE2-LoFTR highly effective in scenarios involving images captured from varying orientations.

SE2-LoFTR is available in three distinct variants: SE2-LoFTR-4*, SE2-LoFTR-4, and SE2-LoFTR-8*. Each variant is tailored to manage rotations differently, such as handling quarter and eighth rotations, thus adjusting the model's internal feature maps accordingly. This fine-tuning ensures proficient rotational handling while maintaining a balance between computational efficiency and model complexity.

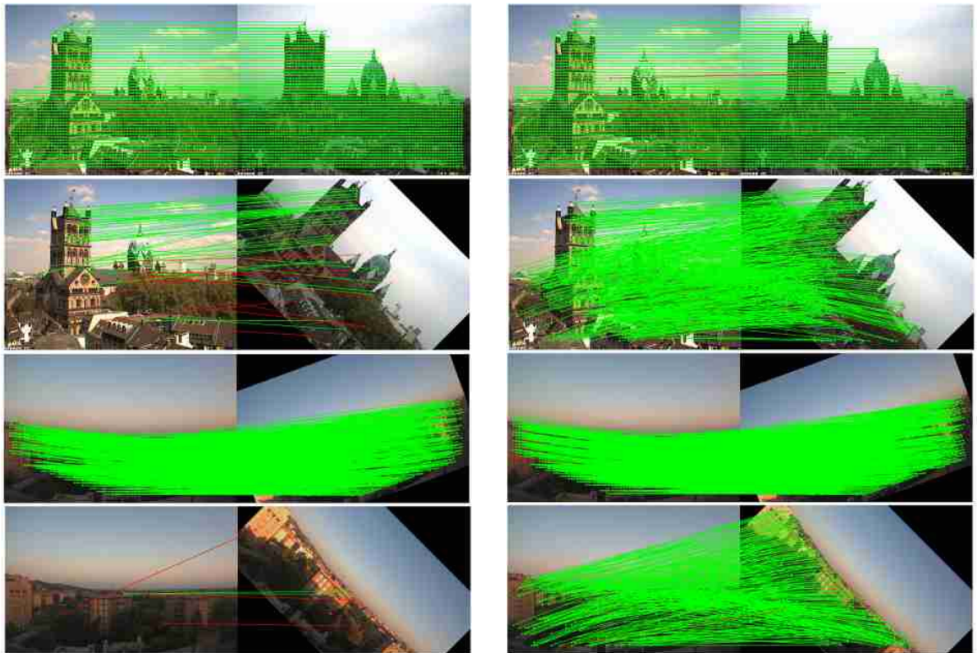


Figure 3.7: **Performance of LoFTR and SE2-LoFTR-4* in varied conditions**, from [BK22]. *Left*: LoFTR [SSW+21] excels in matching images with minor viewpoint and illumination changes, but struggles with large rotational changes. *Right*: SE2-LoFTR-4*, using a rotation-equivariant CNN, adeptly handles both significant rotational variations and minor viewpoint or illumination changes.

The operation of SE2-LoFTR comprises several stages:

1. **Feature extraction**. SE2-LoFTR departs from traditional feature extraction methods by utilizing SE(2)-equivariant steerable CNNs. The steerable CNNs transform input images into feature fields, where each pixel is associated with a unique feature vector, forming a dense feature map. These vectors, aligned with the SE(2) group representations, ensure both translation and rotation invariance. Consequently, as the image undergoes transformations like rotations, the feature vectors adapt, preserving accuracy and consistency in feature representation. The model extracts both coarse and fine features: coarse features, derived from early network layers, represent general, larger-scale patterns at about 1/8 of the image size, while fine features, extracted from deeper layers, capture more detailed aspects at approximately 1/2 of the image size. This dual-level feature extraction is vital for effective feature matching, providing a nuanced understanding of the image at varying resolutions.

2. **Transformation and LoFTR module**. The extracted coarse features are transformed, which includes being concatenated with positional encodings and processed through a specialized LoFTR module. This LoFTR module is a transformer - a special form of CNNs with attention mechanisms - specifically designed for the task of feature matching, preparing the coarse features for the next stage of the process.

3. **Coarse feature matching**. The transformed coarse features are then utilized in a matching module to establish potential correspondences between the images. This module computes a confidence value for each possible match among the coarse feature maps. The output consists of matching positions in the two images that are mutual, highly confident matches, surpassing a predefined threshold hyperparameter.

4. **Refining matches with fine features**. Subsequent to coarse matching, the algorithm

refines these matches using the fine features. Patches from the fine-level feature maps, located around the predicted coarse match areas, are extracted and processed through another LoFTR module. The output from this step is used to compute matches from each location in one image to a subpixel position in the other. This step is pivotal for achieving precise, subpixel-level matches. During training the model focuses on fine tuning both coarse match confidences and fine match accuracies.

SE2-LoFTR stands as a significant advancement in feature matching, particularly adept at handling scenarios where traditional keypoint detectors might underperform, such as images with repetitive patterns, minimal texture, or notable rotational differences. The evaluations performed in [BK22], underscore SE2-LoFTR's superior performance over LoFTR in handling images with extensive rotational differences, as shown in Figure 3.7. Notably, it achieves this heightened accuracy while preserving its effectiveness in matching non-rotated image pairs and managing to do so with minimal increase in computational demands.

4 Methods

This chapter commences with an analysis of the FalKe system's limitations and a detailed outline of the requirements devised to address these challenges. The solutions implemented are grounded in the theoretical fundamentals of projective transformations and neural networks detailed in Chapter 2, and informed by the research on the state of the art local feature matching methods discussed in Chapter 3. This chapter details these solutions and the methodologies employed during their implementation and investigation, providing a comprehensive overview of the approaches taken to enhance the FalKe system's functionality.

4.1 Requirements

This thesis arose from the need to address identified limitations within the FalKe system. Figure 4.1 displays all the components of the system, both hardware and software, highlighting in green the areas where significant limitations were identified. These components work in unison to assist users in reconstructing a document from the given fragments. The process followed by the system to reassemble n snippets is depicted in Figure 4.2. After digitising the fragments placed in the detection area, a segmentation algorithm, called REMBG[1], isolates each fragment by removing the background. The system then uses local feature matching methods to find the region in the reference document where the snippet belongs. If point correspondences are found, a transformation matrix (either homography or affine transformation) is computed using RANSAC [FB81] to determine the precise location and orientation of each fragment in the reference document. This crucial information is subsequently shown to the user in the reconstruction area, as seen in Figure 1.1.

The limitations identified in this process span both hardware and software components and the corresponding requirements formulated to addressed them, are detailed as follows:

- **Hardware layer**. The initial setup lacked a dedicated lighting system, which limited its effectiveness and adaptability to different environmental conditions. This absence was particularly detrimental in low-light scenarios, affecting the accuracy of document reconstruction. To address this, the integration of a uniform lighting system with adjustable intensity was recognized as crucial, enabling the system to adapt to various lighting environments and ensure consistent performance.

- **Acquisition and Digitisation**. Given the reliance of the FalKe system on reference documents, a key limitation of the initial system was its inability to manage more than one reference. Users had to manually insert a reference document image each time a reconstruction task was initiated. Additionally the same reference image would undergo repeated feature extraction processes for each reconstruction task, which was inefficient. To rectify this, a critical improvement identified was the system's ability to store multiple reference images (in various orientations, when necessary) along with precomputed essential information for reconstruction, such as keypoints and descriptors. This enhancement was considered crucial to give instant access to vital data during reconstruction, eliminating redundant computations and boosting the system's performance.

[1] https://github.com/danielgatis/rembg

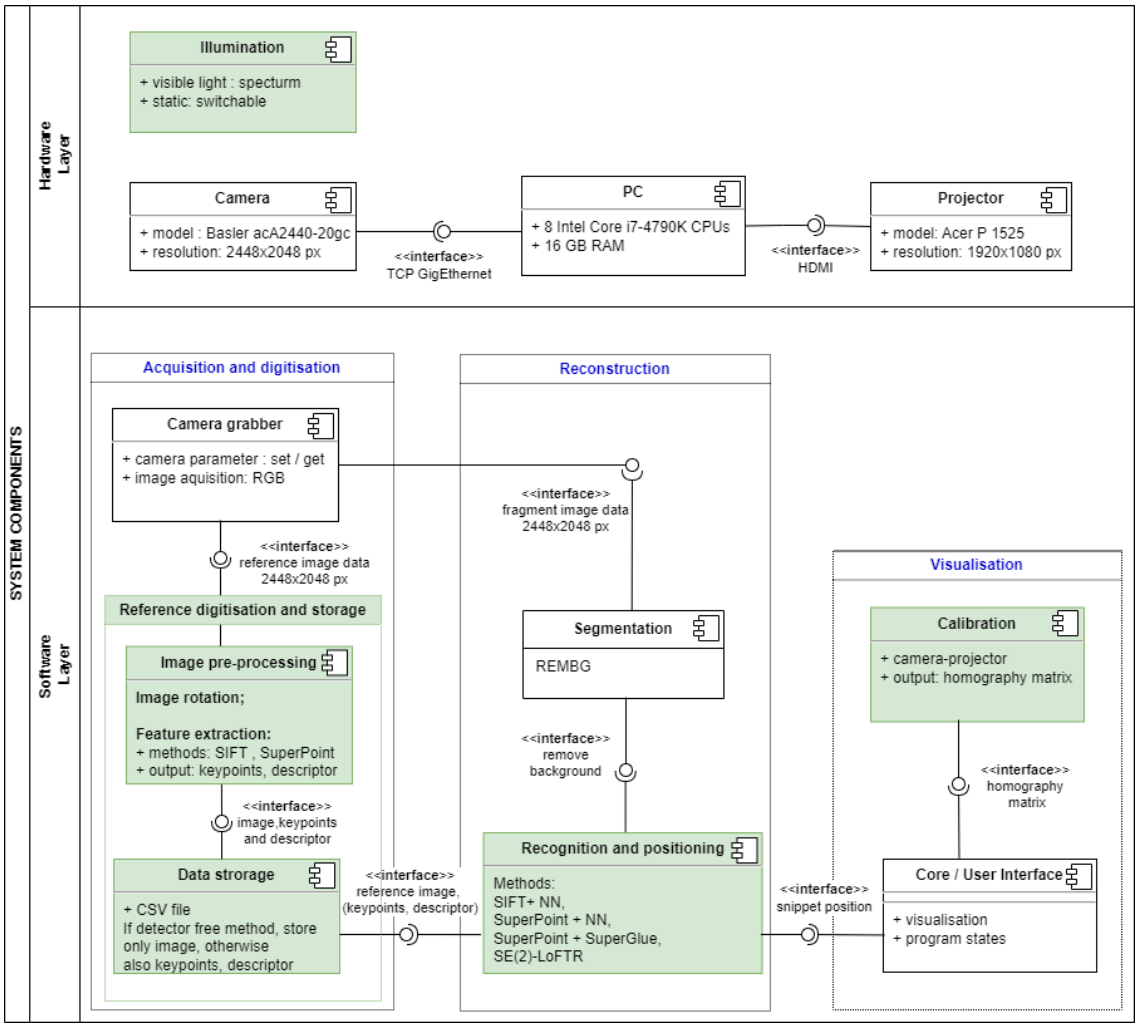


Figure 4.1: **Detailed overview of FalKe system components**. In green the components implemented and improved in this thesis.

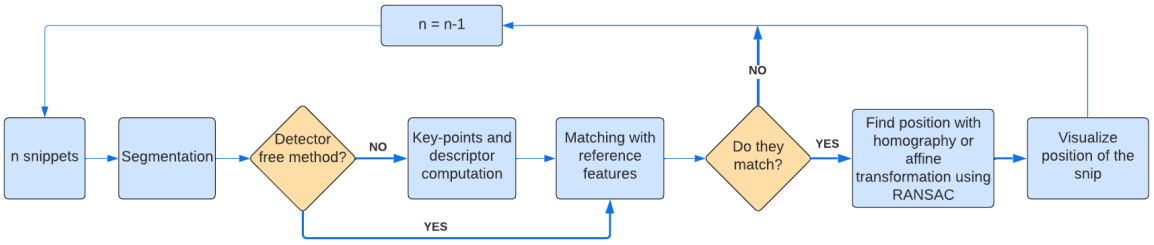


Figure 4.2: **Reconstruction pipeline**.

- **Reconstruction**. The primary limitations were identified in the document reconstruction process. The original system struggled to recognize fragments that were placed in the detection area in the wrong orientation. This issue highlighted the critical need for rotation invariance, enabling the system to seamlessly recognize and accurately locate fragments, irrespective of their initial orientation. This capability is particularly important as users typically don't know a fragment's original orientation prior to reconstruction. Further investigation into the system's performance revealed additional significant challenges. A key concern was the system's ability to accurately process fragments that differ in scale from the reference document. This is vital for the system's consistent performance across different camera resolutions. Given that camera setups can vary, the system must be capable of adapting to these variations without compromising accuracy. Moreover, the system's capacity to deal with variations in texture and appearance due to aging, noise, or damage (including stains, burns, or crumpling) is of paramount importance, particularly in the context of cultural heritage. These potential discrepancies between the fragment and the reference document were considered likely to occur and, thus, crucial to address. It became evident that a thorough evaluation of different local feature matching methods was necessary to enhance the system's ability to recognize and correctly position fragments under these varied and challenging conditions.
- **Visualisation**. The system in its initial state was missing a proper Camera-Projector Calibration, which directly impacts the user interface of the system, vital for guiding through the reconstruction process. The original system's transformation from the camera coordinate system to the projector coordinate system was relying on manual measurements of the detection area rectangle and distances between the camera and projected rectangles. This method, while functional, was limited by its dependence on precise measurements and lacked flexibility. Thus a calibration process designed to be dynamic, user-friendly, and less reliant on fixed hardware measurements, thereby enhancing the system's accuracy and adaptability, was identified as a requirement to ensure the best user experience of the FalKe system.

These requirements are crucial for aligning the system's functionality with the goals of this thesis. Table 4.1 summarizes these essential requirements across different system components. In the subsequent sections, the implemented solutions for these system components, addressing the outlined requirements, will be presented.

4.2 Hardware setup

The hardware setup of the FalKe system is strategically designed to facilitate the capture and projection of images for the document reconstruction process. The setup, illustrated in Figure 4.3, is anchored by a robust support structure designed to securely hold and interconnect all the system components, ensuring stability and reliability during operation. The setup comprises the following components:

- **Camera**. Central to this setup is a Basler acA2440-20gc camera with 5 MP resolution (2448 x 2048 pixels (px), equipped with a Fujinon 1:1.4/25mm lens, which is mounted to the structure by a versatile arm that provides complete freedom of movement. This arm allows the camera to move along the x, y, and z axes and to rotate, offering the flexibility to adjust its position and angle as needed for optimal image capture.
- **Projector**. An Acer P 1525 projector with Full HD (1920 x 1080 px) resolution is installed at at a height of 109cm from the table surface. It has an adjustable mount

System Component	Specific Requirements
Lighting System	• Adjustable intensity
Camera-Projector Calibration	• User-friendly and easy-to-perform process • Precise
Digitisation and Storage System	• Ability to store multiple reference documents in various orientations • Ability to precompute and store information according to different methods
Recognition and Positioning System	• Robustness to different orientations • Robustness to different damage levels • Robustness to different scalings • Robustness to noise

Table 4.1: **FalKe system requirements**.

that allows for horizontal movement and rotation. This adjustability ensures that the projection can be precisely centered over the table's surface, covering a rectangular area sufficiently large for an effective user interface.

- **Illumination**. The newly added lighting system, comprising adjustable diffuse lighting, is critical in creating a consistent environment for image processing. With a variable intensity control switch, it offers the capability to fine-tune the illumination across the workspace to suit different document types and ambient conditions. Initial trials with an infrared camera paired with a ring of infrared lights were deemed unsuitable for the system's requirements, as the infrared illumination did not cover the imaging area evenly. This led to a shift back to the current lighting method, which provides a more uniform distribution of light across the entire field of view.

- **PC**. The system's computational needs are met by a powerful PC equipped with eight Intel Core i7-4790K CPUs, capable of handling the demanding tasks of image processing and content projection with ease. The PC specifications include 16 GB RAM, which contribute to the system's overall performance and speed.

- **Caption and projection surface**. The system's caption and projection surface is equipped with an opaque white slab, chosen for its non-reflective quality to ensure that the illumination system doesn't cause reflections that could degrade the visibility of the projected user interface or the clarity of the camera's image captures. Additionally, in the detection area can be placed a green paper overlay. The contrast provided by the green background allows for more precise detection and segmentation of document snippets in the reconstruction process.

4.3 Camera-Projector Calibration

This thesis introduces a novel approach to camera-projector calibration, significantly advancing from the previously established method. Precise calibration of the camera and projector

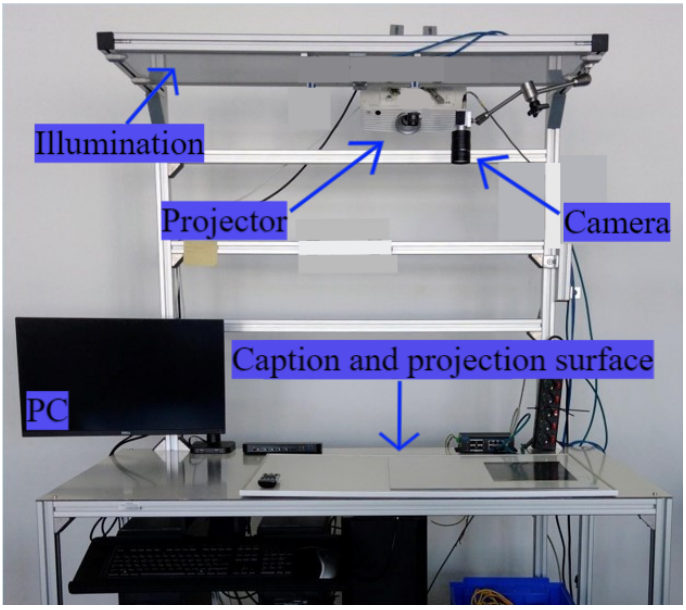


Figure 4.3: **Hardware setup of FalKe system**

setup is central to the document reconstruction system, as it directly impacts the user interface — the component guiding users through the entire document reconstruction process. The adjustable nature of the camera and the possibility of altering the hardware setup, make it imperative to perform calibration before every use to guarantee the best result in the reconstruction process. Thus, the calibration procedure's simplicity becomes as valuable as its precision. In this section a three steps calibration procedure easy to perform and accurate is presented.

The system's focus on 2D image capture and display simplifies the calibration process by eliminating the need for intrinsic and extrinsic calibrations, typically required for 3D reconstructions. Hence, a homography-based approach is employed, which streamlines the calibration while ensuring accurate 2D projections and captures.

As outlined in Section 2.1, a homography transformation relates two planes only if the world scene is planar. In the setup depicted in Figure 4.3, the caption and projection occur on a flat surface, making this condition applicable. The relationship between the camera plane and the projection plane can thus be defined by a planar homography. As described in Section 2.1.3, a homography correlates the camera's and the projector's view of the same planar scene, allowing for seamless computation of points in the projection plane based on the points from the camera plane (Equation 2.6).

The designed calibration process is methodically structured into three steps, each building upon the accuracy of the previous to achieve a finely tuned mapping between the camera image plane and projector image plane. A clear user interface projected on the working surface guides the user through each step. The live camera feed is an essential feature of this process, providing the user with continuous feedback for adjustments and verification. The flowchart in Figure 4.4 gives an overview of this process, illustrating its simplicity and efficacy.

Step 1: Manual camera alignment

The goal at this stage is to roughly align the camera's field of view with the projected detection area's rectangle. This is facilitated by drawing diagonals in both the camera image

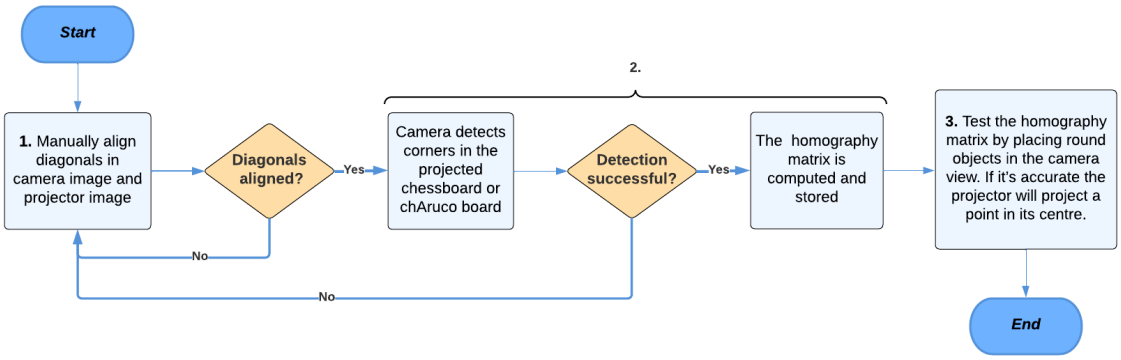


Figure 4.4: **Calibration process pipeline**.

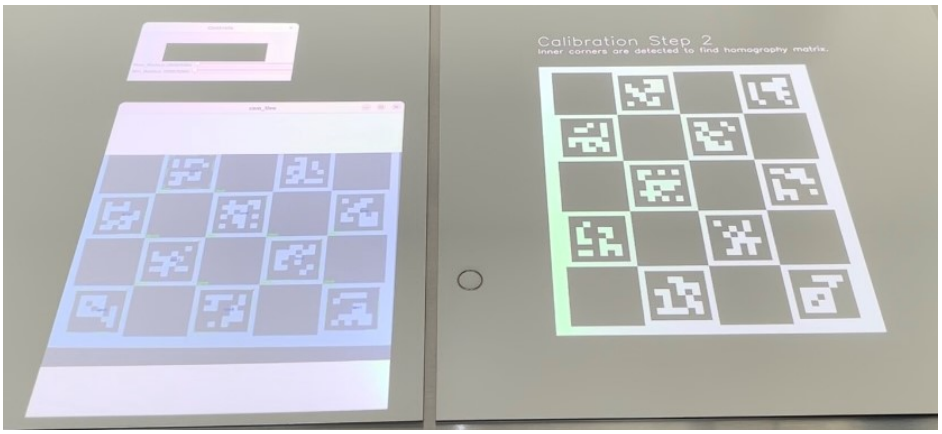


Figure 4.5: **Calibration process: step 2**. A ChAruco Board is projected in the camera view on the right; on the left the camera live feed is visible. The camera detects the 12 inner corners and markers and computes the homography matrix.

and the projected detection area's rectangle, which helps in manual adjustment. The live camera feed is crucial in this step, enabling the user to see the camera view and easily manually adjust the camera's position to align the camera's image diagonals with those of the projected area. This alignment doesn't need to be pixel-perfect but should be sufficient to assist in subsequent detection and document reconstruction process. Additionally, the user can adjust the lighting intensity if necessary, either by modifying the camera's focal aperture or using the new lighting system's control switch.

Step 2: Calibration pattern detection and homography matrix computation

After establishing the rough alignment, the next step, which can be seen in Figure 4.5, involves projecting and detecting a calibration pattern. Users can opt for a traditional chessboard pattern or a ChAruco board of desired dimensions, with a common preference being a 5x4 configuration. A ChAruco board consists of a grid of Aruco marker patterns where each marker has a binary matrix inside and is associated with a unique identifier. This design allows for detection even in partially occluded views and is therefore preferred over the chessboard pattern. The inner corners of the projected pattern are detected in the camera image using OpenCV corner-detection functions. The detection is deemed successful if at least four inner corner points are identified in the camera image. If the detection fails, the process goes back to step 1. As explained in Section 2.1.3 at least four points correspondences

Figure 4.6: **Calibration process: step 3**. The user tests the accuracy of the computed homography matrix by placing a ring in the camera view. On the left side the camera live feed is visible, as well as the controller to adjust the max and min radius of the object to detect. Here the estimated homography matrix is pretty accurate, since the projector is able to draw a point exactly in the ring's center.

between the two planes are needed to uniquely compute a homography matrix. The known positions of the inner corners in the projector's image plane allow for the computation of the homography matrix once the detection of inner corners in the camera's image plane is successful. This matrix is obtained using the OpenCV function findHomography() which relies on RANSAC, as explained in Sections 2.1.3 and 2.1.5. The homography matrix is saved and used to transform points or entire images from the camera's coordinate system to the projector's coordinate system using the openCV function warpPerspective().

Step 3: Testing the calibration accuracy

In the testing phase, the user can test the accuracy of the homography matrix by placing one or more flat, round objects of various sizes within the camera's field of view. The system, through the OpenCV function HoughCircles(), detects these objects in the camera image and retrieves the coordinates of their center points. The radius parameter is adjustable in the UI, ensuring objects of different sizes can be accommodated. The system then applies the previously computed homography matrix to the detected center points in the camera view determining their corresponding location in the projector's plane following Equation 2.6. Now the system can draw these points in the projector's image. If the homography transformation is correct, the projected points will match the objects' center precisely, validating the accuracy of the calibration. This step, which can be seen in Figure 4.6, is crucial as it provides an immediate verification of the calibration's accuracy, ensuring that the system is ready for the intended document reconstruction process.

4.4 Digitization and Storage System for reference documents

The system's approach to document reassembly is fundamentally dependent on the use of reference document images. Given this reliance, a crucial aspect of enhancing the system's efficiency is the ability to have key data about multiple reference documents already computed and stored, enabling easy retrieval during the reconstruction process. To this end a robust digitization and storage system for reference documents was developed. This system component is designed to retrieve and store key information from the given reference

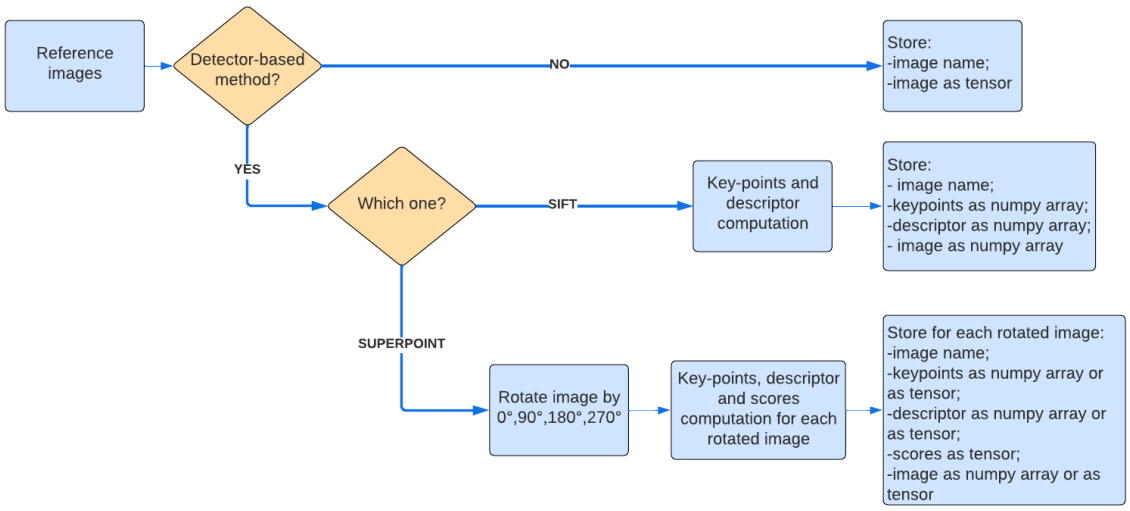


Figure 4.7: **Digitisation and Storage System pipeline.**

```
C:\Users\dia03602\Desktop\BA\reference_dataset>tree
Folder PATH listing for volume Windows
Volume serial number is C4CD-A4D6
C:.
├───descriptors
├───images
├───keypoints
└───scores
```

Figure 4.8: **Tree of the stored data from the reference images**.

images in a manner tailored to the selected combination of feature detection and matching methods. This versatility is crucial for adapting the reconstruction workflow to the specific requirements of each algorithm, as depicted in Figure 4.7.

Detector-based methods like SIFT and SuperPoint benefit significantly from the implemented solution. By precomputing keypoints and descriptors, the system stores this information in a structured dataset. This dataset, shown in Figure 4.8, includes folders for keypoints, descriptors, preprocessed images and feature scores, with the data saved as tensors or numpy arrays depending on the method's requirements. The file paths are then indexed in a CSV file, enabling swift retrieval during the reconstruction phase without the need for repeated computations. When a fragment is presented for reassembly, the system only needs to compute the key-points and descriptors for that specific fragment. It then compares these newly computed features to the pre-stored descriptors of all the reference documents in the dataset. Table 1 in the Annex section shows how the header of the CSV file would look like for the different methods.

A unique aspect of the implemented approach arises when using the SuperPoint method. As identified in Section 3.2.1, SuperPoint's main limitation is its descriptors' lack of rotational invariance. To address this, a strategic solution involves matching each fragment image with reference images at different rotations. Thus, each given reference document is first rotated of 0, 90, 180, and 270 degrees and then for each rotated reference keypoints, descriptors, and scores are computed and stored. During the reconstruction stage, the descriptors of each fragment are matched with those from every rotated reference image. This approach successfully reinforces the SuperPoint method against orientation variations and doesn't increase the computational load during the matching process, as these features are precomputed and readily available for each reference document.

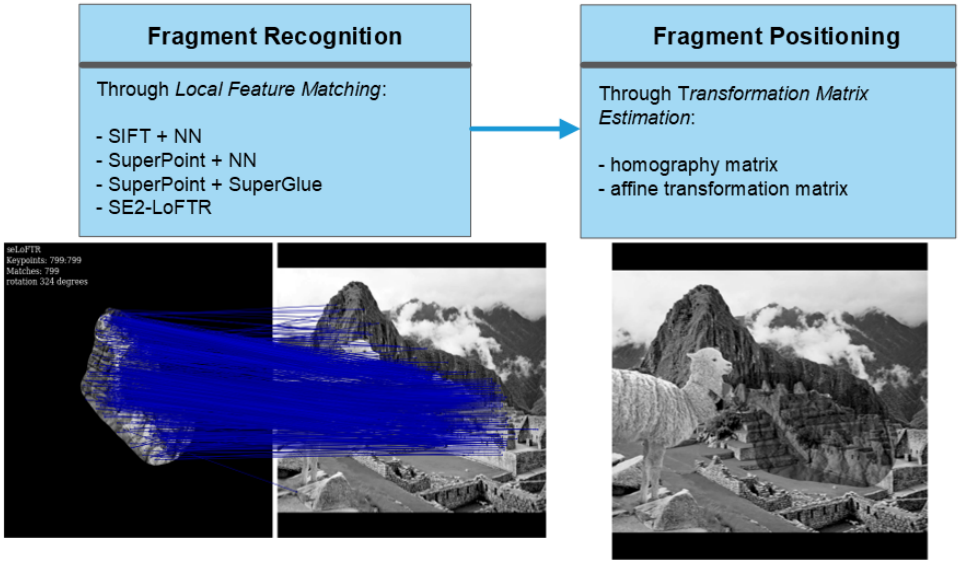


Figure 4.9: **Fragment Recognition and Positioning**.

In contrast, for detector-free methods like SE2-LoFTR, the process differs. As explained in Section 3.2.2, SE2-LoFTR utilizes a learning-based approach to analyze and match patterns across entire image regions directly from the raw image data. Therefore in case of a detector-free method the system just stores the reference image name and the image itself as a tensor.

The integration of this storage system into the main framework introduces an additional step in the reconstruction process. This step involves identifying the reference document to which a fragment belongs, after matching it with all stored references. After each one-to-one matching, the system calculates a score to quantify the degree of match between the fragment the reference. For detector-based methods like SuperPoint or SIFT, this score s is computed using the equation:

$$s = \frac{\text{Number of inliers}}{\text{Number of keypoints in fragment image}} \times 100 \tag{4.1}$$

where the inliers are the geometrically consistent matched keypoints filtered with RANSAC. For detector-free methods like SE2-LoFTR, the score s is determined by the count of matches. The (eventually rotated) reference image that achieves the highest score is then determined to be the best matching reference for that snippet.

This solution considerably augments the system's ability to effectively manage and utilize an extensive dataset of reference documents, underscoring its potential in the preservation and restoration of cultural heritage.

4.5 Recognition and Positioning System

The Recognition and Positioning System is a vital component of the document reconstruction process, as depicted in Figure 4.1. Its role, outlined in Figure 4.9, centers on utilizing local feature matching methods to accurately identify the area where a fragment belongs in the complete document. The matches found are used to estimate a transformation between the images — either a homography or affine transformation matrix — using RANSAC, as detailed in Section 2.1.5. The computed transformation matrix is then applied to the fragment,

precisely positioning and aligning it in the identified area within the reference document. The quality of the matches directly influences the accuracy of the transformation matrix, and consequently, the overall accuracy of the reconstruction. Therefore, the success of the entire reconstruction process hinges on the reliability of the local feature matching algorithm, which should ideally be robust to rotations, scale and variations in lighting and texture conditions.

4.5.1 Selected local feature matching methods

As detailed in section 4.1, the initial system encountered challenges in handling the reassembly of fragments with significant rotations, scale variations, and altered appearances. These limitations were due to the suboptimal performance of the sole local feature matching method implemented in the system: SuperPoint with NN as a matcher. Therefore, the primary contribution of this thesis is the implementation and investigation of various local feature matching methods to enhance the accuracy and robustness of the document reconstruction in the FalKe system. The local feature matching methods implemented for this investigation are:

- SIFT + NN (BF-Matcher or FLANN Matcher)
- SuperPoint + NN (BF-Matcher or FLANN Matcher)
- SuperPoint + SuperGlue
- SE2-LoFTR

The selection of these methods was informed by the research into traditional and state-of-the-art local feature matching methods outlined in Section 3.2. SIFT was chosen for its standout qualities among classical descriptors, notably its robustness in challenging scenarios involving illumination, scale, and rotation. SuperPoint was implemented for its advanced keypoint detection and description capabilities, which are significantly enhanced when combined with SuperGlue, a state of the art matching algorithm known for its precision and reliability in feature correspondence. The original SuperPoint implementation with Nearest Neighbor matching (NN) was also included for comparative analysis to assess enhancements in the reconstruction when varying the feature matching method in the FalKe system. Among detector-free methods, SE2-LoFTR was chosen for its inherent rotation invariance and streamlined workflow, making it highly effective in diverse challenging matching scenarios.

4.5.2 Investigation

The goal of the here presented investigation is to find out which local feature matching method best meets the requirements of the Recognition and Positioning System in the FalKe system. A systematic and automated approach was employed to evaluate the reconstruction accuracy of each method. Key to this inquiry is the development of a snippet dataset, crafted to simulate various forms of paper damage, providing a realistic basis for testing. Three targeted experiments were implemented to evaluate the rotation, scale and noise robustness of each method under diverse damaged paper conditions, aiming to enhance the overall document reconstruction system.

The following sections succinctly outline the dataset creation and detail the experimental setup and methodologies employed. The comprehensive analysis and discussion of the experiments' results, however, will be reserved for the subsequent Chapter 5.

Dataset

Five distinct types of documents were selected as reference images, as shown in Figure 4.10: printed text, typewritten, handwritten, technical drawing and outdoor landscape. Although

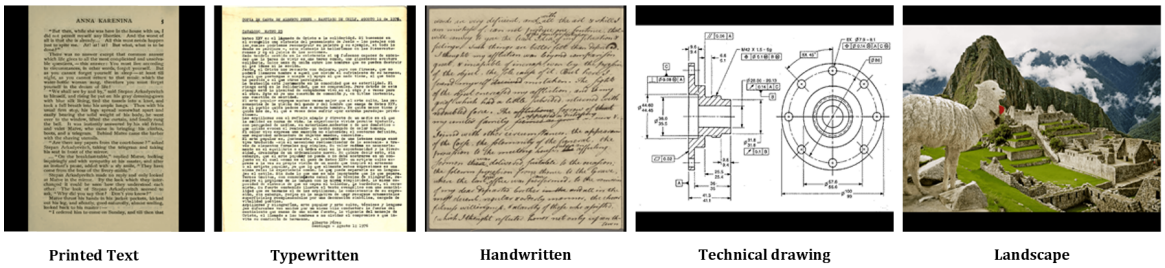


Figure 4.10: **Reference documents**.

printed text and typewritten documents bear similarities, both were included due to their high relevance to the Falke system's use case in cultural heritage restoration. The choice of these document types was strategic: printed text, typewritten, and handwritten documents are characterized by their highly repetitive patterns. In contrast, the technical drawing exhibit sparser features, primarily composed of distinct geometric shapes and lines, presenting a different set of challenges for feature matching methods due to their less textured nature. The outdoor landscape, representing the standard imagery of natural scenes, is classified as the *easy* document type because it offers a broader variety of distinctive features like varying textures and natural elements. This diverse selection of reference images, each with its unique pattern complexity, provides a comprehensive basis for evaluating the robustness and adaptability of the feature matching methods. The dimensions of the reference images are fixed at 2448x2048 pixels (px), corresponding to the camera resolution.

A dataset of snippets was then generated by automatically extracting sections from each reference image using a freehand mask. This process is meant to simulate the manual ripping of paper documents in fragments of irregular shape. A critical aspect of this investigation is the digital creation of discrepancies between the fragments and the reference documents. To mimic the effects of aging and damage on paper a combination of filters simulating burns, stains, and textures was applied to each snippet. These filters were merged using the multiply blending option in an editor[2], creating a realistic representation of aged and damaged documents. To categorize the difficulty of processing these snippet images, four damage levels were defined based on the combination and intensity of the filters applied: easy, medium, hard, and very hard. Figure 4.11 illustrates the various damage levels using the printed text snippet as an example. The *easy* level consists of fragments without any filter applied, showcasing the snippet in its original, undamaged state. For the *medium* level, the single filters burned, stained, and textured were applied, introducing moderate damage effects. The *hard* level involved applying a combination of stained and textured filters to simulate severe paper damage. Finally, the *very hard* level combined all three filters – burned, textured, and stained – to replicate the most challenging damage conditions. This meticulous process yielded a dataset comprising 30 snippets, 6 snippets for each reference.

Experimental setup and implementation details

All experiments were conducted using *Python 3.11* with *OpenCV 4.8*. Specifications of the computer system used are: AMD Ryzen 5 PRO 5675U with Radeon Graphics 2.30 GHz and 16.00 GB RAM.

For SIFT, BF-Matcher and FLANN-Matcher the default OpenCV implementations are used. BF-Matcher and FLANN-Matcher didn't show any difference in the results of the matching, therefore BF-Matcher was chosen as NN-Matcher for SIFT and SuperPoint. SIFT

[2]https://photokit.com/editor/

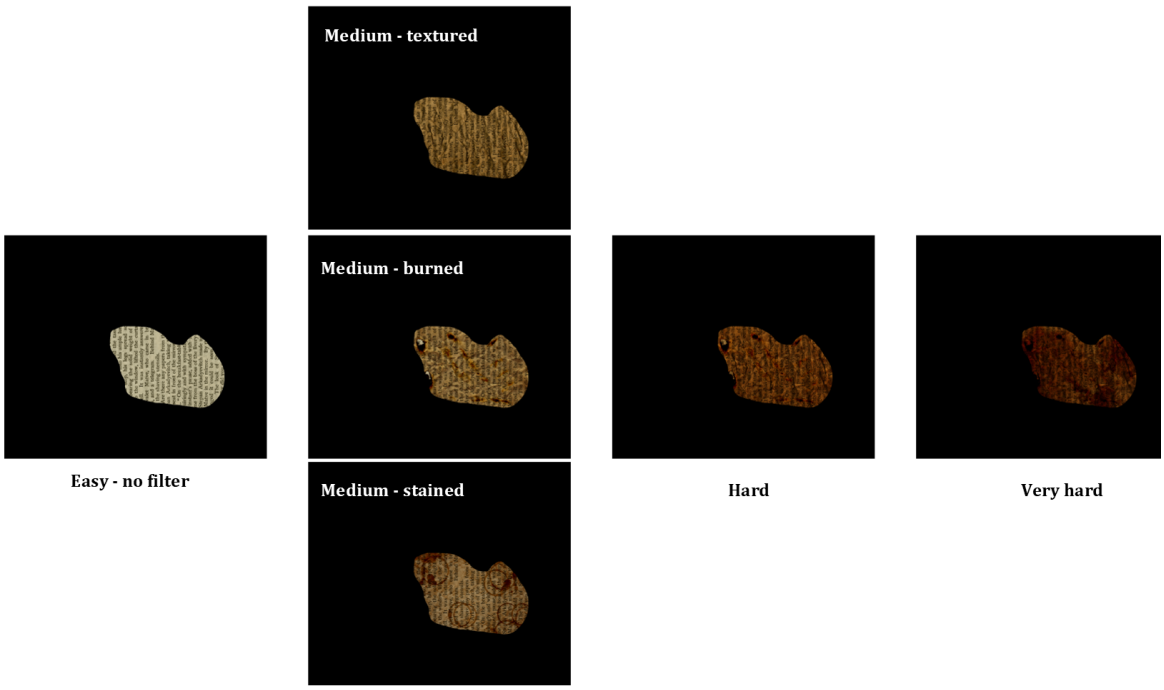


Figure 4.11: **Damage levels on printed text document**.

also uses the *Ratio Test*, explained in section 3.2.1, with a threshold of 0.7.

For SuperPoint, SuperGlue and SE2-LoFTR the pre-trained models provided by the authors are used. SuperGlue uses the pretrained *outdoor* weights, because they showed significantly better results in the matching compared to the *indoor* ones also on text documents. For SE2-LoFTR the *8-rotations* pretrained weights are used. Table 4.2 shows some important implementation parameters.

For optimal positioning, the decision was made to estimate the affine transformation matrix, rather than the homography matrix. This choice was based on the lack of perspective distortions and the superior reconstruction results it yielded, even with minimal matches found. The OpenCV function utilized for this purpose is estimateAffinePartial2D() with RANSAC, giving all the matches as input.

Method	**Implementation parameters**
SuperPoint	nms_radius = 4 keypoints_threshold = 0.05
SuperGlue	sinkhorn_iterations = 20 match_threshold = 0.05
SE2-LoFTR	coarse_threshold = 0.1 match_type = dual_softmax coarse_attention = "linear" cnn_resolution = (8,2)

Table 4.2: **Important models' parameters used for the experiments**.

Experiments

Three experiments were conducted to evaluate how the various methods reconstruct noisy, rotated and scaled fragments. The **Noise Robustness Experiment** assesses each method's ability to handle random noise. The snippet images are subjected to ten levels of noise

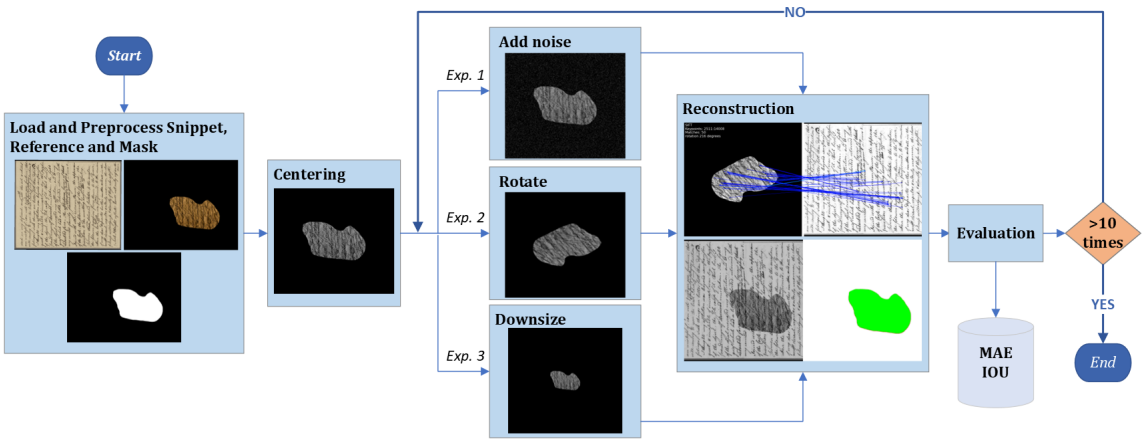


Figure 4.12: **Experiments' pipeline**. Each snippet in the dataset goes through this pipeline. The three experiments are computed and evaluated all at once for every method.

to mimic real-world environmental challenges. The **Rotation Robustness Experiment** evaluates the methods' invariance to rotation. The snippet images are rotated from 0 to 360 degrees in ten fixed increments of 36 degrees, testing the methods' capacity to recognize and match features across different orientations. Lastly, the **Scaling Robustness Experiment** evaluates the algorithms' scale invariance by downsizing snippet images in ten stages, from no reduction to a 90% size decrease.

All the snippets in the created dataset go through the implemented pipeline that executes the three experiments all at once for every method. This pipeline is presented in Figure. 4.12 and comprises the following steps:

1. **Loading and preprocessing of data**. The process begins with loading the reference document, the digitally ripped snippet, and the snippet mask. The mask is crucial for defining the original area (ground truth area) of the snippet in the reference document, essential for the later evaluation of the reconstruction. These 3 images are then preprocessed to adapt them to a suitable format for analysis. They are converted to grayscale, resized to 1/2 of their size (1224x1024 px), and cropped to ensure they are square and of odd dimensions. This is important for consistency across all experiments and to facilitate certain transformations like rotation.

2. **Centering**. The snippet and mask are placed in the center of a blank canvas each. This standardizes the location of the snippet across different experiments, ensuring that any transformations applied are centered around the same point.

3. **Transformation**. Depending on the experiment and the iteration, which represents the noise level, rotation angle or downsize level, the snippet undergoes a transformation:
   - **Noise addition**. Simulating the effect of random noise on the snippet.
   - **Rotation**. Rotating the snippet through 360 degrees.
   - **Downsizing**. Gradually reducing the size of the snippet.

   Each transformation is applied 10 times: 10 noise levels, 10 rotation angles, 10 downsize levels.

4. **Reconstruction**. The transformed snippet is matched against the corresponding reference image with each method. When using SuperPoint as a detector, the solution explained in Section 4.4 to boost the rotation robustness of its descriptors is used: each

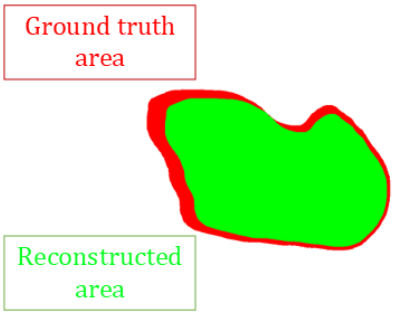


Figure 4.13: **Intersection Over Union (IOU)**. IOU = 0.83

fragment is matched against 16 rotated versions of the reference document. The rotated reference with the highest matching score, calculated following Equation 4.1, gives the right orientation of the snippet. The matches are then used to estimate the homography or affine transformation matrix using RANSAC, as explained in Section 2.1.5. This transformation matrix is then applied to the fragment, aligning it with the reference. The more accurate the transformation matrix, the more precise is the alignment and therefore the reconstruction. For example the reconstruction shown in Figure 4.9 is so accurate that the snippet is hardly identifiable.

5. **Evaluation**: Each experiment captures metrics such as Mean Absolute Error (MAE) and Intersection Over Union (IOU) to quantify the accuracy and precision of the reconstructed fragment. The performance is evaluated using two metrics:

   - **MAE**. This measures the average absolute difference between the reconstructed coordinates of the snippet and its original coordinates - called ground truth coordinates - in the reference image. It assesses the accuracy in terms of the positional correctness of the reconstruction. A lower MAE indicates that the reconstructed points are closer to their true positions. Its mathematical formula is:

   $$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - G_i| \tag{4.2}$$

   where n is the number of points and $|P_i - G_i|$ is the absolute difference between the reconstructed point P_i and the ground truth point G_i of the snippet in pixels.

   - **IOU**. This assesses the overlap between the reconstructed snippet mask and the original mask in the complete reference document, providing insight into the accuracy of the reconstruction. It measures how well the reconstructed snippet's area aligns with the original snippet's area. It's expressed as:

   $$IOU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{4.3}$$

   where the *Area of overlap* is the common area shared by both the reconstructed snippet mask and the original snippet mask; and the *Area of union* is the total area covered by both masks combined, subtracting the overlapping area once to avoid double counting. The closer the value is to 1 the higher the overlap of the masks. Figure 4.13 shows an IOU of 0.83, which is a moderately good overlap.

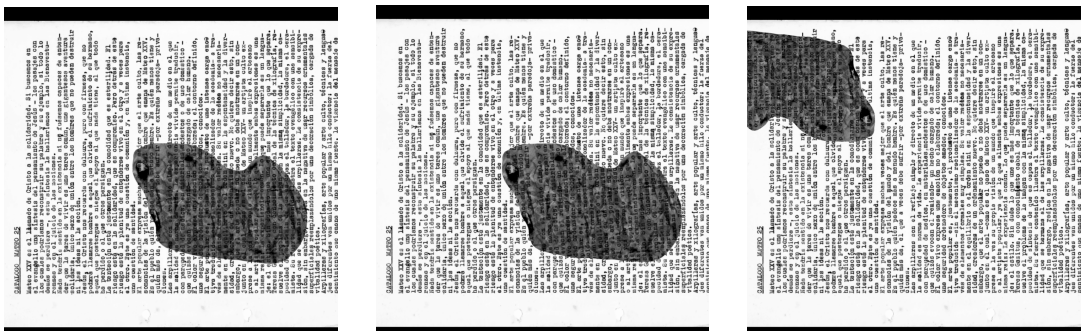
5 Results

This chapter reveals the results from the experimental pipeline outlined in Section 4.5.2, presenting a comparative analysis of the methods across the three experiments. Each method's performance is evaluated by the level of paper damage and by document type, which were predefined in Section 4.5.2. These criteria are pivotal in offering insights into the methods' effectiveness in diverse scenarios.

The results are categorized into three performance levels based on the Mean Absolute Error (MAE): "Good (MAE = [0,8]px)", "Moderate (MAE = [9,22]px)", and "Failed (MAE > 22px)", as illustrated in Figure 5.1. The "Good" category represents cases with perfect alignment between the fragment and the reference document. The "Moderate" category includes reconstructions that are visually correct but exhibit some minor misalignments upon closer examination. The "Failed" category covers instances where the fragment is completely misaligned with the reference document.

5.1 Noise Robustness Experiment

The Noise Robustness Experiment assesses the accuracy of the implemented methods when reconstructing noisy fragment images, already subjected to four different damage intensities (no damage, medium damage, hard damage, very hard damage). Ten levels of noise were layered on top of each snippet image. Figure 5.2 gives an overview of the performance of the investigated methods across the different document types and levels of filtering, counting how many times each method did good, moderately or failed in reconstructing the noisy snippet. The methods' performance was similar across textured, stained, and burned filters, as shown in Figure 1 in the Annex. Notably, the stained filter yielded slightly better outcomes (failure rate of 25%) compared to burned and textured filters (26% failure rate each). Consequently, the textured filter was chosen as the medium damage level. Table 5.7 and 5.8 order the best to worst performative methods by filter and by document type respectively, while Table 5.9 shows the failure rate of the methods by document type across all filter levels.



(a) Good. MAE=2.54 px (b) Moderate. MAE=17.76 px (c) Failed. MAE=751.26 px

Figure 5.1: **MAE categories to evaluate reconstruction**: (a) Good (MAE = [0,8]px), (b) Moderate (MAE = [9,22]px), (c) Failed (MAE>22px).

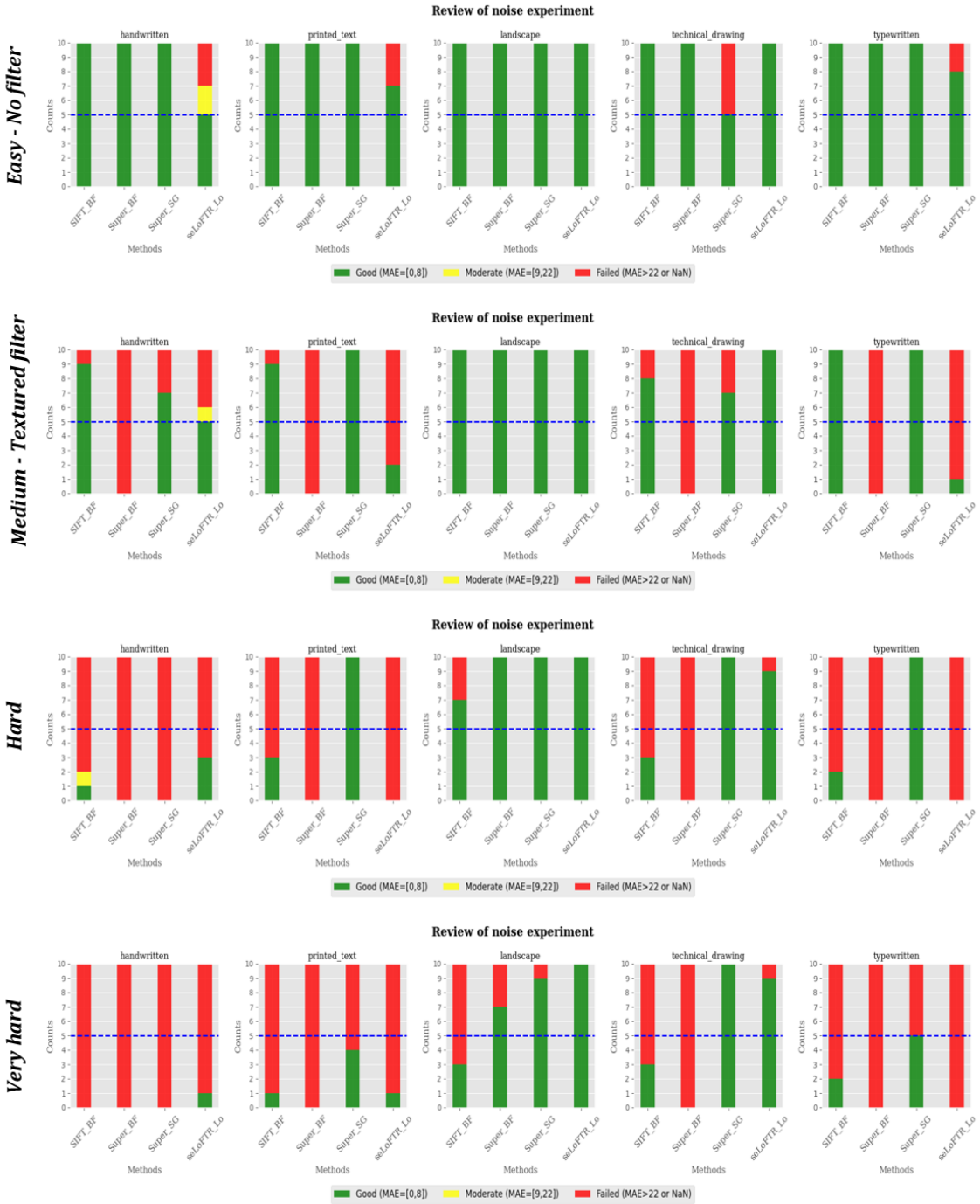


Figure 5.2: **Results of the Noise Robustness Experiment**. The graphs illustrate the performance of each method under different damage levels and document types. For each damage level and document type, the graphs depict the number of times each method reconstructed the fragment across ten levels of noise, categorizing the results as 'Good (MAE = [0,8])' in green, 'Moderate (MAE = [9,22])' in yellow or 'Failed (MAE>22 or NaN)' in red.

Easy - No filter

In the easy scenario, SIFT and SuperPoint + NN exhibited the best performance. SuperPoint + SuperGlue, while generally effective, had challenges in reconstructing the technical drawing, leading to failed reconstruction in 50% of the noise levels. SE2-LoFTR struggled notably from noise level 7 onwards for all document types, except in technical drawing and landscape image where it always performed well.

Medium - Textured filter

With the textured filter, SIFT again led in performance, closely followed by SuperPoint + SuperGlue. SuperPoint + NN consistently failed at all levels and document types. SE2-LoFTR's performance varied, excelling in technical drawings and landscape but faltering in documents with repetitive patterns like printed text, typewritten and handwritten documents.

Hard

Under hard conditions, all methods faced difficulties, particularly with the handwritten document. SuperPoint + NN failed for every document type expect for the landscape, which resulted the easiest to reconstruct for all methods with only SIFT struggling on the very highest levels of noise. SuperPoint + SuperGlue stood out for its exceptional performance for all document types, failing only in the handwritten document. SE2-LoFTR showed commendable results only in landscape and technical drawing images.

Very Hard

In the very hard scenario, all methods struggled with printed text, typewritten, and handwritten documents, with SuperPoint + SuperGlue being the only method that managed to perform moderately for typewritten and printed text documents. The handwritten document proved the most challenging for all methods. SuperPoint + SuperGlue and SE2-LoFTR performed well in technical drawing and landscape images.

Overall performance

In summary, the hardest document types to reconstruct were the ones showing very repetitive patterns, ergo typewritten, printed text and handwritten, the latter being the absolute hardest - as shown by Table 5.3. The landscape image proved to be the easiest to reconstruct across all methods. SuperPoint + SuperGlue emerged as the overall most effective method, particularly in reconstructing the documents with repetitive patterns, even under hard and very hard damage conditions, while SIFT resulted more adapt in handling less severe damage levels (see Table 5.1). SE2-LoFTR outperformed the other methods on landscape and technical drawing documents, as revealed by Table 5.2, especially with medium to very hard damage levels.

Filter Level	**Overall best performance**
No Filter	**SIFT+NN** = **SP+NN** > SP+SG > SE2-LoFTR
Medium - textured	**SIFT+NN** > SP+SG > SE2-LoFTR > SP+NN
Hard	**SP+SG** > SE2-LoFTR > SIFT+NN > SP+NN
Very hard	**SP+SG** > SE2-LoFTR > SIFT+NN > SP+NN

Table 5.1: **Noise Robustness Experiment: overall performance by filter level**.

Document type	Overall best performance
Handwritten	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Printed text	**SP+SG** > SIFT+NN > SE2-LoFTR = SP+NN
Typewritten	**SP+SG** > SIFT+NN > SP+NN > SE2-LoFTR
Technical drawing	**SE2-LoFTR** > SP+SG > SIFT+NN > SP+NN
Landscape	**SE2-LoFTR** > SP+SG > SP+NN > SIFT+NN

Table 5.2: **Noise Robustness Experiment: overall performance in the noise experiment by document type**.

Document type	Failure rate
Handwritten	59.4 %
Printed text	51.9 %
Typewritten	51.3 %
Technical drawing	35 %
Landscape	8.8 %

Table 5.3: **Noise Robustness Experiment: failure rate**. In red the document type with the highest failure percentage.

5.2 Rotation Robustness Experiment

The Rotation Robustness Experiment is designed to evaluate the level of rotation invariance of the implemented methods, testing their resilience across four levels of paper damage (no damage, moderate damage, hard damage, very hard damage). For this test, each snippet was rotated through ten angles ranging from 0 to 324 degrees in increments of 36 degrees. Figure 5.3 gives an overview of the performance of the investigated methods across the four levels of filtering, counting how many times each method did good, moderately or failed in reconstructing the rotated snippet. As medium damage level was chosen the textured filter, since the stained one gave the overall best results and the burned one the worst, with a higher failure rate (25%) compared to the other two (stained: 12%, textured: 20%). The results of this experiment on the stained and burned fragments are shown in Figure 2 in the Annex. Table 5.4 and 5.5 order the best to worst performative method by filter and by document type respectively, while Table 5.6 shows the failure rate of the methods by document type across all filter levels.

Easy - no filter

With unaltered documents, all methods excelled, achieving a MAE between 0 and 8 pixels, demonstrating their efficacy under optimal conditions. However, a notable exception was SE2-LoFTR's singular failure with the technical drawing, indicating a potential weakness with this type of documents.

Medium - Textured filter

Introducing the textured filter, simulating moderate damage, posed varied challenges, particularly for SuperPoint + NN in typewritten, printed text and technical drawing images. SIFT emerged as the most rotation robust, with no failures across all document types. SE2-LoFTR followed with an impressive performance, barring a few moderate difficulties with the technical drawing. SuperPoint + SuperGlue demonstrated very good performance as well, struggling only in the handwritten document. SuperPoint + NN found challenges with all

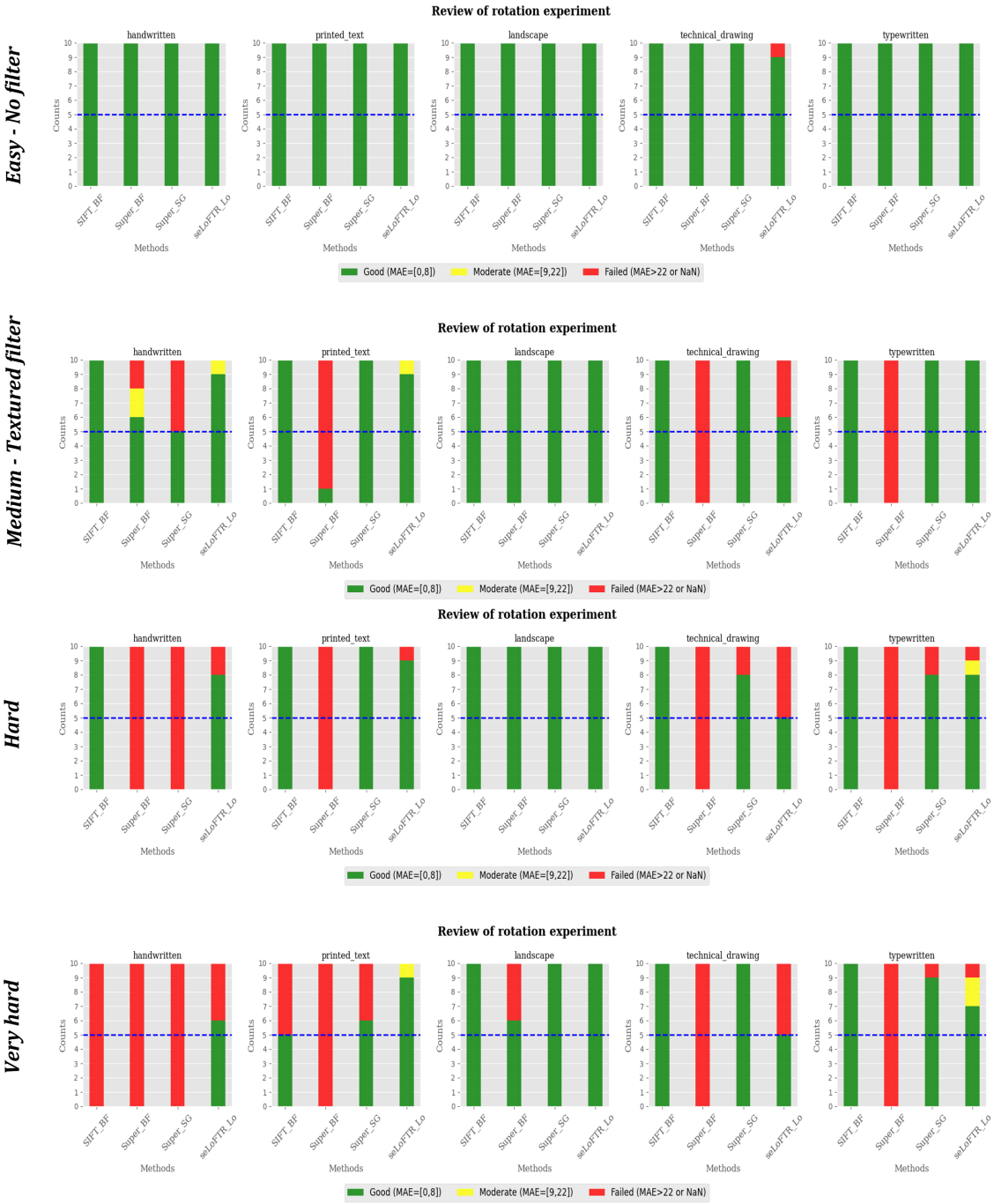
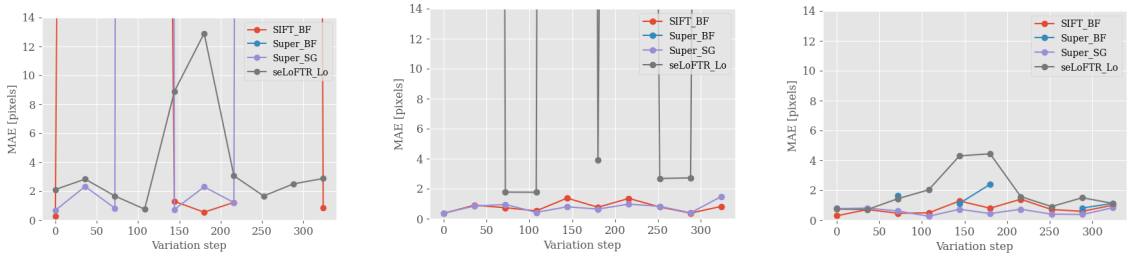


Figure 5.3: **Results of the Rotation Robustness Experiment**. The graphs illustrate the performance of each method under different damage levels and document types. For each damage level and document type, the graphs depict the number of times each method reconstructed the fragment across ten rotation angles, categorizing the results as 'Good (MAE = [0,8])' in green, 'Moderate (MAE = [9,22])' in yellow or 'Failed (MAE>22 or NaN)' in red.



(a) Printed text (b) Technical drawing (c) Landscape

Figure 5.4: **Rotation Robustness Experiment: *very hard* damage level MAE graphs**. (a) Printed text. SE2-LoFTR consistently reconstructs the fragment for every rotation angle with a low MAE, outperforming all other methods. (b) Technical drawing. SIFT and SP+SG outperform by reconstructing the snippet for every rotation angle, while SE2-LoFTR manages to achieve a low MAE only in five rotation angles. (c) Landscape. All methods reconstruct well the fragment for every rotation angle.

document types except for the handwritten one, where it performed moderately good.

Hard

Under the hard filter, the increase in failed outcomes was evident. Yet, SIFT maintained exceptional robustness across all document types, never failing. In contrast, SuperPoint + NN's struggles were pronounced, failing in all document types except for the landscape document. SE2-LoFTR, despite facing challenges with technical drawings, presented itself as the second most robust option, performing well across other document types. SuperPoint + SuperGlue managed to hold its ground with decent performance, except in handwritten documents, where it failed in all cases.

Very Hard

With the very hard filter, all methods encountered increased difficulty, evidenced by a rise in moderate and failed results. SE2-LoFTR stood out with best performance, especially with handwritten and printed text documents, as shown in Figure 5.4a. Most methods managed to maintain good performance with the landscape document, as shown in Figure 5.4c. SIFT, though faced with challenges, avoided failures in landscape, technical drawing, and typewritten documents. SuperPoint + NN failed for every document type expect for the landscape. SuperPoint + SuperGlue excelled in reconstructing technical drawing and landscape documents, as illustrated in Figure 5.4b and 5.4c.

Overall performance

In summary, SE2-LoFTR and SIFT demonstrated the most robust performance in rotation invariance across various document types and levels of damage - as revealed by Tables 5.5 and 5.4, showcasing their inherent rotation invariance. Notably, traditional SIFT often outperformed SE2-LoFTR, particularly with less damaged documents, showcasing its enduring effectiveness despite advancements in deep learning methods like SE2-LoFTR. SuperPoint + NN and SuperPoint + SuperGlue showed more limited robustness, excelling under certain conditions but struggling in others. The handwritten document was the most challenging to reconstruct, followed by the technical drawing, as shown in Table 5.6. The landscape document was the easiest, with only a 2% failure rate. The handwritten document posed

significant challenges for the SuperPoint detector, regardless of the matcher used. Technical drawings resulted particularly difficult for SuperPoint + NN, and the most difficult to reconstruct for SE2-LoFTR.

Filter Level	Overall best performance
No Filter	**SIFT+NN** = SP+SG = SP+NN > SE2-LoFTR
Medium - textured	**SIFT+NN** > SP+SG > SE2-LoFTR > SP+NN
Hard	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Very hard	**SE2-LoFTR** > SIFT+NN = SP+SG > SP+NN

Table 5.4: **Rotation robustness experiment: overall performance by filter level**.

Document type	Overall best performance
Handwritten	**SE2-LoFTR** > SIFT+NN > SP+NN > SP+SG
Printed text	**SE2-LoFTR** > SIFT+NN > SP+SG > SP+NN
Typewritten	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Technical drawing	**SIFT+NN** > SP+SG > SE2-LoFTR > SP+NN
Landscape	**SIFT+NN** = **SE2-LoFTR** = **SP+SG** > SP+NN

Table 5.5: **Rotation robustness experiment: overall performance by document type**.

Document type	Failure rate
Handwritten	39.4 %
Printed text	24.4 %
Typewritten	21.9 %
Technical drawing	29.4 %
Landscape	2.5 %

Table 5.6: **Rotation Robustness Experiment: failure rate by document type**. In red the document type with the highest failure percentage.

5.3 Scale Robustness Experiment

The Scale Robustness Experiment assesses the scale invariance of the implemented methods under four levels of paper damage (no damage, medium damage, hard damage and very hard damage). Snippets were downsized in ten increments, from their original size down to just 10% of their original size. Figure 5.5 gives an overview of how many times each method performed good, moderately or failed in reconstructing the downsized snippet across the four filtering levels. The textured filter was once again chosen as the medium damage level, since it offered a balanced challenge (55% failure) compared to the other 2 filters (stained: 48.5% failure, burned: 56% failure). The results of this experiment on the stained and burned fragments are shown in Figure 3 in the Annex. Table 5.7 and 5.8 order the best to worst performative method by filter and by document type respectively, while Table 5.9 shows the failure rate of the methods by document type across all filter levels.

Easy - No filter

For unaltered documents, SIFT showcased its scale invariance, successfully reconstructing nearly all downsized snippets across all document types, followed by SE2-LoFTR, which only

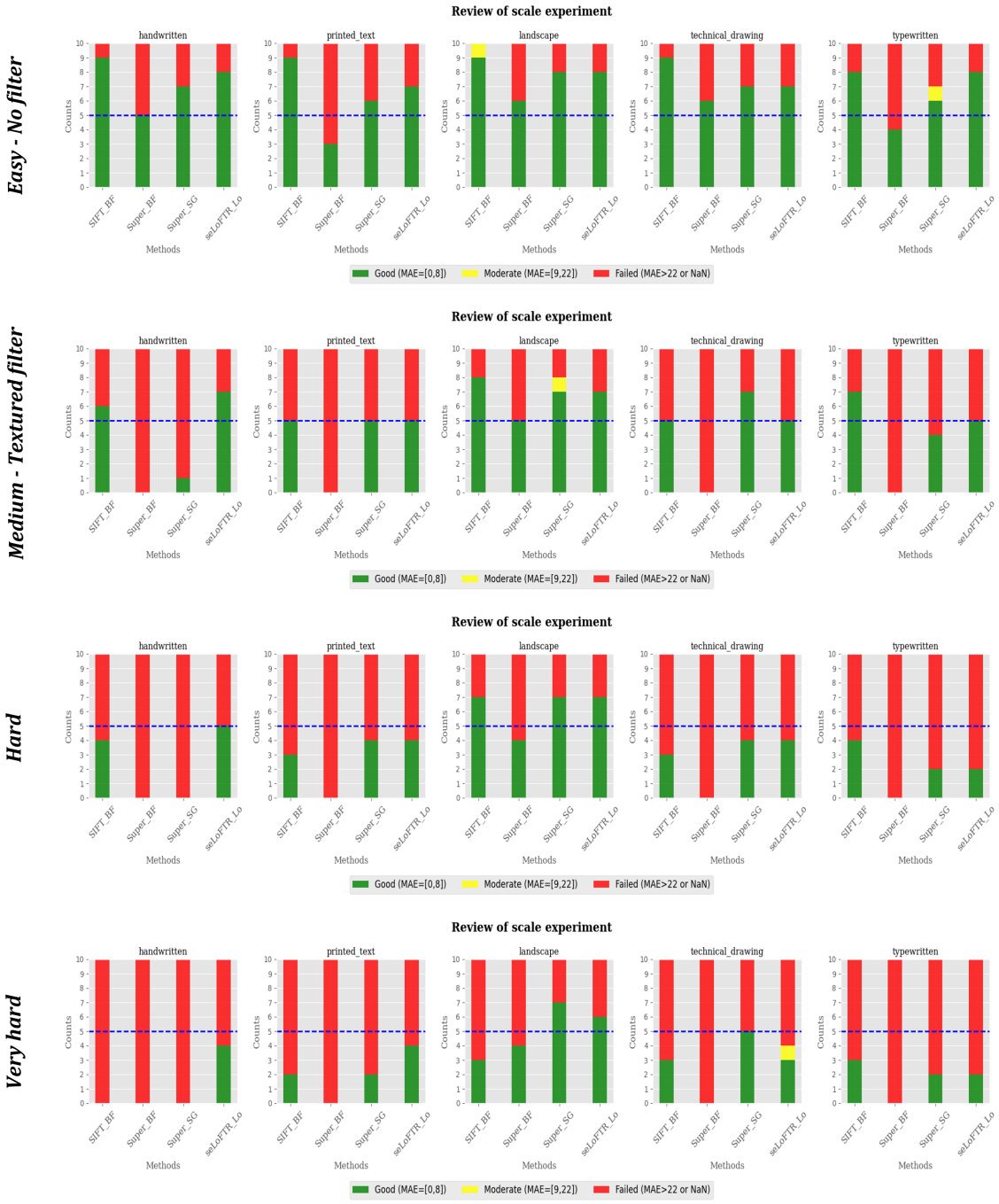
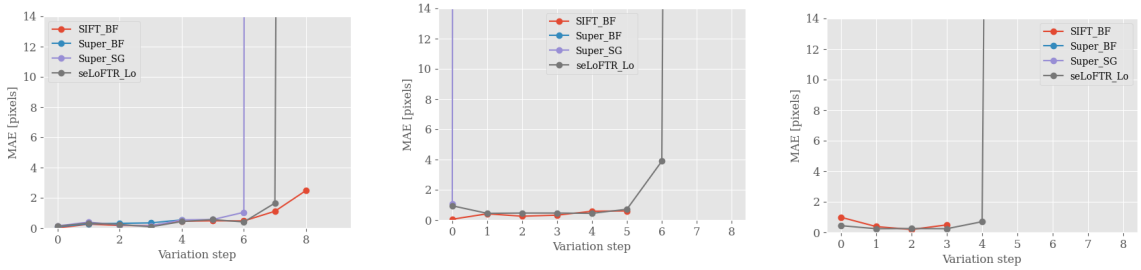


Figure 5.5: **Results of the Scale Robustness Experiment**. The graphs illustrate the performance of each method under different damage levels and document types. For each damage level and document type, the graphs depict the number of times each method reconstructed the fragment across ten levels of downsizing, categorizing the results as 'Good (MAE = [0,8])' in green, 'Moderate (MAE = [9,22])' in yellow or 'Failed (MAE>22 or NaN)' in red.



(a) Easy - No filter (b) Medium - Textured filter (c) Hard

Figure 5.6: **Scale Robustness Experiment: Handwritten document, MAE graphs**. (a)Easy - No filter. SIFT effectively reconstructs the snippet up to level 9 of downsize (90% reduction), followed by SE2-LoFTR, which works up to level 8, and SP+SG, effective only until level 6. (b)Medium - Textured filter. SE2-LoFTR outperforms SIFT by reconstructing the snippet up to level 6 of downsize. (c)Hard. SE2-LoFTR manages to reconstruct the hardly damaged snippet up to level 4 of downsize, followed by SIFT, which stops at level 3; the other methods fail in every level.

struggled in the smallest sizes. SuperPoint + SuperGlue outperformed SuperPoint + NN, reconstructing the snippets over half of the times for all document types. Figure 5.6a shows the MAE graph across all downsize levels for the handwritten document.

Medium - Textured filter

The medium filter level saw an increase in failures. SuperPoint + NN only managed to reconstruct the landscape image effectively (50% of the downsize levels), failing 100% of the levels in all other document types. SIFT remained the top performer, with SE2-LoFTR close behind, despite difficulties with printed text and technical drawing. SuperPoint + SuperGlue showed more effectiveness than SIFT and SE2-LoFTR in technical drawing but struggled a lot with the handwritten document, as can be seen in Figure 5.6b. The landscape image proved to be the easiest for all methods, with SIFT handling up to 80% downsizing.

Hard

At the hard filter level, SuperPoint + NN failed across almost all types and scales, except in the landscape document, where it managed to reconstruct 40% of the downsize levels. SE2-LoFTR edged out SIFT in the handwritten, printed text, and technical drawing documents, as shown in Figure 5.6c. SuperPoint + SuperGlue provided good performance, similar to SE2-LoFTR in all document types, but failed completely in the handwritten document.

Very Hard

In the very hard scenario, all methods faced significant challenges. SE2-LoFTR had the best overall performance, with SuperPoint + SuperGlue as the second best. The handwritten document was the toughest, with SE2-LoFTR the only method handling up to 40% downsizing. The landscape was the easiest, where SuperPoint + SuperGlue slightly outperformed SE2-LoFTR by correctly reconstructing snippets 70% of the downsize levels. SIFT struggled with all document types at this damage level.

Overall performance

In summary, the landscape document emerged as the easiest to reconstruct across all levels of downsizing and filtering as shown in Table 5.9. This consistency underscores its relative

simplicity compared to other document types. In contrast, documents with highly repetitive patterns presented the most significant challenges. Among these, the handwritten document proved to be the most difficult to reconstruct, followed by the typewritten and then printed text documents in terms of difficulty. SE2-LoFTR effectively handled images with repetitive patterns across all filter levels - as revealed by Table 5.8, demonstrating its robustness. SIFT, while second-best for handwritten and printed text, performed slightly better for typewritten documents. SuperPoint + NN was effective only in the no-filter scenario; it failed to perform adequately in all the other damage levels, except with the landscape document. As the level of filter damage increased, SIFT's scale invariance diminished - as seen in Figure 5.6 and revealed by Table 5.7, whereas SE2-LoFTR showed better adaptability in handling the more challenging hard and very hard damage categories.

Filter Level	**Overall best performance**
No Filter	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Medium - textured	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Hard	**SE2-LoFTR** > SIFT+NN > SP+SG > SP+NN
Very hard	**SE2-LoFTR** > SP+SG > SIFT+NN > SP+NN

Table 5.7: **Scale Robustness Experiment: overall performance by filter level**.

Document type	**Overall best performance**
Handwritten	**SE2-LoFTR** > SIFT+NN > SP+SG > SP+NN
Printed text	**SE2-LoFTR** > SIFT+NN > SP+SG > SP+NN
Typewritten	**SIFT+NN** > SE2-LoFTR > SP+SG > SP+NN
Technical drawing	**SP+SG** > SIFT+NN > SE2-LoFTR > SP+NN
Landscape	**SP+SG** > SE2-LoFTR > SIFT+NN > SP+NN

Table 5.8: **Scale Robustness Experiment: overall performance by document type**.

Document type	**Failure rate**
Handwritten	52 %
Printed text	50.5 %
Typewritten	51 %
Technical drawing	45.5 %
Landscape	27.5 %

Table 5.9: **Scale Robustness Experiment: failure rate by document type**. In red the document type with the highest failure percentage.

5.4 Discussion

The investigation conducted in this thesis has yielded critical insights into the performance of the implemented local feature matching methods under different document types and damage conditions. These findings not only demonstrate the potential for significantly enhancing the reconstruction functionality of the FalKe system but also reveal specific strengths and weaknesses of each method.

A key finding of this investigation is the impact of document type variability on the success of feature matching. The landscape document, with its distinct and varied features,

consistently emerged as the easiest to reconstruct in every experiment. In stark contrast, the most challenging to reconstruct was the handwritten document, due to its complex, repetitive pattern, which became increasingly challenging under additional damage. When dealing with documents that feature highly repetitive patterns such as handwritten, typewritten, and printed text:

- the Noise Robustness Experiment highlighted SIFT and SuperPoint + SuperGlue as the top performers across various levels of damage.
- in the Rotation and Scale Robustness Experiments, SIFT and SE2-LoFTR proved more effective, showcasing their robustness in these challenging scenarios.

The degree of damage to the documents' snippets also played a significant role in determining the success of the matching methods, a consideration especially pertinent in the context of cultural heritage where fragments often exhibit extensive damage. In the hard and very hard scenarios, ergo in the highest levels of filtering:

- in the Noise Robustness Experiment, SuperPoint + SuperGlue led the way, with SE2-LoFTR as a close second.
- in the Rotation Robustness Experiment, both SE2-LoFTR and SIFT showed superior performance.
- in the Scale Robustness Experiment, SE2-LoFTR excelled.

It was conclusively demonstrated that the current method used in the FalKe system, SuperPoint + NN, is the weakest performer in all the experiments and should be replaced to improve the system's reconstruction performance. This finding is pivotal, suggesting a clear direction for enhancing the system through the adoption of more robust feature matching techniques.

In the context of the FalKe system's application to cultural heritage, where fragments are often heavily damaged, and their initial orientation is unknown, the most crucial factors identified were robustness against rotation and tolerance to document damage. While noise is a controllable parameter, particularly with good camera and lighting setup, and scale can be managed or compensated for, the ability to handle rotation and severe damage is vital. Therefore, methods like SuperPoint + SuperGlue, which are effective in noisy environments but lack adaptability in dealing with rotation and severe damage, are less critical for the FalKe system, while SIFT and SE2-LoFTR gain importance.

SE2-LoFTR and **SIFT** have emerged as the definitive choices for the FalKe system, showcasing magnificent adaptability to different types of documents and varying conditions of noise, rotation, and scale. The traditional SIFT algorithm, in particular, stands out with its exceptional robustness in less damaged conditions, making it the go-to choice for the reconstruction process. SE2-LoFTR, with its remarkable ability to tackle heavily damaged fragments, becomes indispensable in cultural heritage contexts, where documents often show damage due to aging. Its prowess in managing severe discrepancies in document appearance solidifies its position as a critical tool for challenging reconstruction tasks.

6 Conclusion

6.1 Summary

This thesis successfully explored solutions to enhance the overall effectiveness of the FalKe system in real-time reconstruction of fragmented documents. Key accomplishments of this work include:

- Analysis of the FalKe system's limitations and requirements.
- Addition of a lighting system with adjustable intensity.
- Design and implementation of a Camera-Projector Calibration process in Python.
- Design and implementation of a Digitisation and Storage System for reference documents in Python.
- Selection and implementation of four relevant local feature matching methods in Python.
- Experimental determination of the best local feature matching methods to improve the robustness and accuracy of document reconstruction in the FalKe system.

In the hardware setup a uniform lighting system with adjustable intensity was added to ensure consistent performance of the system under different environments. To address the need for an easy yet precise Camera-Projector Calibration, a three-step process was designed and implemented, utilizing homography. This enabled direct mapping from the camera to the projector, thus enhancing the user interface, which is crucial for guiding users through document reconstruction. Additionally, a Digitization and Storage System was developed to manage and store essential data from multiple reference documents, facilitating easy integration and retrieval of precomputed key data.

The primary contribution of this thesis was the comprehensive investigation of four local feature matching methods: traditional detector-based SIFT+NN, deep learning-based SuperPoint+NN and SuperPoint+SuperGlue, and the detector-free SE2-LoFTR. The goal was to enhance accuracy in recognizing and correctly positioning fragments, seeking a better method than the original SuperPoint+NN. The evaluation of each method's reconstruction accuracy was methodically conducted in three experiments, assessing their robustness against noise, rotation, and scale when matching fragments with varying degrees of damage.

Five different document types were selected to create a dataset of damaged fragments by applying filters. The experimental results highlighted the limited robustness of the initial method across all tests and damage levels, while SIFT and SE2-LoFTR proved to be the best for the needs of the FalKe system due to their robust handling of noise, rotation, and scale across various damage levels. Specifically, SE2-LoFTR was particularly effective in reconstructing very damaged fragments, while SIFT excelled in less damaged conditions.

The presented improvements successfully meet the requirements set for the FalKe system during the design process and collectively enhance the system as an assistance tool in the document reconstruction task.

6.2 Critical considerations

A critical aspect of designing the experimental pipeline was understanding how to accurately evaluate the reconstruction of fragmented documents. Initially, the intention was to digitize

real-world paper documents to use as reference documents and manually rip them to create the snippets dataset. However, this physical approach presented challenges. The digitization process, dependent on hardware, introduced limitations, such as difficulty in keeping the paper completely flat on the surface. Additionally, imprecise borders of ripped documents posed issues for the segmentation algorithm, leading to inaccuracies in the results of the reconstruction. As the focus of this thesis excluded investigation into the hardware system and segmentation algorithm, a digital approach was adopted. Instead of physically ripping documents, "digital ripping" was performed using freehand masks to create the fragments dataset. This choice completely eliminated environmental errors, such as perspective distortions from photographing fragments and irregular borders from manually ripping paper documents, enabling the focus of the evaluation to be solely on the accuracy of the local feature matching methods.

6.3 Outlook

As this thesis concludes, several promising directions for future research and development emerge:

- **GPU integration**. The learning-based local feature matching methods employed in this study have high computational times. While the absence of GPU acceleration doesn't affect the reconstruction results, it prolongs the process. Integrating GPU acceleration is recommended to expedite the reconstruction for production-ready systems.
- **Enhancing the Digitisation and Storage System**. Improving the Digitisation and Storage System for reference documents is a key area for development. Introducing a user-friendly interface would simplify the process of adding new reference documents. This interface could enable automated photographing of documents and the precomputation and storage of essential data for later use. Additionally, developing a more robust similarity measure for selecting the correct reference document from the storage during reconstruction would enhance the system's effectiveness when handling multiple reference documents.
- **Advancing text document reconstruction**.The effectiveness of the investigated local feature matching methods is notably lower for text documents compared to other document types. Future work could concentrate on incorporating advanced text recognition and feature extraction techniques to improve the accuracy of text document reconstruction.
- **Expanding to other flat 2D objects**. Broadening the application of the system to include other flat 2D objects, such as tiles, presents an exciting opportunity. This expansion could not only validate the system's versatility but also pave the way for its use in diverse fields such as industrial applications.
- **Improving the segmentation method**. Further research could explore the integration of more sophisticated AI-based segmentation models, like the *Segment Anything Model* from META AI [KMR+23]. Such technologies could greatly improve the system's ability in image segmentation, enhancing accuracy especially for hand ripped documents with irregular contours.
- **Integrating Augmented Reality Technologies**. To enhance practical applications, integrating the FalKe system with augmented reality technology could significantly revolutionize its usability. This integration would offer a more dynamic and interactive

reconstruction process, making it accessible and engaging for educational and professional purposes.

Each of these potential developments represents a step forward in the evolution of document reconstruction technology. They offer opportunities to enhance the capabilities of the FalKe system, expand its applicability, and make significant contributions to the field.

Bibliography

[ABD12] Alcantarilla, Pablo F. ; Bartoli, Adrien ; Davison, Andrew J.: KAZE Features. In: Fitzgibbon, Andrew (Hrsg.) ; Lazebnik, Svetlana (Hrsg.) ; Perona, Pietro (Hrsg.) ; Sato, Yoichi (Hrsg.) ; Schmid, Cordelia (Hrsg.): *Computer Vision – ECCV 2012*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. – ISBN 978–3–642–33783–3, S. 214–227

[AGP03] Amigoni, Francesco ; Gazzani, Stefano ; Podico, Simone: A method for reassembling fragments in image reconstruction. In: *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)* Bd. 3 IEEE, 2003, S. III–581

[ANB13] Alcantarilla, Pablo ; Nuevo, Jesus ; Bartoli, Adrien: Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In: Burghardt, T. (Hrsg.) ; Damen, D. (Hrsg.) ; Mayol-Cuevas, W. (Hrsg.) ; Mirmehdi, M. (Hrsg.): *Procedings of the British Machine Vision Conference 2013*, British Machine Vision Association, 2013. – ISBN 1–901725–49–9, S. 13.1–13.11

[AZH+21] Alzubaidi, Laith ; Zhang, Jinglan ; Humaidi, Amjad J. ; Al-Dujaili, Ayad ; Duan, Ye ; Al-Shamma, Omran ; Santamaría, J. ; Fadhel, Mohammed A. ; Al-Amidie, Muthana ; Farhan, Laith: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In: *Journal of Big Data* 8 (2021), Nr. 1, S. 53. http://dx.doi.org/10.1186/s40537-021-00444-8. – DOI 10.1186/s40537–021–00444–8. – ISSN 2196–1115

[BB93] Bunke, H ; Bühler, U: Applications of approximate string matching to 2D shape recognition. In: *Pattern Recognition* 26 (1993), Nr. 12, 1797-1812. http://dx.doi.org/https://doi.org/10.1016/0031-3203(93)90177-X. – DOI https://doi.org/10.1016/0031–3203(93)90177–X. – ISSN 0031–3203

[BETV08] Bay, Herbert ; Ess, Andreas ; Tuytelaars, Tinne ; Van Gool, Luc: Speeded-Up Robust Features (SURF). In: *Computer Vision and Image Understanding* 110 (2008), Nr. 3, 346-359. http://dx.doi.org/https://doi.org/10.1016/j.cviu.2007.09.014. – DOI https://doi.org/10.1016/j.cviu.2007.09.014. – ISSN 1077–3142. – Similarity Matching in Computer Vision and Multimedia

[BK22] Bökman, Georg ; Kahl, Fredrik: *A case for using rotation invariant features in state of the art feature matchers*. 2022

[BKC17] Badrinarayanan, Vijay ; Kendall, Alex ; Cipolla, Roberto: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), Nr. 12, S. 2481–2495. http://dx.doi.org/10.1109/TPAMI.2016.2644615. – DOI 10.1109/TPAMI.2016.2644615

[BKF22] Bökman, Georg ; Kahl, Fredrik ; Flinth, Axel: *ZZ-Net: A Universal Rotation Equivariant Architecture for 2D Point Clouds*. 2022

[BLM+17] Bian, Jiawang ; Lin, Wen-Yan ; Matsushita, Yasuyuki ; Yeung, Sai-Kit ; Nguyen, Tan-Dat ; Cheng, Ming-Ming: GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, S. 2828–2837

[BW89] Burdea, B.G. ; Wolfson, H.J.: Solving jigsaw puzzles by a robot. In: *IEEE Transactions on Robotics and Automation* 5 (1989), Nr. 6, S. 752–764. http://dx.doi.org/10.1109/70.88097. – DOI 10.1109/70.88097

[CLSF10] Calonder, Michael ; Lepetit, Vincent ; Strecha, Christoph ; Fua, Pascal: Brief: Binary robust independent elementary features. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11* Springer, 2010, S. 778–792

[CLZ+22] Chen, Hongkai ; Luo, Zixin ; Zhou, Lei ; Tian, Yurun ; Zhen, Mingmin ; Fang, Tian ; Mckinnon, David ; Tsin, Yanghai ; Quan, Long: Aspanformer: Detector-free image matching with adaptive span transformer. In: *European Conference on Computer Vision* Springer, 2022, S. 20–36

[CW16a] Cohen, Taco S. ; Welling, Max: Group Equivariant Convolutional Networks. In: *CoRR* abs/1602.07576 (2016). http://arxiv.org/abs/1602.07576

[CW16b] Cohen, Taco S. ; Welling, Max: *Steerable CNNs*. 2016

[DMR17] DeTone, Daniel ; Malisiewicz, Tomasz ; Rabinovich, Andrew: *Toward Geometric Deep SLAM*. 2017

[DMR18] DeTone, Daniel ; Malisiewicz, Tomasz ; Rabinovich, Andrew: SuperPoint: Self-Supervised Interest Point Detection and Description. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 18/06/2018 - 22/06/2018. – ISBN 978–1–5386–6100–0, S. 337–33712

[DS09] De Smet, Patrick: Semi-automatic Forensic Reconstruction of Ripped-up Documents. In: *2009 10th International Conference on Document Analysis and Recognition*, 2009, S. 703–707

[DV18] Dumoulin, Vincent ; Visin, Francesco: *A guide to convolution arithmetic for deep learning*. 2018

[FB81] Fischler, Martin A. ; Bolles, Robert C.: Random sample consensus. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395. http://dx.doi.org/10.1145/358669.358692. – DOI 10.1145/358669.358692. – ISSN 0001–0782

[FG64] Freeman, H. ; Garder, L.: Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition. In: *IEEE Transactions on Electronic Computers* EC-13 (1964), Nr. 2, S. 118–127. http://dx.doi.org/10.1109/PGEC.1964.263781. – DOI 10.1109/PGEC.1964.263781. – ISSN 0367–7508

[FH04] Fulton, William ; Harris, Joe: *Representation Theory*. Bd. 129. New York, NY : Springer New York, 2004. http://dx.doi.org/10.1007/978-1-4612-0979-9. http://dx.doi.org/10.1007/978-1-4612-0979-9. – ISBN 978–3–540–00539–1

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016. – http://www.deeplearningbook.org

[GDDM16] Girshick, Ross ; Donahue, Jeff ; Darrell, Trevor ; Malik, Jitendra: Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (2016), Nr. 1, S. 142–158. http://dx.doi.org/10.1109/TPAMI.2015.2437384. – DOI 10.1109/TPAMI.2015.2437384

[GLS02] Gama Leitao, H.C. da ; Stolfi, J.: A multiscale method for the reassembly of two-dimensional fragmented objects. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), Nr. 9, S. 1239–1251. http://dx.doi.org/10.1109/TPAMI.2002.1033215. – DOI 10.1109/TPAMI.2002.1033215

[GMB02] Goldberg, David ; Malon, Christopher ; Bern, Marshall: A global approach to automatic solution of jigsaw puzzles. In: *Proceedings of the eighteenth annual symposium on Computational geometry*, 2002, S. 82–87

[GZL+19] Gao, Yuan ; Zhang, Zheng-Dong ; Li, Shuo ; Guo, Yu-Ting ; Wu, Qing-Yao ; Liu, Shu-Hao ; Yang, Shu-Jian ; Ding, Lei ; Zhao, Bao-Chun ; Li, Shuai u. a.: Deep neural network-assisted computed tomography diagnosis of metastatic lymph nodes from gastric cancer. In: *Chinese medical journal* 132 (2019), Nr. 23, S. 2804–2811

[HYL18] Hamilton, William L. ; Ying, Rex ; Leskovec, Jure: *Inductive Representation Learning on Large Graphs*. 2018

[HZ03] Hartley, Richard ; Zisserman, Andrew: Multiple view geometry in computer vision. (2003)

[JOF06] Justino, Edson ; Oliveira, Luiz S. ; Freitas, Cinthia: Reconstructing shredded documents through feature matching. In: *Forensic Science International* 160 (2006), Nr. 2, 140-147. http://dx.doi.org/https://doi.org/10.1016/j.forsciint.2005.09.001. – DOI https://doi.org/10.1016/j .forsciint.2005.09.001. – ISSN 0379–0738

[JSTL19] Jiang, Bo ; Sun, Pengfei ; Tang, Jin ; Luo, Bin: *GLMNet: Graph Learning-Matching Networks for Feature Matching*. 2019

[JTH+21] Jiang, Wei ; Trulls, Eduard ; Hosang, Jan ; Tagliasacchi, Andrea ; Yi, Kwang M.: *COTR: Correspondence Transformer for Matching Across Images*. 2021

[KDB+94] Kosiba, David A. ; Devaux, Pierre M. ; Balasubramanian, Sanjay ; Gandhi, Tarak ; Kasturi, Rangachar: An automatic jigsaw puzzle solver. In: *Proceedings of 12th International Conference on Pattern Recognition* 1 (1994), 616-618 vol.1. https://api.semanticscholar.org/CorpusID:46943401

[KMR+23] Kirillov, Alexander ; Mintun, Eric ; Ravi, Nikhila ; Mao, Hanzi ; Rolland, Chloe ; Gustafson, Laura ; Xiao, Tete ; Whitehead, Spencer ; Berg, Alexander C. ; Lo, Wan-Yen ; Dollár, Piotr ; Girshick, Ross: *Segment Anything*. 2023

[KSH12] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E.: ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in neural information processing systems* 25 (2012)

[LAE+16] Liu, Wei ; Anguelov, Dragomir ; Erhan, Dumitru ; Szegedy, Christian ; Reed, Scott ; Fu, Cheng-Yang ; Berg, Alexander C.: Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* Springer, 2016, S. 21–37

[LCS11] Leutenegger, Stefan ; Chli, Margarita ; Siegwart, Roland Y.: BRISK: Binary Robust invariant scalable keypoints. In: *2011 International Conference on Computer Vision*, 2011, S. 2548–2555

[LCY11] Liu, Hairong ; Cao, Shengjiao ; Yan, Shuicheng: Automated Assembly of Shredded Pieces From Multiple Photos. In: *IEEE Transactions on Multimedia* 13 (2011), 10, S. 1154–1162. http://dx.doi.org/10.1109/ICME.2010.5582544. – DOI 10.1109/ICME.2010.5582544

[LDZ+10] Lin, Hui ; Du, Peijun ; Zhao, Weichang ; Zhang, Lianpeng ; Sun, Huasheng: Image registration based on corner detection and affine transformation. In: *2010 3rd International Congress on Image and Signal Processing* Bd. 5, 2010, S. 2184–2188

[LEM+19] Leoni, Guto ; Endo, Patricia ; Monteiro, Kayo ; Rocha, Elisson ; Silva, Ivanovitch ; Lynn, Theodore: Accelerometer-Based Human Fall Detection Using Convolutional Neural Networks. In: *Sensors* 19 (2019), 04, S. 1644. http://dx.doi.org/10.3390/s19071644. – DOI 10.3390/s19071644

[LHLP20] Li, Xinghui ; Han, Kai ; Li, Shuda ; Prisacariu, Victor A.: *Dual-Resolution Correspondence Networks*. 2020

[Low04] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *Int. J. Comput. Vision* 60 (2004), nov, Nr. 2, 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94. – DOI 10.1023/B:VISI.0000029664.99615.94. – ISSN 0920–5691

[LSD15] Long, Jonathan ; Shelhamer, Evan ; Darrell, Trevor: Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, S. 3431–3440

[LZB+20] Luo, Zixin ; Zhou, Lei ; Bai, Xuyang ; Chen, Hongkai ; Zhang, Jiahui ; Yao, Yao ; Li, Shiwei ; Fang, Tian ; Quan, Long: *ASLFeat: Learning Local Features of Accurate Shape and Localization*. 2020

[LZZC14] Li, Hongsheng ; Zheng, Yuanjie ; Zhang, Shaoting ; Cheng, Jian: Solving a Special Type of Jigsaw Puzzles: Banknote Reconstruction From a Large Number of Fragments. In: *IEEE Transactions on Multimedia* 16 (2014), Nr. 2, S. 571–578. http://dx.doi.org/10.1109/TMM.2013.2291968. – DOI 10.1109/TMM.2013.2291968

[MF13] Marques, M.A.O. ; Freitas, C.O.A.: Document Decipherment-restoration: Strip-shredded Document Reconstruction based on Color. In: *IEEE Latin America Transactions* 11 (2013), Nr. 6, S. 1359–1365. http://dx.doi.org/10.1109/TLA.2013.6710384. – DOI 10.1109/TLA.2013.6710384

[MK03] McBride, Jonah C. ; Kimia, Benjamin B.: Archaeological Fragment Reconstruction Using Curve-Matching. In: *2003 Conference on Computer Vision and Pattern Recognition Workshop*, IEEE, 16/06/2003 - 22/06/2003, S. 3

[ML09] Muja, Marius ; Lowe, David G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: *VISAPP (1)* 2 (2009), Nr. 331-340, S. 2

[MLS23] Ma, Chang-Hsian ; Lu, Chien-Liang ; Shih, Huang-Chia: Vision-Based Jigsaw Puzzle Solving with a Robotic Arm. In: *Sensors* 23 (2023), Nr. 15. http://dx.doi.org/10.3390/s23156913. – DOI 10.3390/s23156913. – ISSN 1424–8220

[MTS+19] Melekhov, Iaroslav ; Tiulpin, Aleksei ; Sattler, Torsten ; Pollefeys, Marc ; Rahtu, Esa ; Kannala, Juho: Dgc-net: Dense geometric correspondence network. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* IEEE, 2019, S. 1034–1042

[PEP+08] Papaodysseus, Constantin ; Exarhos, Mihalis ; Panagopoulos, Mihalis ; Rousopoulos, Panayiotis ; Triantafillou, Constantin ; Panagopoulos, Thanasis: Image and pattern analysis of 1650 BC wall paintings and reconstruction. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38 (2008), Nr. 4, S. 958–965

[PLOP20] Pautrat, Rémi ; Larsson, Viktor ; Oswald, Martin R. ; Pollefeys, Marc: *Online Invariance Selection for Local Feature Descriptors*. 2020

[PPE+02] Panagopoulos, Th ; Papaodysseus, Constantin ; Exarhos, Mihalis ; Alexiou, C. ; Rousopoulos, George: Automated reconstruction of fragmented, 1600 B.C. wall paintings. In: *Recent Advances in Circuits, Systems and Signal Processing* (2002), 01, S. 396–401

[RCA+18] Rocco, Ignacio ; Cimpoi, Mircea ; Arandjelović, Relja ; Torii, Akihiko ; Pajdla, Tomas ; Sivic, Josef: *Neighbourhood Consensus Networks*. 2018

[RD06] Rosten, Edward ; Drummond, Tom: Machine learning for high-speed corner detection. In: *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* Springer, 2006, S. 430–443

[RRKB11] Rublee, Ethan ; Rabaud, Vincent ; Konolige, Kurt ; Bradski, Gary: ORB: An efficient alternative to SIFT or SURF. In: *2011 International Conference on Computer Vision*, IEEE, 06/11/2011 - 13/11/2011. – ISBN 978–1–4577–1102–2, S. 2564–2571

[RWS+19] Revaud, Jerome ; Weinzaepfel, Philippe ; Souza, César D. ; Pion, Noe ; Csurka, Gabriela ; Cabon, Yohann ; Humenberger, Martin: *R2D2: Repeatable and Reliable Detector and Descriptor*. 2019

[SDMR20] Sarlin, Paul-Edouard ; DeTone, Daniel ; Malisiewicz, Tomasz ; Rabinovich, Andrew: Superglue: Learning feature matching with graph neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, S. 4938–4947

[SE06] Sagiroglu, M.S. ; Ercil, A.: A Texture Based Matching Approach for Automated Assembly of Puzzles. In: *18th International Conference on Pattern Recognition (ICPR'06)* Bd. 3, 2006, S. 1036–1041

[SF16] Sizikova, Elena ; Funkhouser, Thomas A.: Wall Painting Reconstruction Using a Genetic Algorithm. In: *Journal on Computing and Cultural Heritage (JOCCH)* 11 (2016), 1 - 17. https://api.semanticscholar.org/CorpusID:894723

[SGT+09] Scarselli, Franco ; Gori, Marco ; Tsoi, Ah C. ; Hagenbuchner, Markus ; Monfardini, Gabriele: The Graph Neural Network Model. In: *IEEE Transactions on Neural Networks* 20 (2009), Nr. 1, S. 61–80. http://dx.doi.org/10.1109/TNN.2008.2005605. – DOI 10.1109/TNN.2008.2005605

[SJT+21] Sun, Weiwei ; Jiang, Wei ; Trulls, Eduard ; Tagliasacchi, Andrea ; Yi, Kwang M.: *ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning*. 2021

[SSW+21] Sun, Jiaming ; Shen, Zehong ; Wang, Yuang ; Bao, Hujun ; Zhou, Xiaowei: LoFTR: Detector-Free Local Feature Matching with Transformers. In: *CoRR* abs/2104.00680 (2021). https://arxiv.org/abs/2104.00680

[Sze11] Szeliski, Richard: *Computer Vision*. London : Springer London, 2011. http://dx.doi.org/10.1007/978-1-84882-935-0. http://dx.doi.org/10.1007/978-1-84882-935-0. – ISBN 978–1–84882–934–3

[TDT20] Truong, Prune ; Danelljan, Martin ; Timofte, Radu: GLU-Net: Global-local universal network for dense flow and correspondences. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, S. 6258–6268

[TP10] Tsamoura, Efthymia ; Pitas, Ioannis: Automatic Color Based Reassembly of Fragmented Images and Paintings. In: *IEEE Transactions on Image Processing* 19 (2010), Nr. 3, S. 680–690. http://dx.doi.org/10.1109/TIP.2009.2035840. – DOI 10.1109/TIP.2009.2035840

[TS18] Tareen, Shaharyar Ahmed K. ; Saleem, Zahra: A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, S. 1–10

[WC21] Weiler, Maurice ; Cesa, Gabriele: *General $E(2)$-Equivariant Steerable CNNs*. 2021

[WGW+18] Weiler, Maurice ; Geiger, Mario ; Welling, Max ; Boomsma, Wouter ; Cohen, Taco: *3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data*. 2018

[WHS18] Weiler, Maurice ; Hamprecht, Fred A. ; Storath, Martin: Learning Steerable Filters for Rotation Equivariant CNNs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, S. 849–858

[WPC+21] Wu, Zonghan ; Pan, Shirui ; Chen, Fengwen ; Long, Guodong ; Zhang, Chengqi ; Yu, Philip S.: A Comprehensive Survey on Graph Neural Networks.

In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2021), Januar, Nr. 1, 4–24. http://dx.doi.org/10.1109/tnnls.2020.2978386. – DOI 10.1109/tnnls.2020.2978386. – ISSN 2162–2388

[WSKL88] Wolfson, Haim ; Schonberg, Edith ; Kalvin, Alan ; Lamdan, Yehezkel: Solving jigsaw puzzles by computer. In: *Annals of Operations Research* 12 (1988), Nr. 1, S. 51–64. http://dx.doi.org/10.1007/BF02186360. – DOI 10.1007/BF02186360. – ISSN 1572–9338

[WSL+21] Wang, Jianfeng ; Song, Lin ; Li, Zeming ; Sun, Hongbin ; Sun, Jian ; Zheng, Nanning: *End-to-End Object Detection with Fully Convolutional Network*. 2021

[YM11] Yu, Guoshen ; Morel, Jean-Michel: ASIFT: An Algorithm for Fully Affine Invariant Comparison. In: *Image Processing On Line* 1 (2011), 02. http://dx.doi.org/10.5201/ipol.2011.my-asift. – DOI 10.5201/ipol.2011.my–asift

[YTLF16] Yi, Kwang M. ; Trulls, Eduard ; Lepetit, Vincent ; Fua, Pascal: *LIFT: Learned Invariant Feature Transform*. 2016

[YXG+20] Yu, Donghang ; Xu, Qing ; Guo, Haitao ; Zhao, Chuan ; Lin, Yuzhun ; Li, Daoji: An efficient and lightweight convolutional neural network for remote sensing image scene classification. In: *Sensors* 20 (2020), Nr. 7, S. 1999

[YY21] Yang, Ruixin ; Yu, Yingyan: Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis. In: *Frontiers in Oncology* 11 (2021). http://dx.doi.org/10.3389/fonc.2021.638182. – DOI 10.3389/fonc.2021.638182. – ISSN 2234–943X

[ZF14] Zeiler, Matthew D. ; Fergus, Rob: Visualizing and Understanding Convolutional Networks. In: Fleet, David (Hrsg.) ; Pajdla, Tomas (Hrsg.) ; Schiele, Bernt (Hrsg.) ; Tuytelaars, Tinne (Hrsg.): *Computer Vision – ECCV 2014*. Cham : Springer International Publishing, 2014. – ISBN 978–3–319–10590–1, S. 818–833

[ZL14] Zhang, Kang ; Li, Xin: A Graph-Based Optimization Algorithm for Fragmented Image Reassembly. In: *Graph. Models* 76 (2014), sep, Nr. 5, 484–495. http://dx.doi.org/10.1016/j.gmod.2014.03.001. – DOI 10.1016/j.gmod.2014.03.001. – ISSN 1524–0703

[ZSQ+17] Zhao, Hengshuang ; Shi, Jianping ; Qi, Xiaojuan ; Wang, Xiaogang ; Jia, Jiaya: Pyramid scene parsing network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 2881–2890

[ZXZ+21] Zhong, Zhen ; Xiao, Guobao ; Zheng, Linxin ; Lu, Yan ; Ma, Jiayi: T-Net: Effective Permutation-Equivariant Network for Two-View Correspondence Learning. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, S. 1930–1939

[ZZH08] Zhu, Liangjia ; Zhou, Zongtan ; Hu, Dewen: Globally Consistent Reconstruction of Ripped-Up Documents. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), Nr. 1, S. 1–13. http://dx.doi.org/10.1109/TPAMI.2007.1163. – DOI 10.1109/TPAMI.2007.1163

Annex

1 Fragment reassembly methods without reference image

The challenge of fragment reassembly without a reference image initially manifested in solving jigsaw puzzles, a field that saw its earliest automated solutions with Freeman and Gardner's work in 1964. Their method focused on partial curve matching for puzzle piece edges, without utilizing pictorial data [FG64]. Subsequent initial approaches mainly relied on the shape information of jigsaw pieces to identify their adjacent counterparts, strongly relying on the regularity and well-defined corners and shapes of the puzzle pieces [WSKL88, BW89, BB93]. In [WSKL88], the authors developed a method that starts by matching the border pieces of the puzzle, applying a traveling salesman problem approach, and then focuses on assembling the central pieces using a combination of best-first and backtracking methods. Building on this, Goldberg et al. [GMB02] enhanced this strategy by introducing new techniques that lead to a more efficient assembly process. Over time, this focus expanded to include color information, exemplified by D. Kosiba et al. [KDB+94] who first integrated color with shape, enhancing the applicability of these reassembly methods in fields like archaeology, forensics, and art restoration.

In archaeology and cultural heritage restoration, the irregular shapes of fragments and lack of original images necessitate robust algorithmic solutions for artifact reassembly. For instance, Leitão et al. [GLS02] developed a multiscale method particularly effective for two-dimensional fragments, focusing on curvature-encoded outlines and utilizing an incremental dynamic programming sequence matching algorithm. This technique has been successfully applied to ceramic fragments and wall paintings and can be applied to other objects with curved surfaces such as pottery. In [PPE+02] a method to reconstruct the outstanding wall paintings of the Greek island Thera was proposed. Here each fragment was digitally captured, its contour obtained and all fragment contours were then compared accordingly. Similarly, Papaodysseus et al. [PEP+08] addressed the reconstruction of severely damaged wall paintings at Akrotiri on the Greek island of Thera, using color image segmentation and pattern analysis to piece together thousands fragments. Sizikova and Funkhouser [SF16] introduced an innovative approach using a genetic algorithm for reconstructing wall paintings, which efficiently handles noisy data by optimizing fragment placements and filtering out mismatches. Tsamoura et al. [TP10] proposed a four-step method for color-based 2-D image fragment reassembly. It begins with identifying adjacent fragments, followed by matching their boundary segments. Then, they align these boundaries and finally reassemble the image.

In forensic science, document reconstruction is a prominent area, dealing with cases like strip-shredded or hand-shredded documents, where semi-automatic solutions are often favored for their ability to allow manual corrections. De Smet [DS09] developed a toolset for reconstructing ripped-up documents interactively, starting with the outer frames and gradually filling in with non-border fragments. In [JOF06] a technique for reassembling shredded documents in forensic investigations was developed. Each fragment's boundary is simplified into a polygon shape, reducing the computational effort needed for matching. The method then employs a greedy algorithm to piece together the fragments sequentially. In the broader scope of document reconstruction, the prevalent strategies emphasize analyzing the shape and

texture of fragments, particularly around their edges, and utilize color data from the boundary pixels to facilitate the matching process between fragments. Marques and Freitas [MF13] developed a technique for reassembling strip-shredded documents by feature matching. They analyze the colors at the edges of the fragments using color models. Then, they use the NN Algorithm to compare and match these color patterns based on their similarities. Amigoni et al. [AGP03] use color content from fragment boundaries for pairwise matching. Sagiroglu and Ercil [SE06] extend fragment content using image inpainting and match pieces with a registration algorithm. Other methods only use the geometry of the fragments' contours to find matches. Zhu et al. [ZZH08] utilize contour information, approximating ripped piece contours with polygons and using their turning angles for matching. Liu et al. [LCY11] propose a method for automatically piecing together shredded fragments from various photographs. Their approach involves calculating a matching score for each pair of pieces, using both their shapes and visual information. Based on these scores, the fragments are grouped into clusters. The final assembly is achieved by constructing a spanning tree for each of these clusters. The robustness of this method to material loss and missing pieces is proven by the experimental results. Similarly, Zhang and Li [ZL14] proposed a method for reconstructing fragmented images that involves three key phases: first, establishing pairwise matches between two image pieces; second, employing a graph-based approach for the overall reassembly of these fragments; and third, refining the reassembled image through graph optimization techniques.

2 Digitisation and Storage system - Annex

Methods	**Header of CSV file**
SIFT + NN	Image Name,Keypoints,Descriptor,Image
SuperPoint + NN	Image Name, Keypoints (0 degrees), Descriptor (0 degrees), Image processed (0 degrees), Keypoints (90 degrees), Descriptor (90 degrees), Image processed (90 degrees), Keypoints (180 degrees), Descriptor (180 degrees), Image processed (180 degrees), Keypoints (270 degrees), Descriptor (270 degrees), Image processed (270 degrees)
SuperPoint + SuperGlue	Image Name, Keypoints (0 degrees), Descriptor (0 degrees), Scores (0 degrees), Image tensor (0 degrees), Keypoints (90 degrees), Descriptor (90 degrees), Scores (90 degrees), Image tensor (90 degrees), Keypoints (180 degrees), Descriptor (180 degrees), Scores (180 degrees), Image tensor (180 degrees), Keypoints (270 degrees), Descriptor (270 degrees), Scores (270 degrees), Image tensor (270 degrees)
Se2LoFTR	Image Name,Image tensor

Table 1: Different headers of CSV storage file for the implemented methods

3 Results - Annex

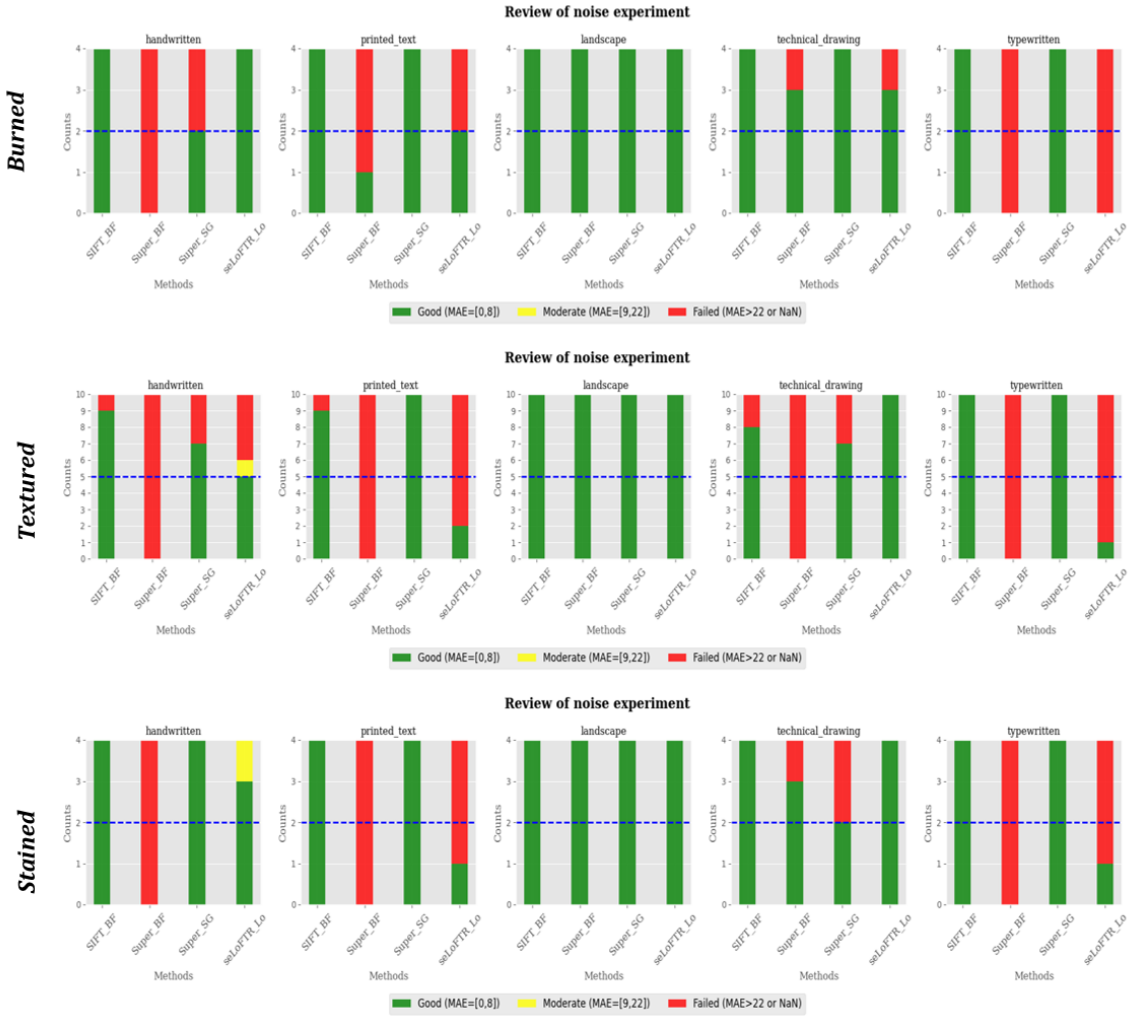


Figure 1: **Results of the Noise Robustness Experiment - Medium damage level**. These graphs give an overview of the results on Burned, Textured and Stained snippets.

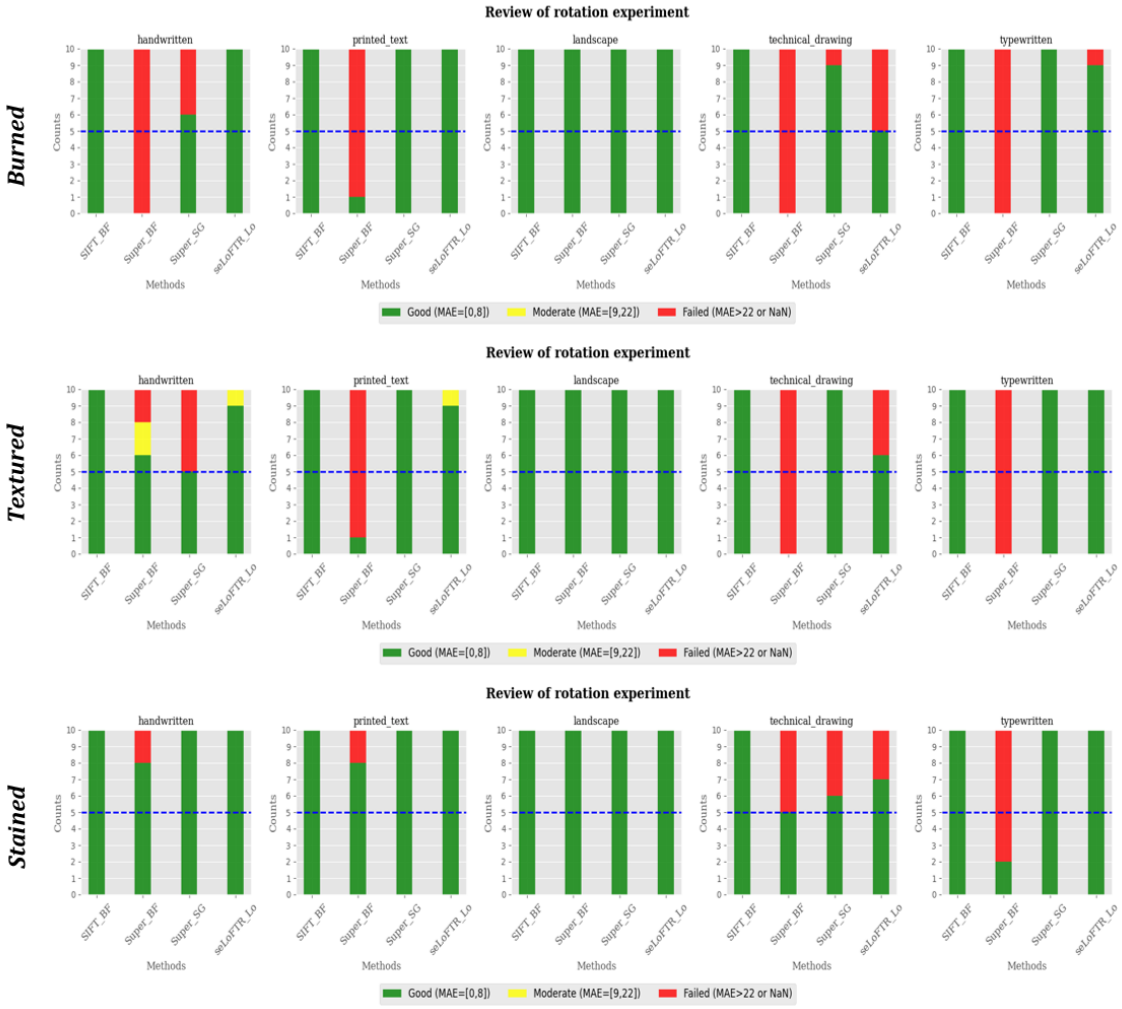


Figure 2: **Results of the Rotation Robustness Experiment - Medium damage level**. These graphs give an overview of the results on Burned, Textured and Stained snippets.

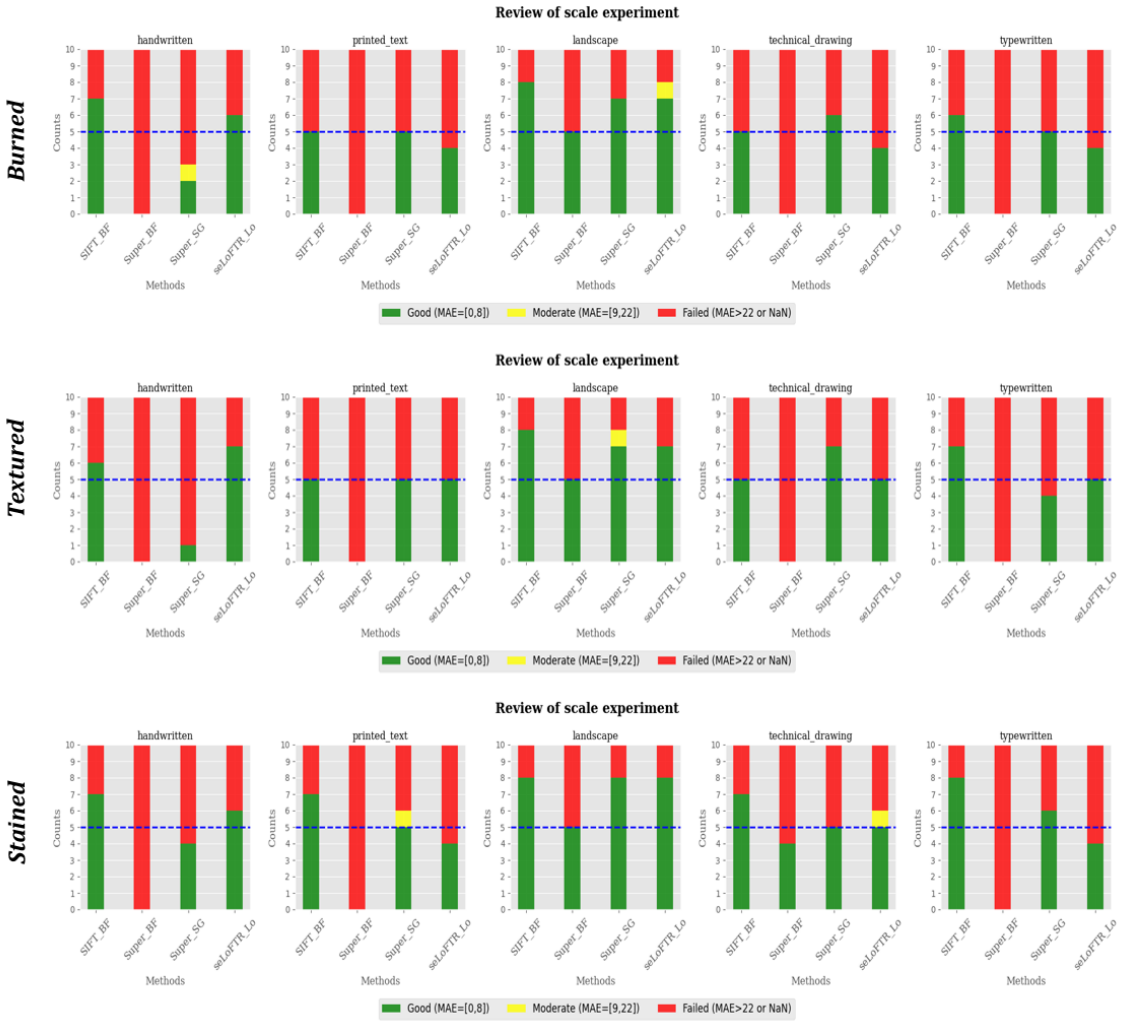


Figure 3: **Results of the Scale Robustness Experiment - Medium damage level**. These graphs give an overview of the results on Burned, Textured and Stained snippets.