Document Image Dewarping and Illumination Correction using Reference Templates

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften
(Dr.-Ing.)

von der KIT-Fakultät für
Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation
von

M.Sc. Felix Jonas Hertlein

Tag der mündlichen Prüfung:	10.04.2025
Referent:	Prof. Dr. York Sure-Vetter
Korreferent:	Prof. Dr. Thomas Setzer

Karlsruhe, 2025

Abstract

In today's fast-paced world, the digitalization of business workflows has become indispensable for organizations to remain competitive and efficient. Documents play a critical role in these workflows, as they contain vital information necessary for decision-making and record-keeping. Despite advancements in digitalization, a significant proportion of documents still exists in physical formats, necessitating digitization for seamless integration into digital workflows. Since manual digitization is labor-intensive and scanner-based digitization inflexible, there is a need for automated document analysis systems that are capable of processing camera-captured document images. Although camera-based digitization offers greater flexibility, it poses significant challenges due to distortions caused by camera angles, document conditions, and varying lighting environments.

In this work, we address the problems of document image dewarping and illumination correction, as they are essential preprocessing steps for document analysis. We aim to enhance document images to achieve a scan-like quality, thereby enhancing downstream tasks such as text detection and document understanding. Although the state-of-the-art methods have made significant progress in this area, further advancements are still needed, especially in real-world scenarios. We work towards improving the existing methods by leveraging additional information about the document structure and visual appearance, which we refer to as reference templates.

The main contributions of this work are as follows:

1. We create the first large-scale, high-resolution dataset for document image dewarping and illumination correction with reference templates, enabling the development of more accurate and robust document image enhancement models.
2. We introduce two novel deep-learning-based systems for geometric dewarping, which integrate reference templates to minimize distortions in warped document images and thereby significantly improve the quality of the dewarped images.
3. We present a new method for illumination correction of document images using reference templates, thus improving the readability and interpretability of the documents.

The contributions are evaluated individually, following predefined requirements and adhering to state-of-the-art evaluation methodologies. The outcome led us to conclude that the information contained in reference templates can be effectively leveraged to improve geometric dewarping and illumination correction. Thereby, we narrow the gap between research and real-world applications, bringing us closer to achieving fully automated document analysis in real-world contexts.

Contents

Part I

Overview

1 Introduction

Since the invention of computers, there has been a rapidly evolving trend to digitalize workflows, especially in the business world [69]. This includes - amongst others - areas such as the retail industry, insurance sector, and healthcare system. While digitalization in these fields has advanced enormously over the last decades, in practice, numerous analogous workflows still rely on paper sheets for information transfer between two parties. Since one party alone cannot easily change the workflow to digitalize it, there is a need to digitize the information received on paper sheets. For instance, this could be a shipping note or invoice from a retail customer.

The availability of digitized workflow information comes with a multiplicity of advantages. The information can be processed fully automatically and linked with other information, thus allowing business decisions to be made quickly and, ultimately, without manual input. These steps provide immense economic value as it reduces the overall cost for a company since human labor is generally more expensive than compute power.

There are three fundamental ways to digitize paper sheets: manual data entry, scanner-based, and camera-based approaches. See Figure 1.1 for an overview of these methods that are explained in the following:

- **Manual data entry.** Digitizing documents by hand entails a human reading the document paper sheet and submitting the displayed information into a computer system. This is a tedious task as there is a large amount of printed documents. For a company, this option is slow and extremely costly since humans are limited in their perception speed, and personnel costs are substantial. Also, the accuracy of human data entry highly depends on the employed individual.
- **Scanner-based approaches.** Scanning documents and extracting the information based on the flatbed image of the document using artificial intelligence (*AI*) is very fast in comparison to manual data entry. This is because only a minor part of the process involves a human being, while most work is done automatically. Since this approach involves less manual labor, it is also more cost-effective at scale than the first option. On the downside, this process requires the availability of the proper hardware, namely a scanner, which might not be available at any time in any location. For instance, in the receiving department of a company, there might not be a scanner available at every point of delivery. The accuracy of this method depends on the ability of the *AI* method to extract the information from the scan. With the recent improvements in deep learning (*DL*) methods, this approach appears viable in many contexts.
- **Camera-based approaches.** This way of digitization uses a camera to take a photo of the printed document sheet, which is subsequently processed by an *AI* tool in order to extract the

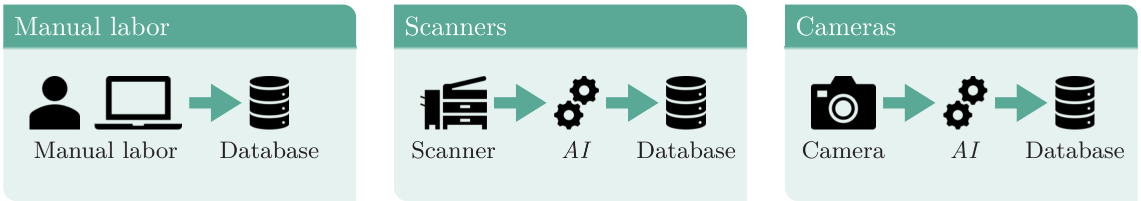


Figure 1.1: Fundamental ways to digitize paper sheets.

information contained in the document. This method is even faster than the scanner-based approaches since handheld cameras, i.e., smartphones, are ubiquitously available, and thus, there is no need to have a scanner available nearby in order to start the digitization process. The disadvantage of this approach is that the task of information extraction on camera images is more challenging as the environment induces geometric and illumination distortions. This affects the accuracy of this approach negatively and thus limits its applicability in the real world.

In this work, we aim to improve camera-based images of documents to mitigate the environmental influences, namely geometric warping and illumination. Given a document image taken by a camera, our goal is to create a new image that looks like one would have used a scanner on the document in order to create this new image. By transforming the camera-based images to scanner-based images, the drop in accuracy for information extraction tasks can be reduced, thus increasing the applicability of camera-based information extraction methods in the real world. This leads to a fast and accurate method for document digitization, which is not bound to a physical location. We introduce the concept of reference templates in this work as an additional source of information for a *DL* model in order to improve the dewarping and illumination correction capabilities of our models.

1.1 Challenges

In this section, we go into detail on the challenges of information extraction (*IE*) from camera-based document images. In contrast to scanned documents, photographed documents exhibit many environmental influences, making information extraction tremendously more difficult. Scanners yield, by their nature, an image of the original document that resembles the paper document closely, as the document needs to be placed flatbed into the device, and the device lights the document uniformly. When digitizing documents using a camera, these constraints do not necessarily apply. The captured document might be in a different pose relative to the camera, there might be deformations on the document paper sheet, and the environment might strongly influence the lighting. We subdivide the challenges into three categories: camera distortions, paper distortions, and lighting influences. It is important to note that all of these effects can occur simultaneously in a single document image. See Figure 1.2 for an overview of the challenges, which are explained in the following:

- **Camera Distortions.** This category contains all challenges with regard to the camera as a capturing device. Since there is no enforcement of a specific pose for the document, the document inside the captured image can be rotated, scaled, translated, and distorted in perspective. In addition, camera lenses can cause non-linear distortions in captured images.

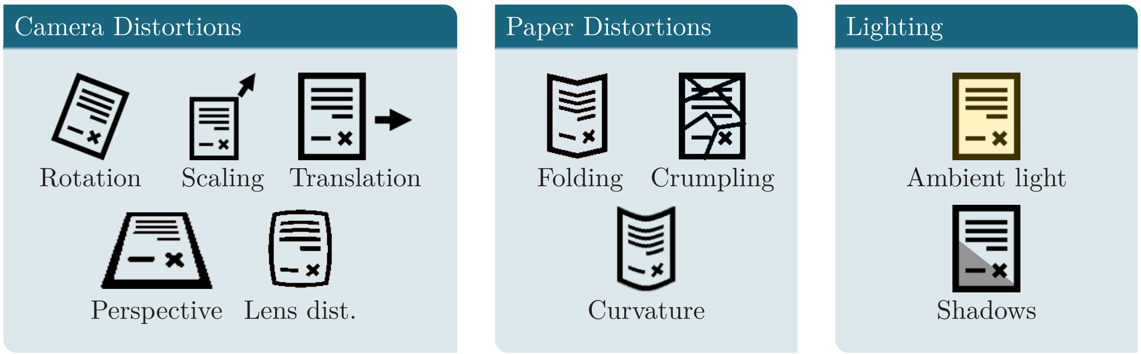


Figure 1.2: Challenges for camera-based information extraction.

This effect leads to curled lines in the captured image, which should be straight according to the real-world object.

- **Paper Distortions.** We denote all deformations affecting the document paper sheet as shape deformations. It contains folding, crumpling, and curvature. The first creates sharp edges on the document while leaving the faces intact but skewed to each other. The second, crumpling, creates many fine edges of irregular character and tiny faces. Lastly, curvature bends the document without creating edges, but the document surface is curled.
- **Lighting.** In addition to the geometric deformations, there are also illumination-changing environmental influences. We summarize these under the category Lighting. It consists of ambient light and shadows. The prior is defined as the illumination effects created by artificial and natural light sources. Here, the effect is typically a change in hue, saturation, and lightness for the entirety of the document in a uniform or smoothly changing manner. The second challenge, shadows, results from objects between the light source and the document, thus creating a shadow on the document, which is captured by the camera.

In addition to the influences above, further influences can affect the document sheet and image-capturing process. These include (partially) destroyed documents, stains of additional substances, e.g., coffee, and blur while image capturing. In this work, we focus our research efforts on mitigating the effects above, as they are widely present in many real-world scenarios.

Extracting details from document images is not a new concept, as many commercial products already do. Therefore, the question arises: Why is the task at hand hard to solve? In this work, we consider heavy distortions applied to the documents in combination with environments in extreme lighting, as well as a multitude of different influences simultaneously. Because of this, a complex geometric reconstruction and illumination correction is necessary in order to create a scan-like image of the document and subsequently extract the information inside the document. Simply extracting the text of a deformed document image performs worse by a large margin, as we show in our results.

1.2 Key Idea & Hypothesis

In this thesis, we aim to improve existing geometric dewarping and illumination correction models to bring these approaches closer to real-world applicability. As described in Section 1.1, this task is challenging due to complex interfering factors while capturing the document. One key observation

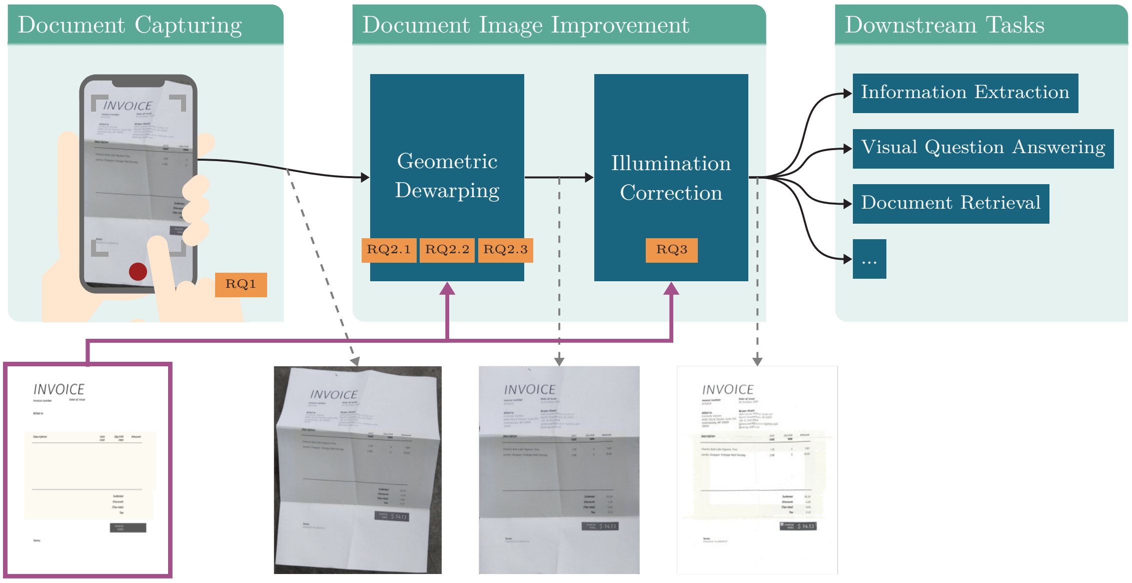


Figure 1.3: Overview of the document image analysis process. Starting with a camera-captured image of a document, geometric dewarping and, subsequently, illumination correction improve the image before applying the downstream tasks. We leverage reference template information in both improvement tasks.

for our hypothesis is that we humans seemingly use our prior knowledge about documents to make sense of the warped and barely readable document in our heads. This appears to be possible since we have seen many documents or semi-structured forms throughout our lives, which we can use to compare the warped document with them. We want to enable our *AI* models also to leverage knowledge about the expected structure and the expected visual queues for the dewarping and illumination correction tasks. The models should be able to leverage this explicitly provided additional information in order to solve their respective task more accurately. As for geometric dewarping, the knowledge of the true location of each visual element (e.g., logo) inside the document eases the task of moving the pixels to the correct location. For illumination correction, the knowledge about the expected true colors helps to remove the illumination artifacts from external light sources.

Our hypothesis is defined as follows:

Hypothesis: Reference Templates

Additional information about the expected document structure and visual appearance can be leveraged to improve the document image enhancement process.

We represent this additional information using RGB images, referred to as **reference templates**, which display the document structure without any instance-specific details. A reference template includes fixed visual elements, such as logos, lines, and static text. For instance, given a document image displaying an invoice, the reference template is an image of the invoice template prior to filling and printing the document. See Figure 1.3 for an example of a reference template and the integration in the enhancement process. For simplicity, we will refer to reference templates simply as templates in this thesis.

1.2.1 Real-world Applicability

In practice, we encounter a multiplicity of scenarios in which the expected template is known a priori due to the context of the usage, such as structured forms in administrative processes, standardized templates in industry-specific documents, or repeated use of consistent layouts in everyday tasks. Fields of application include, but are not limited to, the following: industry, public administration, healthcare, finance, and logistics. Since these interactions are repetitive and, in some cases, even standardized, the templates can be created once during the setup of the system and be reused for all incoming documents. This initial overhead amortizes over time, and the system works without further manual intervention.

Here are two concrete examples of scenarios with a priori known templates:

1. **Company with known suppliers**
   Imagine the situation of a manufacturer in the industry. In practice, the manufacturer has a list of suppliers who create many invoices. Each supplier has its own template, which is usually consistent over time. So, when creating an application to extract the information automatically, the template list must be created by hand only once to process all incoming invoices. The worker in the receiving department can select the correct supplier (and implicitly the correct template) from the available list of suppliers before photographing the invoice. In future work, we can tackle the automatic selection of the best template given a set of templates to automatize the process entirely.

2. **Confirmation of residence**
   In Germany, many bureaucratic processes require transmitting information to the local government agencies using sheets of paper. One example for this case is the confirmation of residence, a document stating a landlord's supplying of the residence to the tenant as specified by the Federal Act on Registration Section 19 [29].

Note that current state-of-the-art approaches for dewarping and information extraction are more flexible as they do not rely on the availability of a reference template. However, they are also less accurate and robust than the proposed approach. By incorporating this a priori knowledge, we can improve the quality and robustness of information extraction models and bring the state-of-the-art closer to the real-world application. Often, these attributes are more important than the model's flexibility since an inaccurate model would lead to a high error rate in the information extraction process, which might not be acceptable for a real-world scenario.

1.3 Research Questions

We divide our research objectives into three categories: (1) data acquisition, (2) geometric dewarping, and (3) illumination correction. The category geometric dewarping is subdivided into three individual questions, whereas the other categories contain a single question each. Figure 1.3 outlines the document image analysis process and locates the research questions within.

The first challenge when working with reference templates for document image enhancement is the availability of suitable datasets for the given task. To the best of our knowledge, prior to this work, no suitable dataset that offers both ground truth annotations and reference templates was available. Therefore, our first research question concerns acquiring suitable data for both tasks, geometric dewarping and illumination correction. RQ1 is given as follows:

RQ1: Data Acquisition

How can we generate a **large-scale, high-resolution dataset** of document images with **ground truth annotations** for geometric dewarping and illumination correction with corresponding **reference templates**?

With the data from RQ1, we want to investigate deep learning architectures and methods to leverage the additional information about the expected document structure and visual cues provided by the reference templates. For this research question, we focus our endeavors on implicit ways to integrate the additional information. In this context, implicit means that the model is provided with additional information as input, but it is expected to figure out how to use the information to improve geometric dewarping. We define the RQ2.1 as follows:

RQ2.1: Implicit Geometric Dewarping

How can we dewarp document images using a **reference template** to improve the quality of the document images?

The second part of RQ2.1 poses the question on the correct metric for measuring the geometric dewarping model performance. We find that all existing metrics show at least one flaw each, making them unsuitable for the evaluation. Visual metrics are based on image comparisons, but they lack sensitivity with regard to text readability. Text-based metrics compare two texts, but they are susceptible to the correct order of detected words, which leads to inconsistent behavior. That is why we introduce a new metric in RQ2.2:

RQ2.2: Dewarping Metric

How can we **evaluate the quality** of the geometric dewarping process with regard to text readability and positional awareness?

In the third part of RQ2, we strive to find new models and approaches for geometric dewarping that outperform the results of RQ2.1. For this approach, we build a multi-stage architecture with interpretable intermediate results at each stage, which allows for more model supervision during the process. That way, we can decompose the complex task of geometric dewarping into a subset of smaller and simpler tasks.

RQ2.3: Explicit Geometric Dewarping

How can we dewarp document images with a reference template by **explicitly leveraging the template information** to improve the quality of the document images?

Our last research question (RQ3) addresses the task of illumination correction. After the geometric correction stage, the resulting images are usually not 100 % correctly dewarped. Given these partially dewarped images, illumination correction aims to remove the illumination artifacts originating from external light sources during image-capturing. Since the input image and the reference template images do not match pixel perfect, the illumination correction becomes an interesting task. We formulate our research question as follows:

RQ3: Illumination Correction

Given the partially dewarped document images, how can we **correct the illumination** to improve the quality of the document images?

1.4 Contributions

Our main contributions comprise two novel datasets for document image improvement, three state-of-the-art deep learning models, and one new evaluation metric. In detail, the contributions are:

RQ1 We contribute two high-quality datasets for geometric dewarping and illumination correction, called Inv3D and Inv3DReal. The prior is a fully synthetic, large-scale dataset containing warped invoices, reference templates, and full ground-truth annotations. These entail, amongst others, many per-pixel annotations in the warped domain and several annotations in the flat domain. The dataset consists of 25,000 samples. The second dataset (Inv3DReal) consists of semi-realistic data for evaluation. Similar to Inv3D, the invoices are generated synthetically, but the captured document images show real printed invoice sheets. It consists of 360 samples in total.

RQ2.1 We propose a novel deep-learning model called GeoTrTemplate. It extends the previous state-of-the-art model GeoTr [21] by incorporating reference templates implicitly. We show that our new model GeoTrTemplate outperforms GeoTr on all metrics, most notably in local distortion (*LD*) by 26.1 %.

RQ2.2 In order to mitigate the flaws in existing metrics for the evaluation of geometric dewarping models, we propose a new metric called matched normalized Character Error Rate (*mnCER*). The new metric is text-aware, thus, non-readable documents due to fine-grained dewarping errors get a low score. Also, it removes the need to arrange the detected words in the document linearly, and, thus, becomes insensitive to minor positional changes.

RQ2.3 We develop a new multi-stage deep learning model called DocMatcher, which incorporates the information from the reference template explicitly in order to geometrically dewarp camera-based document images. The model focuses on structural and textural lines in the warped and the flat domain and associates both. Based on these matches, a pixel-wise flow mapping is computed to dewarp the image. Our approach improves upon the state-of-the-art methods in all metrics, most notably in *LD* by 32.6 % and in *mnCER* by 40.2 %.

RQ3 To correct the illumination artifacts from external light sources, we propose a new model called IllTrTemplate. It leverages the reference templates to improve the color correction compared to methods without reference templates. Our model demonstrates a 15.0 % relative improvement in Learned Perceptual Image Patch Similarity (*LPIPS*) and 6.3 % in Character Error Rate (*CER*).

1.5 Outline

This thesis is divided into six parts: (I) overview, (II) preliminaries, (III) data generation, (IV) geometric dewarping, (V) illumination correction, and (VI) synthesis. The details for each part are presented below:

Part II presents the preliminaries in which we explain relevant foundations for understanding this thesis. In addition, we discuss existing approaches and datasets in the chapter-related work.

Part III introduces the data generation pipeline for our datasets Inv3D and Inv3DReal. We analyze the data requirements for document image enhancement and derive our dataset design decisions. Furthermore, we introduce the reference templates formally. This part of the thesis builds upon the following publication:

- Felix Hertlein et al. "Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping". In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: `10.1007/S10032-023-00434-X`

Part IV addresses the question of geometric dewarping. First, we formally define the problem at hand before presenting both our implicit and our explicit approach to geometric dewarping with the use of reference template images. Besides, we introduce our new metric *mnCER* as a robust way of evaluating geometric dewarping models. This part of the thesis builds upon the following publications:

- Felix Hertlein et al. "Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping". In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: `10.1007/S10032-023-00434-X`
- Felix Hertlein et al. "DocMatcher: Document Image Dewarping via Structural and Textual Line Matching". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2025

Part V provides our approach on illumination correction. Firstly, we formalize the problem before presenting our solution to template-based illumination correction, assuming that the prior geometric dewarping stage fell short of perfection. This part of the thesis builds upon the following publication:

- Felix Hertlein and Alexander Naumann. "Template-guided Illumination Correction for Document Images with Imperfect Geometric Reconstruction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.* IEEE, 2023, pp. 904–913. DOI: `10.1109/ICCVW60793.2023.00097`

Part VI presents our synthesis. We conclude the thesis and give an outlook on future research directions.

Part II

Preliminaries

2

Foundations

This chapter outlines the foundational concepts of this work, categorized into three key areas: (1) coordinate transformations, (2) deep learning architectures, and (3) evaluation metrics. The first section provides details on the transformations between coordinate systems relevant to this work, while the second section explores various architectures that form the basis for our proposed models. In the final section, we introduce the established evaluation metrics in the field of geometric dewarping and illumination correction, which we use to assess the performance of our models.

2.1 Coordinate Transformations

A coordinate transformation is a function that maps points from a source coordinate system $\mathcal{S}$ to a target coordinate system $\mathcal{T}$. This section describes the coordinate transformations used in this work, namely Frenet coordinates, sinusoidal positional encoding, homography, and Delaunay triangulation. These transformations are essential to our work for a variety of reasons:

1. They allow us to represent the input data in a more machine-interpretable format by transforming the data into a rectangular shape.
2. They improve the generalization capabilities of deep learning models by the smooth and predictable nature of the transformations.
3. They allow us to model partial dewarping transformations.
4. They provide a way to interpolate dewarping transformations defined by sparse control points.
5. They enable fine-grained control over the transformations.

2.1.1 Frenet Coordinates

Frenet coordinates are derived from the Frenet frame, a mathematical concept in differential geometry used to describe the movement of a particle along a curve in three-dimensional space. Werling et al. [91] apply the two-dimensional case of Frenet frames to the optimal trajectory planning problem for autonomous vehicles. In the following, we describe the 2D Frenet coordinates and their relation to the Cartesian coordinate system. Contrary to the work of Werling et al. [91], we do not focus on the dynamic properties of the Frenet frame since we do not require them in this thesis.

Given a point $P = (x, y)$ in the Cartesian coordinate system and a curve C, we can describe its position using Frenet coordinates (s, d), where s is the distance along the C (longitudinal

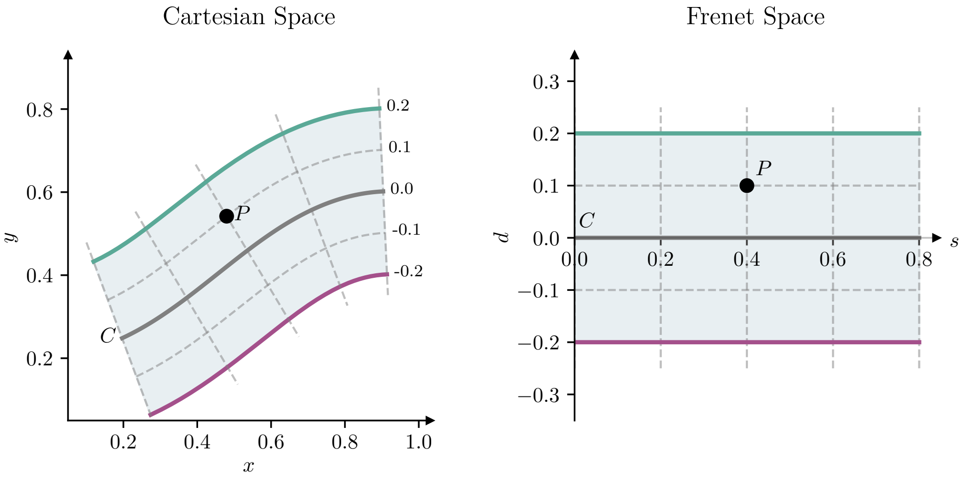


Figure 2.1: Frenet coordinate transformation.

displacement), and d is the distance perpendicular to the curve (lateral displacement). Formally, we can express Frenet coordinates as follows:

$$\begin{aligned} \xi_C : \mathbb{R}^2 &\to \mathbb{R}^2 \\ (x, y) &\mapsto (s, d) \end{aligned}$$

See Figure 2.1 for an illustration of the Frenet coordinates. An intuitive understanding of the Frenet coordinates can be obtained by considering a car driving along a road. In this context, C represents the road's centerline, s is the distance traveled along the road, and d is the distance from the center of the road.

The transformation from Cartesian coordinates to Frenet coordinates enables us to map a tube-like region around the curve C to a rectangular region in the Frenet coordinate system, as shown in Figure 2.1. Thus, the Frenet coordinates provide a more compact and uniform representation of the curve C and its proximity, independently of the course of the curve. By rescaling the Frenet coordinates to the range $[0, 1]^2$, we can normalize the representation of the curve and its surroundings. In this thesis, we use the normalized Frenet coordinates to represent the visual appearances of the lines inside the warped document and their surrounding context in a fixed-size form suitable for deep learning models. Note that the conversion from Cartesian to Frenet coordinates is a non-linear transformation that distorts the space, i.e., it stretches the space in some regions and compresses it in others.

2.1.2 Sinusoidal Positional Encoding

This section is primarily based on the publication *Attention Is All You Need* by Vaswani et al. [84].

Sinusoidal positional encoding is a method to represent coordinates using a combination of several sine and cosine functions with different frequencies. It is used in the Transformer architecture [84] to encode the position of the tokens within the model. Vaswani et al. [84] proposed the one-dimensional sinusoidal positional encoding, which later got extended to two dimensions by Wang and Liu [89].

Given a sinusoidal encoding feature depth of D, the two-dimensional encoding PE is defined as follows:

$$\begin{aligned}
\mathrm{PE}_D(x, y, 2i) &= \sin\left(\frac{x}{10000^{\frac{4i}{D}}}\right), \\
\mathrm{PE}_D(x, y, 2i+1) &= \cos\left(\frac{x}{10000^{\frac{4i}{D}}}\right), \\
\mathrm{PE}_D(x, y, 2j+\frac{D}{2}) &= \sin\left(\frac{y}{10000^{\frac{4j}{D}}}\right), \\
\mathrm{PE}_D(x, y, 2j+1+\frac{D}{2}) &= \cos\left(\frac{y}{10000^{\frac{4j}{D}}}\right),
\end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ denote the 2D position to encode, and the third parameter denotes the channel position within the D-dimensional encoding. The numbers i and j range from 0 to $D/4$, not including $D/4$ itself, which results in D channels in total. Note that D is required to be a multiple of 4.

Another way of expressing this is by using the following notation:

$$\mathrm{PE}_D(x, y) = \begin{bmatrix} \sin(\omega_0 x) \\ \cos(\omega_0 x) \\ \vdots \\ \sin(\omega_m x) \\ \cos(\omega_m x) \\ \\ \sin(\omega_0 y) \\ \cos(\omega_0 y) \\ \vdots \\ \sin(\omega_m y) \\ \cos(\omega_m y) \end{bmatrix} \tag{2.2}$$

$$\text{where} \quad \omega_k = \frac{1}{10000^{\frac{4k}{D}}},\ k = 0, 1, \dots, m := \frac{D}{4} - 1.$$

Within the encoded vector $\mathrm{PE}(x, y)$, the first $\frac{D}{2}$ dimensions represent the x-position, while the second $\frac{D}{2}$ dimensions represent the y-position. Figure 2.2 illustrates the sinusoidal positional encoding for a depth of $D = 8$.

Sinusoidal positional embeddings have a set of desirable properties, which support deep learning architectures in grasping the concept of positional information[1][2][3]:

[1] https://medium.com/@pranay.janupalli/understanding-sinusoidal-positional-encoding-in-transformers-26c4c161b7cc, Accessed 29th of November 2024

[2] https://www.scaler.com/topics/nlp/positional-encoding/, Accessed 29th of November 2024

[3] https://kazemnejad.com/blog/transformer_architecture_positional_encoding/, Accessed 29th of November 2024

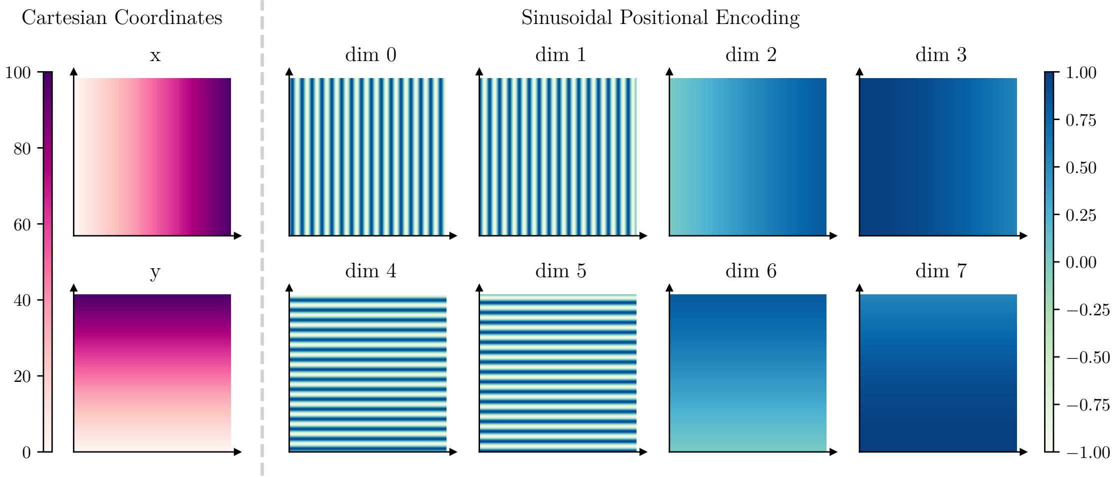


Figure 2.2: Visualization of sinusoidal positional encoding. The left side shows 2D Cartesian coordinates. The right side visualizes the corresponding sinusoidal positional embeddings for $D = 8$.

1. **Normalized Range.** Since the embeddings have a value range of $[-1, 1]$, the model is likely to encounter the full value range during the training process and, thus, is likely to generalize to unseen positions as well. Additionally, normalized data values contribute to training stability by helping prevent potential exploding gradients.
2. **Smoothness.** The fact that sine and cosine functions are differentiable, smooth, and periodic is beneficial for generalization to unseen positions.
3. **Relative Distance.** Sinusoidal positional embeddings enable a model to learn relative distances easily since for a fixed offset k, there is a linear function f such that $f(PE_D(x)) = PE_D(x + k)$.
4. **Global Coordinates**. The embeddings capture the nested coordinates on a global scale, enabling the model to interpret and reason about the overall positions of each embedded coordinate.
5. **Unique Position Encoding.** Lastly, for any given position in the Euclidean space, there should be a unique representation in the embedded space to avoid disambiguation. This property holds until the sine wave with the largest frequency in embedding repeats itself. Due to the large constant of 10,000 in the definition of the embedding, this does generally not occur in praxis.

In this work, we use sinusoidal positional encodings to represent the position of each pixel in a given image in a machine-learning-friendly manner. Subsequently, we transform a region of the image and its associated coordinates using the Frenet coordinate transformation while keeping track of the original pixel coordinates as a sinus embedding. This approach encodes the original shape and global position of the region of interest prior to the Frenet transformation.

2.1.3 Homography

This section is based on the book *Multiple View Geometry in Computer Vision* by Harltey and Zisserman [27].

A two-dimensional homography, also known as a perspective transformation, is a special case of a projective transformation, which maps a two-dimensional plane to another two-dimensional plane:

$$\begin{aligned} \zeta \colon \mathbb{R}^2 &\to \mathbb{R}^2 \\ (x, y) &\mapsto (\tilde{x}, \tilde{y}) \end{aligned}$$

A homography is defined by a 3×3 matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^\top & 1 \end{bmatrix} \tag{2.3}$$

where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

The sub-matrix **A** is called affine matrix, as it encodes the geometric transformations of an affine transformation except the translation. **t** describes the translation necessary to transform one point from the source to the target plane. The vector **v** determines the perspective distortion, which allows parallel lines to meet at a vanishing point. Since we are using homogenous coordinates to define the coordinate transformation, only relative ratios between the matrix elements matter, meaning that the overall scale of the matrix **H** does not affect the transformation.

For the conversion of a point $p = (x, y)$ from one plane to the other, we need to represent the point in homogenous coordinates $p_h = (x, y, 1)$ and multiply with $\mathbf{H}$:

$$\begin{bmatrix} \tilde{x}_h \\ \tilde{y}_h \\ \tilde{w}_h \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ v_1 & v_2 & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The transformed coordinates in Cartesian form can be obtained as:

$$\tilde{x} = \frac{\tilde{x}_h}{\tilde{w}_h}, \quad \tilde{y} = \frac{\tilde{y}_h}{\tilde{w}_h}.$$

Figure 2.3 illustrates the homography coordinate transformation. We can see that the homography transformation can be used to map a plane to another plane while preserving the straight lines.

In this work, we estimate the homography transformation to minimize the overall warping of the document in the input image. Since the document in the input image is approximately a plane, we use the estimated homography to map the document to a frontal view without perspective distortion and skew, as well as non-optimal scaling and translation. Note that this mapping is not capable of dewarping the fine-grained displacement since it only has eight degrees of freedom.

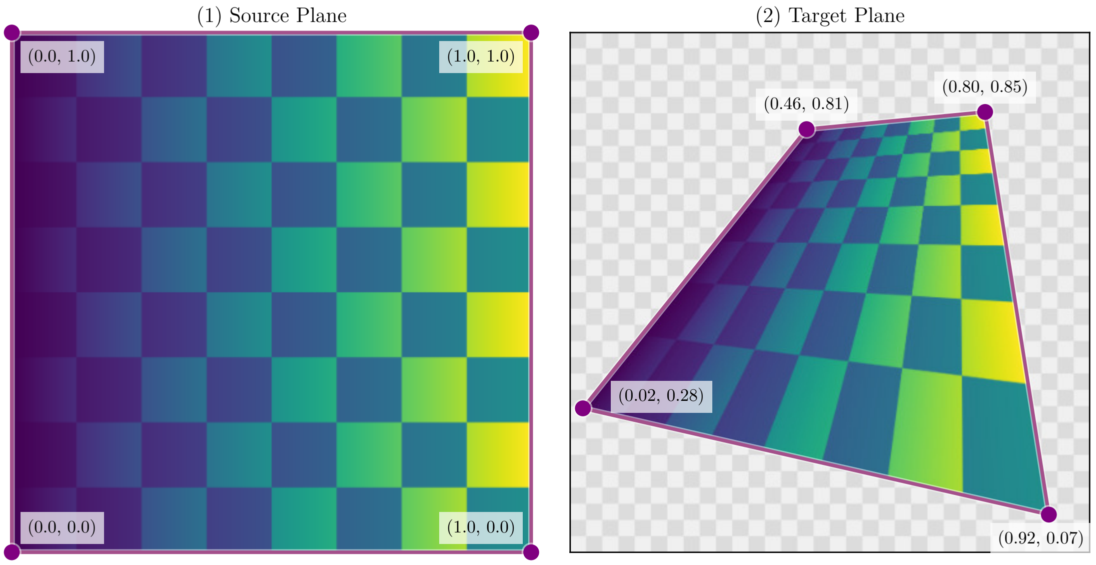


Figure 2.3: Homography coordinate transformation.

2.1.4 Delaunay Triangulation

This section is based on the book *Delaunay mesh generation* by Cheng et al. [11].

Delaunay Triangulation, introduced by Boris Nikolaevich Delaunay in 1934, is a method for triangular mesh generation given a finite set of points $\mathcal{P}_n \subset \mathbb{R}^n$. Even though the triangulation method works on n-dimensional data points, we focus on the 2-dimensional case for our tasks.

In this work, we use the triangular mesh generated using the Delaunay Triangulation algorithm to interpolate sparsely given coordinate mappings in order to allow for a coordinate transformation from a rectangular source domain $[0, 1]^2 \subset \mathbb{R}^2$ to another rectangular target domain $[0, 1]^2 \subset \mathbb{R}^2$. See Figure 2.4 (1) for an example set of points. Note that our example point set contains, among others, the points $(0, 0), (0, 1), (1, 0), (1, 1)$. These points are required for image-to-image mapping since interpolation via meshing can only interpolate values inside the convex hull of our point set.

Given the point set $\mathcal{P}_2 \subset \mathbb{R}^2$, there are many possibilities to generate a triangular mesh. One desirable property of triangular meshes is maximizing the smallest angle between the sides of any triangle. This property leads to a mesh, where the triangles are as little flat as possible. According to this property, an optimal mesh contains only triangles with an angle of 60°. This is desirable for interpolation since it leads to smaller distances between the points we want to interpolate and, thus, less interpolation error. Delaunay triangulation satisfies this property, making it a suitable interpolation method. Figure 2.4 (2) shows the Delaunay Triangulation of our example point set.

For transforming coordinates from a rectangular source domain to another rectangular target domain via sparse control points, we can now associate a target vector $v_i \in [0, 1]^2$ to each point $p_i \in \mathcal{P}_2$. Figure 2.4 (3) shows the x-component of v_i for each point in our example point set. Given these target vectors and the triangulated mesh, we can interpolate the target vectors linearly inside each triangle to generate a dense mapping, one for the x- and one for the y-component of the mapping. Figure 2.4 (4) shows the dense mapping for the x-component of this example coordinate transformation.

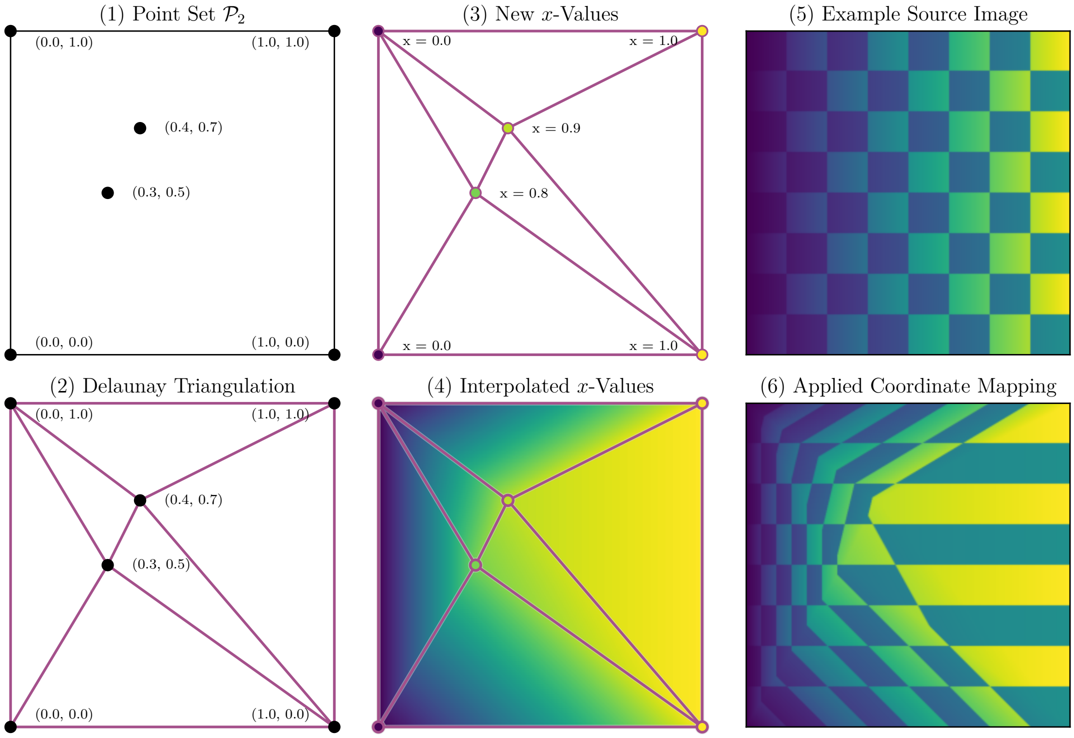


Figure 2.4: Visualization of a coordinate transformation using Delaunay triangulation. For simplicity, we only define and apply the transformation in x-direction.

For clarity, we applied our example coordinate transformation to an example input image (Figure 2.4 (5)), which results in the transformed image Figure 2.4 (6).

2.1.5 Dense coordinate mapping

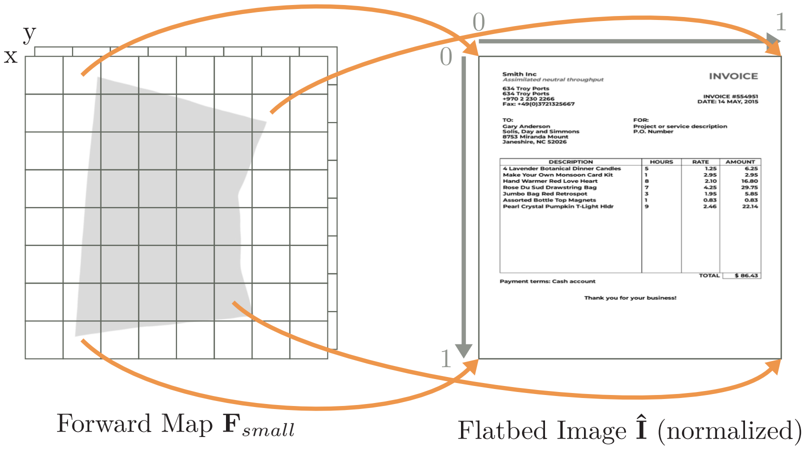
Contrary to sparse coordinate mapping, 2D dense coordinate mapping is defined for a regular grid of points, e.g., the pixels of an image. For each point in the grid, the dense coordinate mapping provides an individually definable target coordinate in the target domain. Therefore, the dense coordinate mapping can be represented as a 2D tensor $\mathbf{M} \in [0,1]^{w \times h \times 2}$, where w and h are the width and height of the grid, respectively.

The mapping from source to target coordinate can be expressed as a function η_M:

$$\begin{aligned} \eta_{\mathbf{M}} \colon [0,1]^2 &\to [0,1]^2 \\ (x,y) &\mapsto \begin{cases} \mathbf{M}_{x*w,y*h} & \text{if } (x*w, y*h) \text{ is on the grid,} \\ \text{Interp}(\mathbf{M}, x*w, y*h) & \text{otherwise.} \end{cases} \end{aligned}$$

Any point (x, y) on the grid is mapped to its target coordinate $\mathbf{M}_{x*w,y*h}$. For points not on the grid, we use bilinear interpolation to determine the target coordinate.

In this work, we use dense coordinate mapping to represent the geometric transformation from the warped document image to the flatbed domain. We refer to the dense coordinate mapping from


y
x
0
1
0
1
Forward Map $\mathbf{F}_{small}$
Flatbed Image $\hat{\mathbf{I}}$ (normalized)

Smith Inc
Assimilated neutral throughput

634 Troy Ports
634 Troy Ports
+970 2 230 2266
Fax: +49(0)3721325667

INVOICE

INVOICE #554951
DATE: 14 MAY, 2015

TO:
Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
		TOTAL	$ 86.43

Payment terms: Cash account

Thank you for your business!

Figure 2.5: Example of a small forward map $\mathbf{F}_{small} \in [0,1]^{8\times 8\times 2}$. Each vector in $\mathbf{F}_{small}$ maps from the warped domain to the flatbed domain. For visual clarity, we visualized only four vectors. Note that the vectors outside the warped image are undefined.

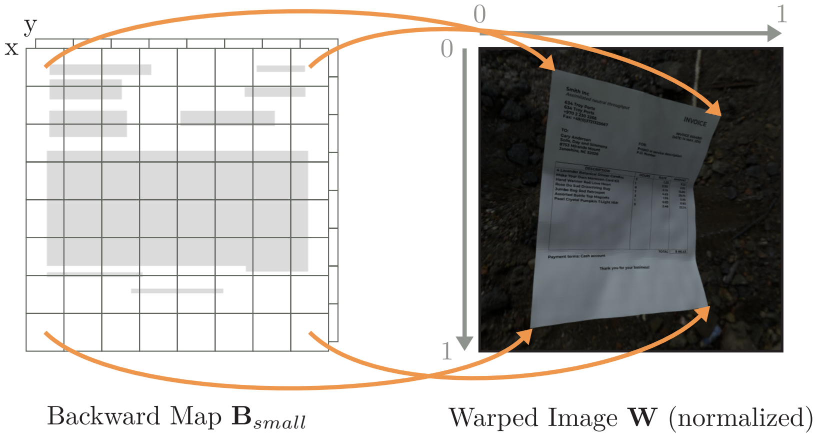

y
x
0
1
0
1
Backward Map $\mathbf{B}_{small}$
Warped Image $\mathbf{W}$ (normalized)

Figure 2.6: Example of a small backward map $\mathbf{B}_{small} \in [0,1]^{8\times 8\times 2}$. Each vector in $\mathbf{B}_{small}$ maps from the flatbed domain to the warped domain. For visual clarity, we visualized only four vectors.

the warped domain to the flatbed domain as the forward map $\hat{\mathbf{F}} \in [0,1]^{w_f \times h_f \times 2}$ where w_f and h_f define the resolution of the forward mapping. See Figure 2.5 for an example of a forward map.

In order to apply the geometric dewarping to an image, we need the reverse projection from the flatbed domain to the warped domain. This projection, called backward map, is defined as $\hat{\mathbf{B}} \in [0,1]^{w_b \times h_b \times 2}$, where w_b and h_b define the resolution of the backward mapping. Each cell in the backward map contains a 2D vector pointing to the position in the warped image where the flatbed pixel was moved during warping. Figure 2.6 shows an example of a backward mapping on the left side. For comprehensibility, only the outermost backward vectors are visualized.

2.2 Deep Learning Architectures

This work builds upon several deep learning architectures, namely the Vision Transformer [19] (*ViT*), Segment Anything [36] (*SAM*), and LightGlue [49]. This section provides an overview of these architectures and their components.

2.2.1 Transformer

This section is primarily based on the publication *Attention Is All You Need* by Vaswani et al. [84].

A transformer is a network architecture centered around the attention mechanism. It was introduced by Vaswani et al. [84] and was initially intended for natural language processing (*NLP*) tasks such as machine translation. Contrary to previous architectures in *NLP*, the transformer architecture does not rely on recurrent or convolutional layers but instead uses the attention mechanism to model dependencies between input and output tokens. The attention mechanism allows the model to selectively focus on specific information, similar to the human cognitive attention. Through the attention mechanism, the model can process important information more efficiently, as it functions as a resource allocation mechanism. Furthermore, it can be used for explainability, as it allows visualizing the importance of each token in the input sequence for the output sequence [68].

The concrete attention mechanism, as used in [84], combines three feature vectors, each intended to represent different types of information: queries, keys, and values. Queries are feature vectors that describe what the model is looking for, i.e., the information the model is interested in. The keys are feature vectors that relate to the kind of information that is stored in the values or when it is relevant. Finally, the values are feature vectors that store the information that is relevant to the queries.[4] Formally, the scaled-dot product attention mechanism can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.4}$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. Note that the attention mechanism can be extended to multi-head attention, where the input is projected linearly into multiple subspaces, and the attention is computed in each subspace before being concatenated and projected back into the original space. This allows the model to capture different aspects of the input and to learn more complex patterns.

The transformer architecture consists of an encoder-decoder structure, where the encoder processes the input sequence, and the decoder generates the output sequence. Figure 2.7 shows the architecture of the transformer. Given an input sequence $(x_1, \ldots, x_n)$, the encoder generates a sequence of representations $\mathbf{z} = (z_1, \ldots, z_n)$. Thereafter, the decoder generates the output sequence $(y_1, \ldots, y_m)$ based on the output of the encoder $\mathbf{z}$. In the following, we describe the architecture in more detail. Note that this is not a comprehensive description of the transformer architecture.

The encoder contains self-attention layers, i.e., the queries, keys, and values are derived from the input sequence using linear transformations. This allows the model to attend between every pair of tokens in the input sequence, thus capturing their relationships.

[4] `https://notesonai.com/attention+mechanism`, Accessed 07th of December 2024

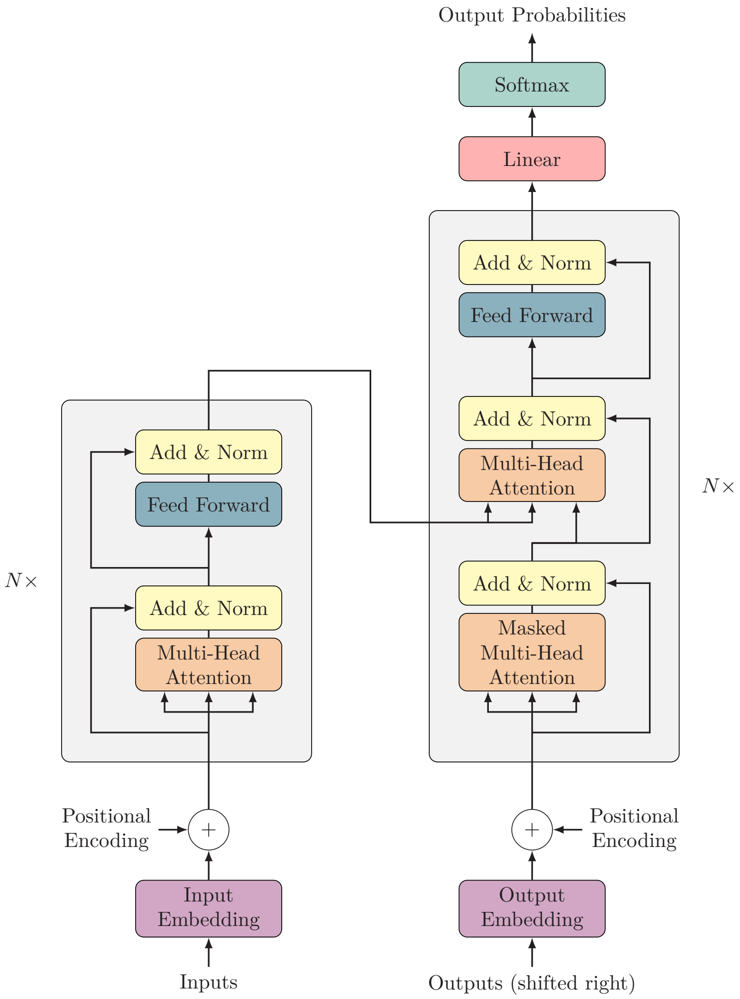


Figure 2.7: Transformer architecture. This figure is derived from Vaswani et al. [84], without significant modifications..

The decoder differs from the encoder in two aspects: (1) it uses masked self-attention and (2) an additional attention mechanism to attend to the encoder's output. First, masked self-attention is used to prevent the model from attending to future tokens, i.e., the model can only attend to already generated tokens. During inference, the decoder generates the output sequence iteratively, starting with a special start token. However, during training, the full target sequence is available, and therefore, a masking mechanism is necessary to prevent the model from attending to future tokens. Secondly, the decoder uses an additional attention mechanism to attend to the encoder's output. The queries are derived from the previous decoder layer, while the keys and values are derived from the encoder's output. This allows the decoder to attend to the full input sequence and to generate the output sequence based on the already generated tokens.

The positional encodings are added to the input embeddings to provide the model with information about the position of the tokens. Since the transformer architecture has no recurrence or convolution, the model does not have any information about the position of the tokens in a given sequence. Vaswani et al. [84] employ one-dimensional sinusoidal positional encodings as described in Section 2.1.2.

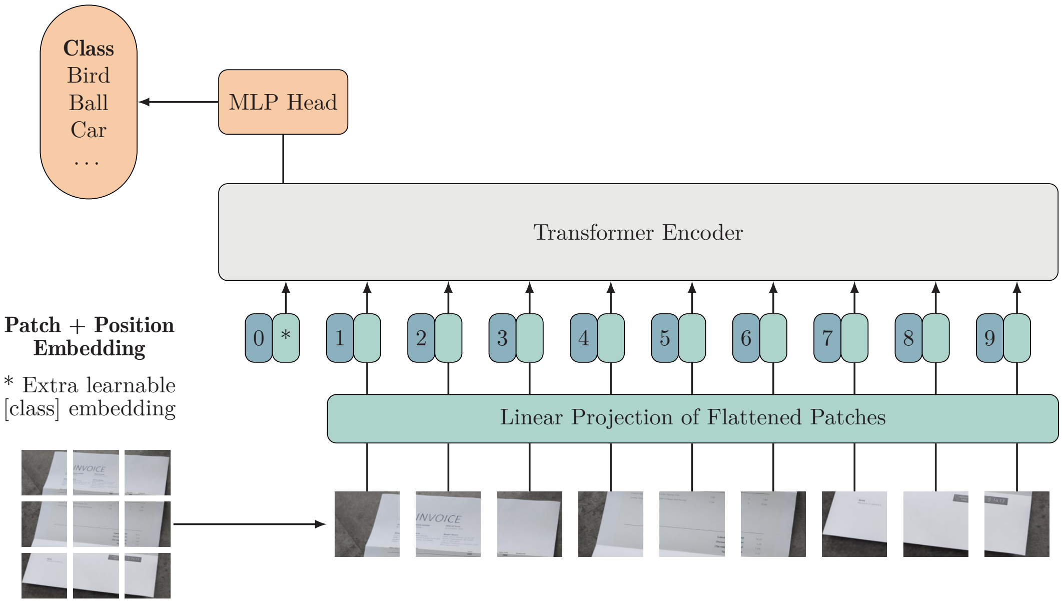

Class
Bird
Ball
Car
. . .
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
INVOICE

Figure 2.8: Vision Transformer architecture. This figure is derived from Dosovitskiy et al. [19], without significant modifications.

In this work, we use an extension of the transformer architecture, the Vision Transformer (*ViT*), as a base model for geometric dewarping and illumination correction. In the following section, we describe the Vision Transformer in more detail.

2.2.2 Vision Transformer

This section is primarily based on the publications *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* by Dosovitskiy et al. [19].

The Vision Transformer (*ViT*) is a transformer-based architecture that was introduced by Dosovitskiy et al. [19] for image classification tasks. It was originally designed as an alternative to the classic convolution-based ResNet [28] architectures and as a direct application of transformers to images with the fewest possible modifications.

Figure 2.8 shows the architecture of the *ViT*. Given an input image $I \in \mathbb{R}^{h \times w \times c}$, the image is split into a sequence of flattened 2D patches $x_i \in \mathbb{R}^{p^2 \cdot c}$, where p is the patch size, and (h, w) are the height and width of the image, and c is the number of channels. The total number of patches is $n = \frac{hw}{p^2}$. Given the patches, a linear layer maps the patches to a D-dimensional space, where D is the dimensionality used by the transformer. Since the transformer requires positional information, learnable 1D positional embeddings are added to the patches. The authors did experiment with 2D positional embeddings but did not observe a significant difference between 1D and 2D embeddings. Note that the positional embeddings are learned during training instead of the sinusoidal positional encodings used in the original transformer architecture. Additionally, a learnable class token, denoted as $[class]$, is appended to the sequence. This token is used for the classification task, and its final state after passing through the transformer encoder is processed by a small MLP to make the final classification. The class token is designed to gather all the relevant information for classification from the whole image.

The transformer encoder itself consists of a stack of L layers. These layers are identical to the original transformer layers, with one exception: the layer normalization is applied before the self-attention and before the MLP. This modification is called pre-normalization and is beneficial because it reduces the risk of vanishing or exploding gradients [87].

When comparing *ViT*s to *CNN*s, Dosovitskiy et al. [19] found that *ViT*s lack some inductive biases of *CNN*s, such as translation equivariance, locality, and the two-dimensional neighborhood structure. Translation equivariance means the model should produce the same but translated output when the input is translated. Locality and the two-dimensional neighborhood structure refer to the relation of pixels in a 2D image. *CNN*s inherently provide these biases through their architecture, while *ViT*s must learn them. As a result, *ViT*s need larger datasets to outperform *CNN*s in performance. However, when trained on sufficiently large datasets, *ViT*s are capable of surpassing *CNN*s.

*ViT*s are not limited to image classification tasks but can be used for various tasks, such as object detection, semantic segmentation, and image restoration [2, 33]. Specifically, the latter is interesting for this work, as we use the *ViT* architecture for geometric dewarping and illumination correction of document images.

2.2.3 Segment Anything

This section is based on the publication *Segment Anything* by Kirillov et al. [36].

Segment Anything (*SAM*) [36] is a deep learning-based instance segmentation model, i.e., a model that detects and isolates individual objects in an image. The intention behind *SAM* is to provide a flexible yet powerful foundation model for image segmentation, which can be used as a building block for more complex computer vision tasks, for example, semantic segmentation. In order to achieve this, Kirillov et al. [36] define a new segmentation task, that is, segmenting individual objects in an image based on prompts. A prompt is a user-provided input, such as a point or bounding box, specifying which object or region the model should segment in the image. *SAM* can handle four types of prompts: points, bounding boxes, coarse masks, and plain text. Points indicate whether a specific region is included in the valid mask or excluded. A bounding box or a coarse mask prompt specifies the region of interest. Lastly, a plain text prompt can guide the model in focusing on a specific object or region in the image based on a textual description. *SAM* consists of three components: (1) image encoder, (2) prompt encoder, and (3) mask decoder. For a visual illustration of the different kinds of prompts and the architecture of *SAM*, see Figure 2.9.

The image encoder is a large vision transformer pre-trained as a masked autoencoder, with slight modifications to the original transformer architecture to process high-resolution images [47]. Since the image encoder is computationally expensive, the authors differentiate between the image encoder and the prompt encoder to reuse the image encodings for different prompts. That reduces the computational cost of the model for multiple prompts.

The prompt encoder is used to encode the prompts into a fixed-size representation. Points and bounding boxes are encoded using two-dimensional Fourier feature-based positional encodings [83] and a learnable vector per prompt type. Plain text prompts are processed and encoded using the text encoder from CLIP [72], a powerful multimodal model designed to link text and images in a shared embedding space. For mask prompts, a convolutional network is used to encode the masks, which are then added directly to the image features.

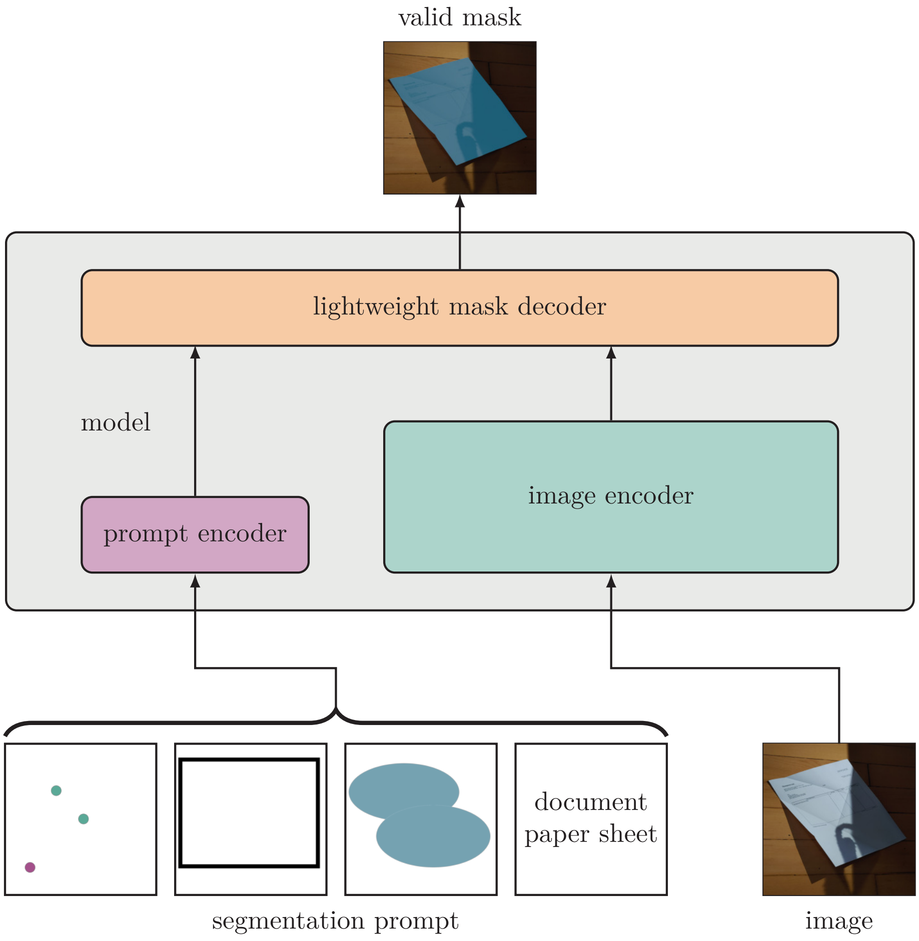


Figure 2.9: Segment Anything architecture. This figure is based on Kirillov et al. [36]

Finally, the mask decoder is used to predict the masks based on the image and prompt encodings. It consists of a variant of a transformer decoder block [84] and a mask prediction head. The mask decoder predicts multiple low-resolution mask predictions and corresponding confidence scores. The decision to predict multiple masks instead of a single mask allows the model to resolve ambiguity in the segmentation task. Given a region in an image, the intended object can be covered by multiple masks, which can lead to ambiguity. During training, the model is trained to predict at least one mask that covers the object by backpropagating only the minimum mask loss.

In this work, we use the *SAM* architecture for the instance segmentation of document images. We use the detected document regions to remove the background and reduce the overall warping of the document in the input image. This simplifies the geometric dewarping task for the subsequent models.

2.2.4 LightGlue

This section is based on the publication *LightGlue* by Lindenberger et al. [49].

LightGlue [49] is a deep-learning model designed to match local features across images. These features are typically sparse interest points with high-dimensional descriptors, which encode the local visual appearance of the region around the point. The model can find correspondences between the images by matching these features across images.

Given two images, A and B, along with two sets of sparse features, $F_A = \{\mathbf{d}_i\}_{i=1}^{M}$ and $F_B = \{\mathbf{d}_i\}_{i=1}^{N}$, where each $\mathbf{d}_i \in \mathbb{R}^d$ is a d-dimensional real vector derived from a 2D point $\mathbf{p}_i = (x_i, y_i)$, the goal is to establish correspondences between the features in F_A and F_B. The correspondences are defined as a set of pairs of indices $\mathcal{C} = \{(i, j)\}$, where i and j are indices into the feature sets F_A and F_B, respectively.

The architecture of LightGlue consists of L stacked identical DL layers that process two sets of features. In order to create a fast and memory-efficient model, the authors designed the model to preemptively stop after a few layers instead of processing all layers. Additionally, the model can discard unmatchable points early, which allows the model to focus on the most informative points. Within each layer, the model takes the input features F_A^k and F_B^k, as well as the points P_A and P_B, and computes a series of self-attention and cross-attention between the two sets of features. Given the updated representations of the points, the authors calculate two different scores using linear layers: assignment scores and matchability scores. The prior encodes the affinity for all pairs of features to form a correspondence, while the latter encodes the likelihood of a feature having a correspondence at all. Based on these scores, the model calculates a soft partial assignment matrix $\mathbf{P} \in [0,1]^{M \times N}$, where P_{ij} denotes the likelihood of the i-th feature in F_A being matched to the j-th feature in F_B. The final assignments are determined by selecting (i, j) pairs, where $\mathbf{P}_{ij} > \tau$ for a threshold τ, and there is no larger value in the same row or column.

In order to make the model fast and memory-efficient, it predicts a confidence score c_i for each feature at the end of each layer. A point is considered confidently matched if the confidence score for that point is above a minimum confidence threshold. If the ratio of confidently matched points is above a minimum ratio threshold α, the model stops processing further layers. Additionally, the model reduces the number of points to process in the next layer by discarding points that are neither confidently matched nor unmatchable.

In this work, we use the LightGlue architecture to match features from the warped document images to their reference templates. Differently from the original approach, we do not match points but instead features that resemble structural or textual lines in the document. We use the matched features to improve the geometric dewarping process by aligning the warped document images with the reference templates.

2.3 Evaluation Metrics

This section is based on the following publication:

Publication

Felix Hertlein et al. “Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping”. In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: 10.1007/S10032-023-00434-X

Since geometric dewarping and illumination correction are image-to-image translation tasks, the evaluation metrics measure the similarity between the generated and the reference images. The similarity is evaluated between the flat invoice image and the processed image, whether it is dewarped for geometric correction or light-corrected for illumination correction. Since not all metrics are symmetric, we refer to the flatbed invoice image as the reference image and the processed image as the source image. Note that the evaluation of both tasks individually cannot yield the optimal evaluation scores, as both warping and illumination negatively impact image similarity. Only the combination of both tasks can yield the optimal evaluation scores for a perfect model. Nevertheless, the evaluation of both tasks individually is valuable as we can compare the performance of the models with the state-of-the-art methods.

In the following, we describe established metrics in the field of geometric dewarping and illumination correction to evaluate the performance of these approaches. The metrics can be divided into visual and text-based metrics and are explained in the following sections.

2.3.1 Visual metrics

Visual metrics directly measure the perceived similarity between two images. In related work, common visual metrics include *MS-SSIM* [90], *LD* [97], and *LPIPS* [104], which we explain in the following. To make all visual metrics scale-invariant, both images are resized to a fixed area of 598400 pixels while retaining the ground truth aspect ratio similar to the work of Ma et al. [61].

MS-SSIM. Multiscale Structural Similarity (*MS-SSIM*) [90] is an established perceptual metric that measures the perceived change in structural information by calculating statistical properties on multiple image windows at different scales. The metric consists of multiple structural similarity (*SSIM*) calculations on different scales of input and reference image in order to become scale-invariant. Following the evaluation approach of Das et al. [16], the source and reference images are converted to grayscale to ensure comparable values before applying the metric. The *MS-SSIM* ranges between 0 and 1, whereas 1 denotes the optimal score.

LD. The local distortion (*LD*) as defined by You et al. [97] quantifies the similarity of two images based on the *SIFT* flow [52]. Input and reference images are converted to dense *SIFT* feature matrices and subsequently are matched pixel-wise to form the *SIFT* flow. The local distortion is defined as the mean L2-norm of the *SIFT* flow. We applied the implementation and parametrization of Ma et al. [61] to grayscale images, following the procedure of Das et al. [16]. The *LD* ranges between 0 and infinity, where 0 is the optimal score.

LPIPS. In addition to *MS-SSIM* and *LD*, the Learned Perceptual Image Patch Similarity (*LPIPS*) metric, introduced by Zhang et al. [104], is employed to measure perceived image similarity. The authors show that the *LPIPS* metric is better suited for measuring perceived image similarity than *SSIM*. The metric is learned using a large-scale similarity preference dataset. For our evaluation, we used the pre-trained weights provided by the authors based on the AlexNet [37] model. *LPIPS* ranges between 0 and infinity, where 0 denotes the optimal score.

2.3.2 Text-based metrics

For many use cases, such as information extraction, a text-based metric is better suited to evaluate geometric dewarping and illumination correction, as they focus on the text-readability within the images. Given both images to be compared, the text-based metrics evaluate the similarity by first extracting the text from the images using optical character recognition (*OCR*) and then comparing the extracted text. Following Das et al. [16], the open source engine Tesseract 4.0.0 [78] is used for *OCR*. In order to detect the text in images, the input image requires a sufficiently high resolution with respect to the contained text size. Therefore, we scaled both images to a size of 3740000 pixels while preserving the aspect ratio of the reference image. To evaluate the recognized text, there are two different metrics ED and CER as described below.

ED. The Edit Distance (*ED*), also known as the Levenshtein distance [43], is defined as the minimum number of insertions, deletions, and substitutions required to transform an input text to the corresponding reference text. The optimal score is 0, where the input text equals the reference text.

CER. The character error rate (*CER*) [1] is defined as the edit distance between input and reference divided by the number of characters in the reference text, i.e., the detected text in the flatbed invoice image. The normalization by the number of characters in the reference text allows for a length-independent comparison between texts of different lengths. Note that even though the *CER* is normalized with respect to the reference text length, it still ranges from 0 to infinity, as the length of the source image text can be arbitrarily long.

2.4 Summary

Summarizing the foundational concepts of this work, we began by introducing various coordinate transformations, including Frenet coordinates [91], sinusoidal positional encoding [84], homography [27], Delaunay triangulation [11], and dense coordinate transformations. Representing data using Frenet coordinates and sinusoidal positional encodings enables a compact and machine-learning-friendly representation of curves and their surroundings. Homography, Delaunay triangulation, and dense coordinate mappings are used to map between 2D-dimensional coordinate systems to model geometric dewarping transformations. Homographies are limited to eight degrees of freedom, while Delaunay triangulation allows for interpolating sparse control points to generate free-flow dense mappings.

In Section 2.2, we introduced the deep learning architectures used in this work, including the Transformer [84], Vision Transformer [19], Segment Anything [36], and LightGlue [49]. The transformer architecture is the fundamental building block for the other architectures. Vision Transformer is introduced since it is the core model for our approaches to geometric dewarping and illumination correction. We explain the Segment Anything architecture, which we use, for instance, in the segmentation of document images in this work. Lastly, we introduce LightGlue, a model capable of matching local features across images. We use LightGlue to match features across the warped documents and the reference templates to improve the geometric dewarping process.

Lastly, we introduced the established evaluation metrics for geometric dewarping and illumination correction, including visual metrics such as *MS-SSIM* [90], *LD* [97], and *LPIPS* [104], as well as text-based metrics such as *ED* [43] and *CER* [1].

3 Related Work

In this chapter, we present an overview of the related work with regard to our research questions. In general, document image enhancement is performed in two stages. First, the geometric distortions are corrected, followed by the correction of lighting artifacts in the captured image. We divide the literature accordingly into three categories: (1) geometric dewarping (Section 3.1), (2) illumination correction (Section 3.2), and (3) datasets (Section 3.3). In the following, we present the relevant works from these areas and highlight the differences in the approaches described in this thesis.

3.1 Geometric Dewarping

Document image dewarping is a widely researched topic in document analysis aimed at correcting geometric distortions in document images to enhance subsequent analysis tasks. However, the geometric dewarping task is challenging due to the complex deformations that can occur in document images.

In recent years, several approaches have been proposed to tackle the geometric dewarping problem. In Table 3.1, we present an overview of the different approaches for geometric dewarping over the last several years. Further details on the proposed approach are provided along with the paper reference, year of publication, and a short name for the architecture. In column *Base Architecture*, we state the employed machine learning architecture or established network for each work. The remaining columns show the following attributes, which can help to distinguish between the approaches:

- *Multiscale* approaches consider the input image or the internal features at different resolutions to enable the network to handle local and global deformations differently.
- *Patch-based* works break down the dewarping task by subdividing the input image in smaller patches, solve the dewarping on the patch level, and stitch the results together.
- *Line-based* approaches focus on visual lines in the documents, such as text lines, structural lines, and borders of the documents.
- *Iterative* papers try to generate an initial dewarping guess and progressively update/improve the dewarping guess until there is a termination condition.
- *Model-based* approaches assume a specific model for the document deformation and try to estimate the model's parameters.
- And lastly, we highlight solutions that leverage reference templates in addition to the input image as an aid for the model on how to dewarp the input image.

Reference	Year	Name	Base Architecture	Multiscale	Patch-based	Line-based	Iterative	Model-based	Leverage Templates
Ma et al. [61]	2018	DocUNet	U-Net	✓	-	-	-	-	-
Meng et al. [64]	2018	-	-	-	-	-	-	✓	-
Li et al. [46]	2019	DocProj	*CNN*	-	✓	-	-	-	-
Das et al. [16]	2019	DewarpNet	U-Net	✓	-	-	-	-	-
Burden et al. [7]	2019	-	-	-	-	✓	-	✓	-
Ramanna et al. [73]	2019	pix2pixHD	*CGAN*	-	-	-	-	-	-
Markovitz et al. [62]	2020	CREASE	U-Net + DenseNet	✓	-	-	-	-	-
Xie et al. [92]	2020	-	*CNN*	✓	-	-	-	-	-
Liu et al. [56]	2020	AGUN	U-Net	✓	-	-	-	-	-
Vinod and Niranjan [86]	2020	-	-	-	-	✓	-	-	-
Feng et al. [21]	2021	GeoTr	Transformer	-	-	-	-	-	-
Xie et al. [93]	2021	DDCP	*CNN*	-	-	-	-	-	-
Das et al. [18]	2021	-	U-Net + DenseNet + *FPN*	✓	✓	-	-	-	-
Garai et al. [24]	2021	-	*CNN*	-	-	✓	-	✓	-
Simon and Tabbone [75]	2021	-	-	-	-	-	-	✓	-
Bandyopadhyay et al. [5]	2021	RectiNet-v2	U-Net + *CNN*	✓	-	-	-	-	-
Xue et al. [96]	2022	FDRNet	*CNN*	-	-	✓	-	-	-
Jiang et al. [34]	2022	-	U-Net	✓	-	✓	-	-	-
Zhang et al. [102]	2022	Marior	*CNN*	-	-	-	✓	-	-
Ma et al. [60]	2022	PaperEdge	*CNN*	-	-	✓	-	-	-
Feng et al. [23]	2022	DocGeoNet	*CNN*	-	-	✓	-	-	-
Feng et al. [22]	2022	DocScanner	*CNN* + *ConvGRU*	-	-	-	✓	-	-
Xu et al. [95]	2022	-	U-Net + DenseNet	✓	-	-	-	-	-
Das et al. [15]	2022	-	*MLP*	-	-	-	-	-	-
Luo and Bo [59]	2023	-	-	-	-	✓	✓	-	-
Feng et al. [20]	2023	DocTr++	*CNN* + Transformer	✓	-	-	-	-	-
Zhang et al. [100]	2023	DocAligner	*CNN* + *ConvGRU*	✓	-	✓	✓	-	-
Dai et al. [14]	2023	MataDoc	Transformer	-	-	✓	-	-	-
Li et al. [44]	2023	-	*CNN* + Transformer	-	-	✓	-	-	-
Li et al. [45]	2023	-	*CNN*	-	✓	-	-	-	-
Liu et al. [53]	2023	-	Transformer	-	-	✓	-	-	-
Nachappa et al. [66]	2023	-	-	-	-	✓	-	-	-
Zhang et al. [106]	2023	Polar-Doc	Transformer	-	-	-	-	-	-
Liu et al. [54]	2023	DocMAE	Masked *AE*	-	-	-	-	-	-
Yu et al. [98]	2024	DocReal	*CNN* + Transformer	-	-	✓	-	-	-
Kumari and Das [39]	2024	DocTLNet	*CNN*	✓	-	-	-	-	-
Hertlein et al. [31] (ours)	**2023**	**GeoTrTemplate**	**Transformer**	**-**	**-**	**-**	**-**	**-**	**✓**
Hertlein et al. [32] (ours)	**2025**	**DocMatcher**	**Transformer**	**-**	**-**	**✓**	**-**	**-**	**✓**

Table 3.1: Overview of the related work on geometric dewarping. We list our publications for completeness.

As indicated by the table, to the best of our knowledge, there are no publications that leverage reference template information except our own publications. Note that a given paper may fit into multiple categories or might not be associated with any of them.

Multiscale approaches. Many works are based on multiscale approaches [61, 16, 73, 62, 56, 5, 18, 34, 95, 20, 100]. Ma et al. [61] presented a pioneering approach to this problem called DocUNet. The approach is based on two stacked U-Nets, one for estimating the forward map and the second for inverting the forward map to get a backward map. Given the architecture of the U-Nets, this approach handles features representing the input image at different scales. Based on this work, Bandyopadhyay et al. [4] propose to integrate gated convolutional layers in the architecture to help the network focus on the line-level document features. Similar to [61], Das et al. [16] also employ U-Nets for dense grid estimations, but contrary to the prior works, they lift the problem from 2D to 3D. Instead of searching for a 2D displacement map (forward map), they estimate the 3D positions of all pixels before inferring the backward map based on the 3D positions. Markovitz et al. [62] approach the problem similarly via the intermediate 3D map and integrate additional losses for the document curvature and local angles to straighten it further. Unlike previous works, Liu et al. [56] present a pyramid encoder-decoder architecture that concatenates the input image at different scales explicitly. Xie et al. [92] propose a hierarchical Convolutional Neural Network (*CNN*) similar to a U-Net without skip-connections to estimate the displacement flow and the document segmentation. Additionally, they present a local smoothness constraint for regularization during training, which preserves local document details. Das et al. [18] employ a U-Net module to estimate the 3D shape of the document similar to [16], before dividing the overall document into patches for further processing. Contrary to the previous approaches, Jiang et al. [34] leverage the power of DocUNet and U-Net to detect the boundaries and the text lines of the document, which then build the basis for a grid regularization in order to determine the dense backward map. Similar to [16, 18], Xu et al. [95] employ a U-Net as a backbone to reconstruct the 3D coordinates of the document. The authors extend the previous works by training the model as a Siamese Network, i.e., using contrasting features from contrasting input images. The approaches DocTr++ and DocAligner by Feng et al. [20] and Zhang et al. [100] are both primarily based on a transformer architecture, which is adapted to handle image features at different resolutions. Most recently, Kumari and Das [39] proposed a new hierarchical geometric dewarping model called DocTLNet based on transfer learning. Their method is trained to simultaneously categorize the distortion type and estimate the backward map using a U-Net like architecture.

Patch-based approaches. The first patch-based approach to document dewarping was presented by Li et al. [46] in 2019. Their model slices the input image into patches, dewarps the patches individually using a *CNN*, and stitches the resulting flows back together. The advantage of this approach over full image networks lies in the problem simplification since the geometric deformation in each patch is less complex than the overall deformation. Das et al. [18] propose a model that attempts to infer both the global warping and the local warping per patch before merging all of them together. Notably, their model is end-to-end trainable since they employ a neural network to combine all warp information. Recently, Li et al. [45] proposed a patch-based layout-aware model. The authors employ a segmentation model to yield layout-specific segmentation masks, which are then leveraged in the patch-merging process.

Line-based approaches. Since documents contain visually rich structures such as text lines, table lines, and document borders, many approaches were presented that detect those features in the warped document image and leverage this knowledge in the dewarping process. Earlier line-based approaches (since 2018) were presented by Burden et al. [7], Vinod and Niranjan [86], and Garai et al. [24]. All three models only consider simple deformations of the document, for instance curls, which simplifies the dewarping task significantly. In contrast, Jiang et al. [34] propose a line-based model which can also process complex deformations. They detect the document boundaries and the text lines before determining the backward map using grid regularization. Crucially, this method assumes horizontally and vertically aligned lines in the document. Similarly, Luo and Bo [59] consider the dewarping problem a constrained optimization problem based on the document boundaries and text lines. In contrast to the prior method, the constraints are formulated differently. They constrain the text lines only to be straight and the paper sheet to be non-stretchable. In a different approach, many works first pre-dewarp the input images before performing a follow-up fine-grained dewarping step. This involves implicitly or explicitly detecting the document boundaries and using them to partially dewarp the document, ensuring it fills the entire image space [60, 100, 98]. This simplifies the task for the downstream network, as all pixels can be assumed to be within the document borders. Similarly, Xue et al. [96] and Liu et al. [53] build on a two-stage process: a coarse and a refinement transformer. In contrast to the three works mentioned above [60, 100, 98], the edge is not explicitly represented but rather a full coarse displacement field. The former approach [96] is unique since the training partially takes place in the Fourier space, enabling low-pass filtering of the image. This focuses the model on the details rich in information. Interestingly, in the latter approach [53], the authors introduce a self-consistency loss based on the observation that the rectified image should be identical for multiple warped document images, which were derived from the same base image. Differently from the previous works, Feng et al. [23] and Dai et al. [14] both define additional losses based on the text lines within a document. The prior is a transformer-based approach intended to learn the global 3D coordinates and simultaneously the text mask. The latter, MataDoc, is trained with losses based on the text lines and document borders. In contrast to prior work, MataDoc was designed to dewarp partially visible documents as well. Li et al. [44] explicitly detect the text lines and document mask and use this information to cross-attend between both branches. Vastly different from all previous works, Nachappa et al. [66] propose a non-learning-based approach to dewarp documents. They detect control points along the document borders using the Hough transformation, followed by a cylindrical surface projection as proposed in [63].

Iterative approaches. The model DocScanner presented by Feng et al. [22] contains a recurrent architecture based on a convolution-based variant of Gated Recurrent Units (*GRU*s) [12]. With this recurrent model, they keep a single estimate of the dewarping flow and iteratively refine it. Thus, their model is relatively lightweight compared to a single-pass network. With Marior, Zhang et al. [102] propose a different iterative approach. Instead of a recurrent machine learning network that contains the current dewarping encoded in its state, they infer the backward map multiple times and apply it directly to the input image to reduce the necessary displacement with each iteration. Unlike previous works, Luo and Bo [59] present a non-learning-based iterative approach. Instead of a neural network, they search iteratively for an isometric mapping from the 3D to the 2D plane, which follows the constraints of the document boundaries and text lines. In 2023, Zhang et al. [100] pick up the idea of using convolution-based *GRU*s for refining their estimate of a backward map in a more detailed resolution. Their previous step yields a flow estimation of $1/4$ the size of the input image, and their refinement module allows for a refinement of said flow in full resolution.

Model-based approaches. Some approaches are based on a model of the document deformation [64, 7, 24, 75]. These works typically assume that a simple parametrized model can describe the deformation. Meng et al. [64] and Simon and Tabbone [75] both utilize a cylindrical surface model to represent document deformation. In contrast, Burden et al. [7] propose a more complex model that combines multiple curved surface models to represent text and non-text regions individually. Lastly, Garai et al. [24] introduce a mathematical model of warping, which is capable of describing three patterns for curling and propose a *CNN*-based approach to find these dewarping factors.

Other approaches. Not all approaches fall into any of the above-mentioned categories. Ramanna et al. [73] consider the dewarping problem as an image-to-image translation task and employ a Conditional Generative Adversarial Network (*CGAN*) in their approach. Xie et al. [93] present an encoder-based approach that aims to detect sparse control points mapped to a regular grid and interpolate the dewarping map between the control points. Sparse control points allow for a simple human interaction when correcting suboptimal points. Zhang et al. [106] likewise attempt to detect control points, the 3D shape and points along the border. The authors propose to represent these points in polar coordinates. In 2021, Feng et al. [21] proposed using attention-based models for geometric dewarping for the first time. They divide the input image into patches, deep encode them, and let the patches attend to each other using a transformer architecture before recombining and upscaling the resulting features. In contrast to all previous works, the approach DocMAE by Liu et al. [54] uses self-supervised representation learning to transfer the document image encoding capabilities from autoencoders to a dewarping network. In a pretraining stage, they train an autoencoder (*AE*) to fill in the gaps in a masked image. The encoder weights are later used in an encoder-decoder architecture for displacement flow prediction. This approach enables the pretraining of the encoder on large-scale unlabeled datasets, thus reducing the gap between the synthetic annotated data domain and the real-world applications. In a fundamentally different approach, Das et al. [15] employ neural rendering to learn an implicit surface representation of a dewarped document from multiple views of the document. This approach not only allows for document dewarping, but also for texture replacement in the warped domain.

3.1.1 Discussion

In contrast to prior work, our attempt to solve the geometric dewarping problem uses reference templates as an additional input along with the warped document images. To the best of our knowledge, there are no previous works for document dewarping with the help of reference templates in the era of deep learning. In this thesis, we investigate the benefits of the provided reference template. In our evaluation, we will compare ours and the state-of-the-art papers to show the benefit of reference templates. When comparing our methods to the related work, it is crucial to remember that our approaches have more information available in both training and inference time. While the availability of reference templates is less flexible than template-free methods, many real-world use cases exist where reference templates are available or can be provided easily to facilitate the dewarping problem. See Section 1.2.1 for more details on the applicability of our approach.

Our second approach, DocMatcher, can also be categorized as line-based since we explicitly detect and match structural and text lines from the warped domain to the normalized domain. We are not the first to explicitly detect the text lines within the document [34, 59]. However, since their approaches do not have reference templates available, thus, it cannot leverage the knowledge about the true position of each line in the flatbed document. Instead, they attempt to straighten and axis align the detected lines. We, on the other hand, can associate the warped lines to the template lines and, thus, know the absolute position for said line within the document.

3.2 Illumination Correction

This section is based on the following publication:

Publication

Felix Hertlein and Alexander Naumann. "Template-guided Illumination Correction for Document Images with Imperfect Geometric Reconstruction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. IEEE, 2023, pp. 904–913. DOI: 10.1109/ICCVW60793.2023.00097

Illumination correction can be subdivided into two categories: Document Image Binarization (*DIB*) and Document Image Enhancement (*DIE*). *DIB* maps all colors to a binary signal, effectively reducing the image to a black-and-white representation. On the other hand, *DIE* aims to enhance the document image by attempting to restore the original colors. Table 3.2 provides an overview of the related work on illumination correction.

3.2.1 Document Image Binarization

In the past, various works have been conducted on document image binarization [58, 82, 50, 3, 51, 8, 35, 79, 26, 80, 66, 67, 103]. Early work was presented by Lu et al. [58] and Su et al. [82]. A diverse array of binarization algorithms was assessed by Lins et al. [50]. Almeida et al. [3] proposed an approach for image binarization inspired by Otsu's method [70] for pixel thresholding. In 2019, the International Conference on Document Analysis and Recognition (ICDAR) hosted a binarization competition that included a benchmark of thirty distinct binarization algorithms [51]. Calvo-Zaragoza and Gallego [8] presented a convolutional auto-encoder architecture. DE-*GAN* by Souibgui and Kessentini [81] uses conditional Generative Adversarial Networks for multiple document enhancement tasks. Kang et al. [35] proposed a document binarization method using cascading UNets to cope with the limited number of training images. In 2022, Souibgui et al. [79] presented DocEnTr, an encoder-decoder architecture based on vision transformers without any convolutional layers. Gonwirat and Surinta [26] proposed DeblurGAN, a *CNN*-*GAN* hybrid, for enhancing noisy handwritten characters. In 2023, Souibgui et al. [80] introduced Text-DIAE, a transformer-based model that utilizes self-supervised pretraining to enhance document binarization. Contrary to previous work, Nachappa et al. [66] integrate a traditional binarization algorithm called Adaptive Thresholding [6] in their dewarping and illumination correction pipeline. Recently, Neji et al. [67] proposed a binarization model called Doc-Attentive-GAN. The core idea is to use attention maps to support the generator's focus on the transformation from the input domain to the binarized image version. Finally, Zhang et al. [103] presented a unified model for multiple document image restoration tasks, including *DIB* and *DIE*. The authors present one feature extraction method per task to yield a unified input format, which is then restored using a transformer-based architecture called Restormer [99].

3.2.2 Document Image Enhancement

Significant advancements have been made in the field of *DIE* recently. Das et al. [16] introduced a refinement network based on a stacked U-Net architecture, as outlined in their work titled DewarpNet. The proposed network aims to predict a shading map, which is subsequently applied

Reference	Year	Name	Base Architecture	Binarization	Enhancement	Leverage Templates
Lu et al. [58]	2010	-	-	✓	-	-
Su et al. [82]	2012	-	-	✓	-	-
Lins et al. [50]	2017	-	-	✓	-	-
Almeida et al. [3]	2018	-	-	✓	-	-
Lins et al. [51]	2019	multiple	various	✓	-	-
Calvo-Zaragoza and Gallego [8]	2019	SAE	*AE*	✓	-	-
Das et al. [16]	2019	DewarpNet	U-Net	-	✓	-
Li et al. [46]	2019	DocProj	*CNN*	-	✓	-
Souibgui and Kessentini [81]	2020	De-gan	*GAN*	✓	-	-
Lin et al. [48]	2020	BEDSR-Net	*CNN* + *CGAN*	-	✓	-
Kang et al. [35]	2021	CMU-Nets	U-Net	✓	-	-
Feng et al. [21]	2021	IllTr	Transformer	-	✓	-
Souibgui et al. [79]	2022	DocEnTr	Transformer	✓	-	-
Gonwirat and Surinta [26]	2022	Deblurgan-cnn	*GAN* + *CNN*	✓	-	-
Xue et al. [96]	2022	FDRNet	*CNN*	-	✓	-
Wang et al. [88]	2022	UDoc-GAN	*GAN*	-	✓	-
Souibgui et al. [80]	2023	Text-DIAE	*AE*	✓	-	-
Nachappa et al. [66]	2023	-	-	✓	-	-
Zhang et al. [101]	2023	GCDRNet	U-Net	-	✓	-
Neji et al. [67]	2024	Doc-Attentive-GAN	*GAN*	✓	-	-
Zhang et al. [103]	2024	DocRes	Transformer	✓	✓	-
Kumari and Das [39]	2024	DocTLNet	*CNN*	-	✓	-
Hertlein and Naumann [30] (ours)	**2023**	**IllTrTemplate**	**Transformer**	**-**	✓	✓

Table 3.2: Overview over the related work on illumination correction. We list our publications for completeness.

to the dewarped image using element-wise division. Li et al. [46] proposed a convolutional network with residual layers to infer the illumination-corrected image directly. In 2021, Feng et al. [21] introduced a transformer encoder-decoder structure for document image enhancement called IllTr. Their approach splits the input document image into a sequence of patches, which are then processed individually and stitched together afterwards. Xue et al. [96] proposed to remove illumination artifacts by transforming the input image and a blank paper to the Fourier space and then replacing the lower frequencies in the input image with the frequencies of the blank paper. In contrast to the previous works, Lin et al. [48] and Wang et al. [88] focus on *GAN*s for *DIE*. Both works put a preparation network in front of their *DIE* network, whose task is to detect global priors such as the document background color or global illumination. Furthermore, Zhang et al. [101] present a stacked U-Net architecture with multiple losses at different scales. The prior U-Net processes a down-scaled version of the document in order to capture global context, whereas the latter U-Net focuses on the details. In 2024, Kumari and Das [39] present DocTLNet, a network for geometric dewarping and illumination correction. The latter uses a *CNN* with skip connections. Lastly, there is DocRes, the unified approach for multiple document image improvement tasks (including *DIE*) by Zhang et al. [103]. See Section 3.2.1 for a brief explanation.

3.2.3 Discussion

In this thesis, we propose a document image enhancement approach called IllTrTemplate. Our approach integrates reference templates as an additional prior in contrast to the previous approaches. To the best of our knowledge, there is no other work that leverages reference templates for illumination correction in the deep learning era. The work of Xue et al. [96] comes closest to the idea of templates, since they replace the low-frequency components of the dewarped image with the respective components of a blank paper. Thus, the blank paper can be considered a primitive template as it provides information about the expected appearance. In contrast to this work, we integrate and leverage full reference templates, so our references contain not only information about the expected structures and text in the document but also information regarding the true colors of the document.

Similar to many related works [16, 46, 21, 96, 66, 39], we consider the illumination correction as a downstream task after geometric dewarping. Since geometric dewarping is not entirely solved, there are still some geometric distortions in the input images for illumination correction. In contrast to the previous works, we investigate the effects of the remaining geometric distortions at different levels.

3.3 Datasets

In order to train and evaluate deep learning models, there is a need for realistic large-scale datasets. Over the years, numerous datasets for geometric dewarping and illumination correction have been created and, in many cases, published. We list relevant datasets and their key characteristics in Table 3.3. For each dataset, the table denotes the authors, year of publication, dataset name, number of samples, and whether the dataset is generated synthetically or not. Furthermore, we state the following binary flags:

1. *Public Availability.* The first flag indicates whether the dataset was made public. Note that a few datasets are intended to be public but cannot be downloaded at the time of writing.
2. *3D.* The 3D flag indicates whether the document deformations are in 3D or 2D.
3. *Crumples.* With the third flag, crumples, we indicate the existence of crumples in the deformed documents since they are the hardest kind of deformation due to their fine-grained displacements.
4. *High Resolution.* The high-resolution flag is true if at least one side of the document images has a resolution of 1000 pixels or higher. For the downstream task of information extraction, text readability is essential, which requires high-resolution images.
5. *Pixel-wise GT.* With flag five, we indicate whether a dataset has per-pixel ground truth annotations for geometric dewarping, i.e., a forward or backward map.

6. *Flat Document.* The sixth flag indicates the availability of the pristine flat document for each warped document.
7. *Reference Templates.* And lastly, we show whether reference templates are available for each sample in the respective dataset.

Note that we could not determine all values in the table for unpublished or unavailable datasets. We marked those values with ?.

The datasets can be divided into three principal categories based on their generation process: (1) synthetic, (2) real, and (3) hybrid. The following sections give more details on the datasets and the differences between them.

3.3.1 Synthetic datasets

Since synthetic data generation can be fully automated, synthetic datasets are generally larger than real-world datasets. In literature, there are eight datasets with at least 10k samples [61, 16, 62, 93, 55, 23, 20, 100] and only two with fewer samples [46, 4]. Another advantage of synthetic data generation is that there is full knowledge about the ground truth dewarping available. Thus, all synthetic datasets, without exception, provide the ground truth per-pixel dewarping annotations. There are two fundamental ways to generate synthetic samples: 2D and 3D. The 2D approaches take a flat mesh and apply various 2D deformations to the mesh in order to generate the ground truth deformation, which is then applied to a flat document image [61, 92, 4, 93, 55, 100]. One example of such a 2D deformation could be shifting control points and proportional editing of the surrounding mesh. On the other hand, 3D-based synthetic dataset generation pipelines use a 3D rendering engine in order to simulate the document under configurations that are as realistic as possible [16, 46, 62, 23, 20, 45]. When comparing the 2D and the 3D-based approaches, the former method is significantly faster to compute whilst still generating reasonably realistic samples. On the contrary, 3D-based methods are highly computationally expensive but generate the most realistic synthetic samples since the rendering engine can simulate the physical piece of paper and realistic lighting conditions. Moreover, another significant difference to the previous datasets was introduced by Li et al. [45] and Zhang et al. [100]. The datasets resulting from the two works both provide details about the layout of the warped document in the form of segmentation masks and labels. This layout information is suitable for training models to recognize and consider the layout blocks in the document. However, these annotations cannot be used to generate reference templates, as they do not distinguish between fixed layout elements (e.g., logos) and content-dependent layout elements (e.g., shipping addresses).

3.3.2 Real datasets

Since the capturing process of real document datasets cannot be fully automated and consequently requires manual labor, the dataset sizes are generally significantly smaller than synthetic datasets. In literature, we found eight datasets with less than 1.1k samples [61, 24, 96, 55, 23, 20, 66, 108] and only three larger datasets [60, 100, 107]. Moreover, the larger datasets are still relatively small compared to synthetic datasets. Generating real datasets generally includes capturing real-world physical documents with cameras and associating these warped images with the pristine versions of the document, either the unprinted digital version or a scan of the physical document. Unlike simulated data, all real datasets lack precise per-pixel ground truth annotations. This is

Reference	Year	Name	Num. Images	Synthetic / Real	Public Availability	3D	Crumples	High Res.	Pixel-wise GT	Flat Document	Reference Templates
Ma et al. [61]	2018	DocUNet	100k	synthetic	-	-	-	-	✓	✓	-
Ma et al. [61]	2018	DocUNet Bench.	130	real	✓	✓	-	✓	-	✓	-
Das et al. [16]	2019	Doc3D	100k	synthetic	✓	✓	✓	-	✓	-	-
Li et al. [46]	2019	DRIC	2450	synthetic	✓	✓	-	✓	✓	✓	-
Markovitz et al. [62]	2020	CREASE	15k	synthetic	-	✓	✓	✓	✓	-	-
Das et al. [17]	2020	Doc3DShade	90k	hybrid	✓	✓	-	-	-	-	-
Xie et al. [92]	2020	DIWF	80k	synthetic	-	-	-	✓	✓	✓	-
Bandyopadhyay et al. [4]	2021	RectiNet	8k	synthetic	-	-	-	?	✓	?	-
Xie et al. [93]	2021	DDCP	30k	synthetic	-	-	-	✓	✓	✓	-
Garai et al. [24]	2021	WDID	258	real	✓	✓	-	✓	-	✓	-
Ma et al. [60]	2022	DIW	5k	real	✓	✓	✓	-	-	-	-
Xue et al. [96]	2022	WarpDoc	1020	real	✓	✓	✓	✓	-	✓	-
Liu and Liu [55]	2022	ADIU-syn	10.8k	synthetic	-	-	-	?	✓	✓	-
Liu and Liu [55]	2022	ADIU-real	200	real	-	✓	-	?	-	✓	-
Feng et al. [23]	2022	DIR300 train	100k	synthetic	-	✓	✓	-	✓	✓	-
Feng et al. [23]	2022	DIR300 test	300	real	✓	✓	✓	✓	-	✓	-
Li et al. [45]	2023	SP	?	synthetic	-	✓	✓	?	✓	?	-
Verhoeven et al. [85]	2023	UVDoc	20k	hybrid	✓	✓	✓	✓	✓	✓	-
Zhang et al. [100]	2023	DocAlign12k	12k	synthetic	-	-	-	?	✓	✓	-
Zhang et al. [100]	2023	DocAligner-applied	2.5k	real	✓*	✓	✓	✓	✓†	✓	-
Feng et al. [20]	2023	UDIR train	100k	synthetic	-	✓	✓	-	✓	✓	-
Feng et al. [20]	2023	UDIR test	195	real	✓	✓	-	✓	-	✓	-
Nachappa et al. [66]	2023	multiple	958	real	✓*	✓	?	✓	-	?	-
Zhang et al. [107]	2024	DocReg	12.5k	real	✓*	✓	?	✓	✓†	✓	-
Zhang et al. [108]	2024	WarpDoc-R	840	real	✓*	✓	✓	✓	✓†	✓	-
Hertlein et al. [31] (ours)	**2023**	**Inv3D**	**25k**	**synthetic**	✓	✓	✓	✓	✓	✓	✓
Hertlein et al. [31] (ours)	**2023**	**Inv3DReal**	**360**	**real**	✓	✓	✓	✓	**-**	✓	✓

Table 3.3: Overview over the related datasets on geometric dewarping and illumination correction. We list our datasets for completeness. The public availability information is stated as of November 9, 2024. Attributes marked with the ? operator indicate unknown values due to the dataset's unavailability to the public. Datasets marked with ✓* are intended to be publicly accessible but are currently unavailable due to technical issues. Datasets marked with ✓† have estimated per-pixel annotations instead of true flow fields. Parts of this table were previously published in [31].

because even for humans, it is unclear where a pixel from the warped domain should correspond in the flat domain. Due to this problem, most real datasets do not provide any per-pixel ground truth annotations [61, 24, 60, 96, 55, 23, 66, 20], which poses a unique challenge for training and evaluation of AI models. Recently, three datasets were proposed to circumvent the lack of per-pixel annotations [100, 107, 108]. Zhang et al. [100] first train a model on synthetic data, then apply the trained model on real data to obtain the pixel alignment, and lastly, retrain their model with the newly generated real-world dataset. Differently, Zhang et al. [107] and Zhang et al. [108] propose document registration approaches, i.e. given a warped document and the flat image as input, the goal is to find the true per-pixel annotation. Document registration methods are the closest approach in the literature to our concept of using reference templates for document dewarping. In document registration, a flat, pristine version of the document is available for inference, whereas our approach relies solely on template images that can be prepared in advance. It is crucial to note that all three approaches to determining the true per-pixel annotations yield approximated flow fields whose quality depends on the performance of the employed models. However, this problem does not exist for synthetic data, as true per-pixel annotations can be extracted from the simulation process.

3.3.3 Hybrid datasets

Hybrid datasets [17, 85] combine synthetic and real dataset generation techniques. The idea is to capture and isolate the lighting conditions in real settings and apply those to synthetic textures. This approach enables the generation of large-scale datasets while simultaneously enabling realistic illumination. The most significant difference between the datasets of Das et al. [17] and Verhoeven et al. [85] is that the former lacks dewarping ground truth and, therefore, is limited to illumination correction, while the latter can be used for both dewarping and illumination correction.

3.3.4 Discussion

In this work, we propose two datasets: (1) a synthetic dataset called Inv3D with 25k samples and (2) a real dataset with 360 samples called Inv3DReal. In contrast to all prior work, both datasets provide the correct reference template along with the warped document and the flat document. To the best of our knowledge, there are no datasets for document dewarping and illumination correction that include said reference templates. The synthetic dataset is based on the rendering pipeline of Doc3D [16], which is why the dataset characteristics are largely similar to the ones of Doc3D. The main differences are the document type, dataset size, resolution of the warped document, and the availability of a reference template. Like most real datasets, our dataset Inv3DReal is relatively limited in size and does not provide per-pixel annotations. However, unlike synthetic datasets, it does not contain rendering artifacts which might affect the generalization of models trained on synthetic data to real-world applications. Contrary to the widely used benchmark DocUNet [61], our dataset is generated systematically by applying different deformation categories per document to allow for a comparison based on the deformation type.

3.4 Summary

In this chapter, we reviewed related work on geometric dewarping and illumination correction, covering both the approaches and the datasets, and highlighted how our contributions relate to them. To the best of our knowledge, our key idea of utilizing a priori known reference templates is entirely novel in the field of deep learning. We focused our research on related work from the years 2018 and newer since the deep learning field is rapidly evolving, and thus, older publications lose relevance quickly.

In Section 3.1, we reviewed the literature on geometric dewarping techniques and showed the need for new approaches capable of leveraging additional information. Furthermore, we compared our novel contributions, GeoTrTemplate and DocMatcher, with the existing methods. To facilitate this comparison, we categorized the geometric dewarping approaches based on five key characteristics: multiscale, patch-based, line-based, iterative, and template-based. Section 3.2 focused on related work in illumination correction, divided into document image binarization and document image enhancement. Our literature review showed a research gap in the field of illumination correction, which we aim to fill with our novel approach IllTrTemplate. We compared our approach to the existing methods and highlighted the differences. Lastly, in Section 3.3, we listed and categorized datasets for geometric dewarping and illumination correction, describing their characteristics using nine attributes. We highlighted the differences between existing datasets and our proposed datasets, Inv3D and Inv3DReal.

Part III

Data Generation

4 Problem Analysis

As shown in the related work (see Chapter 3), there is no suitable dataset for geometric dewarping and illumination correction using reference templates. We fill this research gap by creating a large-scale dataset with full annotations and reference templates for model training called Inv3D. In addition, we create an evaluation dataset named Inv3DReal, consisting of samples captured in a real-world setting.

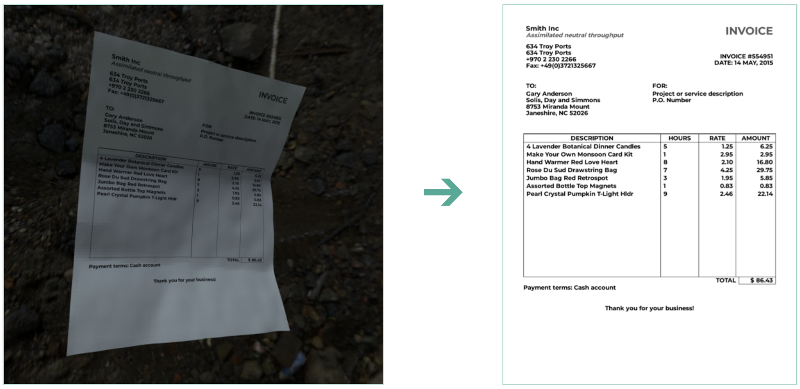
First, we start this chapter by analyzing the requirements for the tasks of geometric dewarping and illumination correction in Section 4.1. We differ between the requirements for training and evaluation datasets. Thereafter, we discuss the advantages and disadvantages of synthetic and real data generation approaches in Section 4.2. Based on this analysis, we justify using the respective technique for the data generation tasks at hand.

4.1 Requirements

This section covers the requirements for a suitable dataset to address the tasks of document image dewarping and illumination correction. Both tasks require an RGB image $\mathbf{W} \in \mathbb{R}^{h_0 \times w_0 \times 3}$ as input showing the surface of a potentially deformed document, where w_0 and h_0 are the width and height of the image, respectively. For a complete listing of all deformations considered in this thesis, please refer to Section 1.1. The input image $\mathbf{W}$ must fully show the document sheet, as the geometric dewarping cannot reconstruct all cut-off parts. See Figure 4.1 for an example of an input image.

We denote the corresponding flatbed document image as $\hat{\mathbf{I}} \in \mathbb{R}^{h_1 \times w_1 \times 3}$ with w_1 and h_1 being the width and height of the flatbed document image, respectively. It shows the filled document in its pristine condition, i.e., the condition before printing and warping the document sheet. The flatbed document image can be obtained by digitizing a physical document using a scanner or rendering a digital document. The objective for document image enhancement, i.e., the combination of geometric dewarping and illumination correction, is to determine the mapping from $\mathbf{W}$ to $\hat{\mathbf{I}}$, i.e., the function f such that $f(\mathbf{W}) = \hat{\mathbf{I}}$. Since we employ supervised deep learning methods, the dataset must provide the desired flatbed image $\hat{\mathbf{I}}$ as ground truth for the training process. See Figure 4.1 for an example.

Since we want to leverage a-priori-provided information in the form of reference templates for both tasks, a suitable dataset must also provide these templates. A reference template is defined as an RGB image $\mathbf{T} \in \mathbb{R}^{w_1 \times h_1 \times 3}$ showing all information shared between all instances of a specific document category or design. That is, the references templates display visual elements such as table lines, logos, background colors, and more. In addition, reference templates contain labels used to indicate the document's structure, such as "Bill To:". Note that the dimensions and the



Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
Fax: +49(0)3721325667

INVOICE

INVOICE #554951
DATE: 14 MAY, 2015

TO:
Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
		TOTAL	$ 86.43

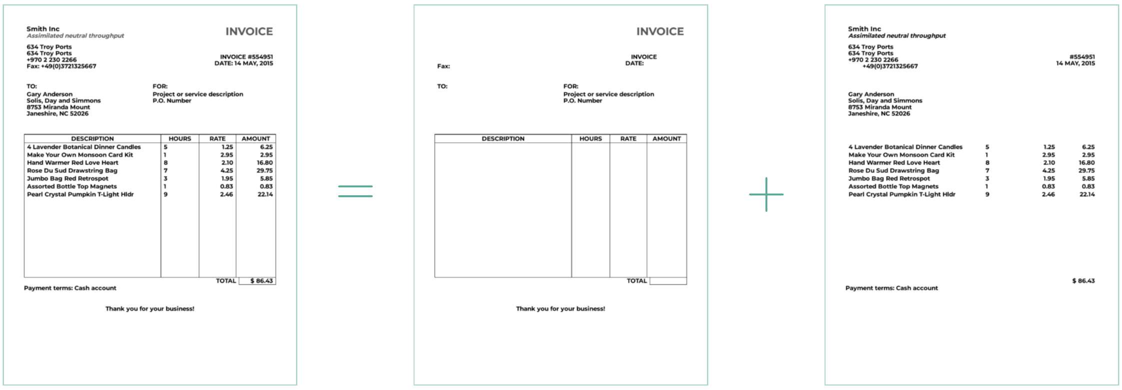
Payment terms: Cash account

Thank you for your business!

Input: Warped Image $\mathbf{W}$

Output: Flatbed Document $\hat{\mathbf{I}}$

Figure 4.1: Display of the initial input image and the desired image after perfect geometric dewarping and illumination correction.



INVOICE

INVOICE
DATE:

Fax:

TO:

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
		TOTAL	

Thank you for your business!

Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
+49(0)3721325667

#554951
14 MAY, 2015

Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
			$ 86.43

Payment terms: Cash account

Flatbed Document $\hat{\mathbf{I}}$

Template $\mathbf{T}$

Information Delta

Figure 4.2: Decomposition of a flatbed document into a reference template and matching information delta.

aspect ratios of the template $\mathbf{T}$ and the input image $\mathbf{W}$ do not necessarily match due to the image-capturing process by the camera. Given by the nature of the reference template, the aspect ratio of the template image $\mathbf{T}$ corresponds to the aspect ratio of the original document sheet. Figure 4.2 shows an example of a flatbed document $\hat{\mathbf{I}}$, as well as its decomposition in a reference template $\mathbf{T}$ and its corresponding information delta. The information delta is an image that highlights all the additional information present in the document but absent in the template, such as shipping address or invoice number.

Since the reference template does not contain any information about a template instance, it can be provided before attempting to geometrically dewarp and correct the lighting. Thus, it can be used as an input for both tasks at hand.

4.1.1 Training Requirements

We aim to solve both tasks using supervised deep learning. Therefore, our training dataset Inv3D needs to fulfill additional requirements.

First, our dataset needs to be sufficiently large to train deep learning models without overfitting them. The required number of samples depends highly upon the problem and the model employed; thus, it is usually determined empirically.

Secondly, the data must represent the entire domain to ensure that the model can effectively learn to address the complete problem space. This is achieved by ensuring much variety in the training data. The evaluation on a test dataset usually shows if the model is capable of transferring the learned knowledge to realistic settings.

Thirdly, we must provide the ground truth supervision signals for supervised *DL* training. For geometric document dewarping, the ground truth annotations are defined as a dense coordinate mapping as previously introduced in Chapter 2, namely the forward map $\hat{\mathbf{F}}$ and the backward map $\hat{\mathbf{B}}$. The former defines the transformation from the warped domain to the flatbed domain, while the latter maps the domains in the reverse direction.

Our second task, illumination correction, requires ground truth knowledge about the original colors of the document. This annotation must be given per pixel. We refer to the map containing ground-truth color values as the albedo map $\mathbf{A} \in \mathbb{R}^{h_a \times w_a \times 3}$, where w_a and h_a are the width and height of the albedo map, respectively. An albedo map, as defined in the context of computational graphics, contains the base color of the rendered objects, which equals the color of the object in a bright, evenly-distributed environment[1].

In addition to the ground truth annotations defined above, further per-pixel annotations can benefit a deep learning model, as more supervision signals can support the learning process. For instance, the dataset provides the 3D coordinates of the document for each pixel in the image. That knowledge can be used to train a 3D reconstruction module. Helpful annotations are, amongst others, depth maps, normal maps, reconstruction maps, word coordinate maps, warped angles maps, curvature maps, and text masks. In chapter 5, we provide a detailed overview of the annotations provided in our dataset. See Figure 5.3 for examples of those annotations.

4.1.2 Evaluation Requirements

To measure the capability of our approaches to solve both tasks, we need an evaluation dataset that is entirely independent of the training dataset. For this dataset, it is crucial to resemble the target domain as closely as possible to derive conclusions from the results. Optimally, the samples are directly drawn from a real-world application. Compared to the training dataset, the evaluation dataset can be considerably smaller.

Provided the ground truth backward map $\hat{\mathbf{B}}$, evaluating the geometric dewarping approaches is straightforward as one could compare it directly with the generated backward map $\mathbf{B}$ using the L_2 distance. Unfortunately, this kind of annotation is virtually impossible for real-world samples since it would require a human annotator to define an alignment between the warped and the flatbed domain for each pixel individually. Given the difficulties in obtaining the ground truth backward maps, we rely on alternative evaluation metrics for the geometric dewarping task, which are based on the comparison of the dewarped image $\mathbf{I_{dwp}}$ and the flatbed image $\hat{\mathbf{I}}$. Therefore, we can eliminate the need for the backward map as a requirement for the evaluation dataset.

[1] `https://www.a23d.co/blog/difference-between-albedo-and-diffuse-map` Accessed 19th of December 2024

Synthetic Data		Real Data
+ easy to generate		+ realistic data
+ large-scale data		- not scalable
+ pixel wise annotations	vs.	- expensive
+ no privacy issues		- privacy issues
+ no physical setup required		- physical setup required
- creates artifacts		- no pixel wise annotations

Figure 4.3: Advantages and disadvantages of synthetic versus real data generation.

4.2 Synthetic vs. real data

There are two fundamental ways of data generation: synthetic and real. Both ways have their unique set of advantages and disadvantages. In this section, we discuss them and justify using each technique to generate our novel datasets Inv3D and Inv3DReal. See Figure 4.3 for comparing both techniques.

The first technique, synthetic data generation, attempts to mimic real-world data as closely as possible through simulations, models, and rendering techniques. Using this method, synthetic warped documents could be generated by creating fake but realistic appearing documents before projecting them to 3D meshes. Since this process can be fully automated and does not require a physical setup, it is easy to generate a large-scale dataset. In addition, synthetic data generation allows computation of full ground truth annotations, particularly all pixel-wise annotations introduced in Section 4.1. Crucially, the synthetic data samples only closely resemble the real domain but are not captured in real scenarios. Therefore, the synthetic data does not contain any information from real personas. This fact is important for sharing the dataset with the research community, as there are no privacy-related concerns. While synthetic data generation has many positive aspects, there is also a considerable downfall. There is always a difference between the simulated and the real-world domain when simulating data since the simulations are not perfect. This difference is also called "Sim-To-Real Gap" and could potentially reduce the utility of models trained on simulated data in the real world. That gap may be represented by non-matching data distributions between both domains, for instance, overly light or dark renderings or rendering artifacts in the rendered image that would not occur in a real-world setting.

The second technique collects the data from existing use cases in the real world. For our task of generating a dataset of warped documents, this technique requires collecting real invoices from companies and capturing them twice: once with a scanner and once with a smartphone. While this method might yield the most realistic data, many disadvantages exist. First, there are no pixel-wise annotations, as it is unclear which pixel in the warped image matches which pixel in the flatbed image. Manually annotating this would require extreme effort since an image has vast amounts of pixels. It is often infeasible for a human to annotate this. The human would need to align each pixel in the warped image with the corresponding pixel in the flatbed image, which is particularly difficult for the white areas in the document. Secondly, in contrast to the simulated data, real invoices from existing companies or other personas usually contain private data subject to the respective country's data regulations. This could prevent a dataset from being published.

Thirdly, and most importantly, this data generation process is not scalable for creating a large-scale dataset, as it is not fully automated and relies on costly human labor. Therefore, only a small dataset can be generated using this technique.

Since both techniques have their own set of advantages and disadvantages, we decided to employ both for different use cases. Our new training dataset, Inv3D, must be large-scale and have full ground-truth annotations. Therefore, we chose to generate our training data synthetically. For the evaluation dataset Inv3DReal, we combined both techniques. We could not use real invoices since we aimed to provide the dataset to the research community. That is why we decided to use synthetic techniques to generate the flatbed invoices. In order to generate a realistic evaluation dataset, we decided to employ the second technique for warping the invoices, i.e., printing, deforming, and finally capturing them by hand. This ensures that there are no rendering artifacts in the samples. The warping method inherently does not provide pixel-wise ground truth annotations. However, this lack of pixel-wise ground truth annotations is acceptable since the evaluation methods selected do not necessarily require a true backward mapping $\hat{\mathbf{B}}$.

5 Inv3D: Synthetic training data

This chapter describes our methodology for generating our novel training dataset Inv3D. In order to create a large-scale dataset with full ground truth annotations and reference templates, we employ synthetic data generation approaches.

Our approach is split into three stages: (1) base template generation, (2) instance generation, and (3) instance warping. In the first stage - base template generation - we search for publicly available invoice templates on the web and prepare them for the subsequent stages. The instance generation takes the prepared base templates, generates random content in random formats, and renders a flatbed invoice image. Lastly, we project the flatbed invoice images to 3D meshes. For a complete overview of the data generation pipeline, see Figure 5.1.

This chapter is based on the following publication:

Publication

Felix Hertlein et al. “Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping”. In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: `10.1007/S10032-023-00434-X`

In the following, we describe each stage in detail.

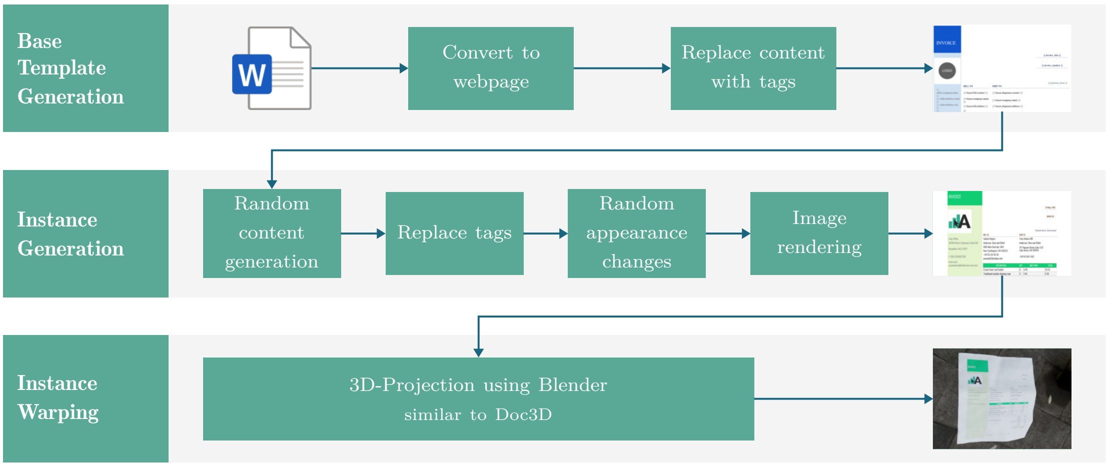

Base Template Generation
Convert to webpage
Replace content with tags
Instance Generation
Random content generation
Replace tags
Random appearance changes
Image rendering
Instance Warping
3D-Projection using Blender similar to Doc3D

Figure 5.1: Overview of the Inv3D dataset generation pipeline

5.1 Base Template Generation

In order to create convincingly realistic invoices, we collected publicly available invoice templates for entrepreneurs in standard text processing formats. By converting them to HTML documents, we can manipulate given formats and contents through simple text modifications while preserving their overall layout. We replaced all exemplary content provided by the input document with machine-readable tags such as `{{ seller.company.name }}`. Thus, our system can automatically insert the correct content at the correct position within the invoice web page as intended by the invoice template creators. In total, we collected and prepared 100 different invoices, which form the basis for the subsequent stages.

5.2 Instance Generation

Creating a realistic invoice instance is the first step in creating a dataset sample. We generate random content from an invoice web template and apply random appearance changes before rendering the invoice instance files.

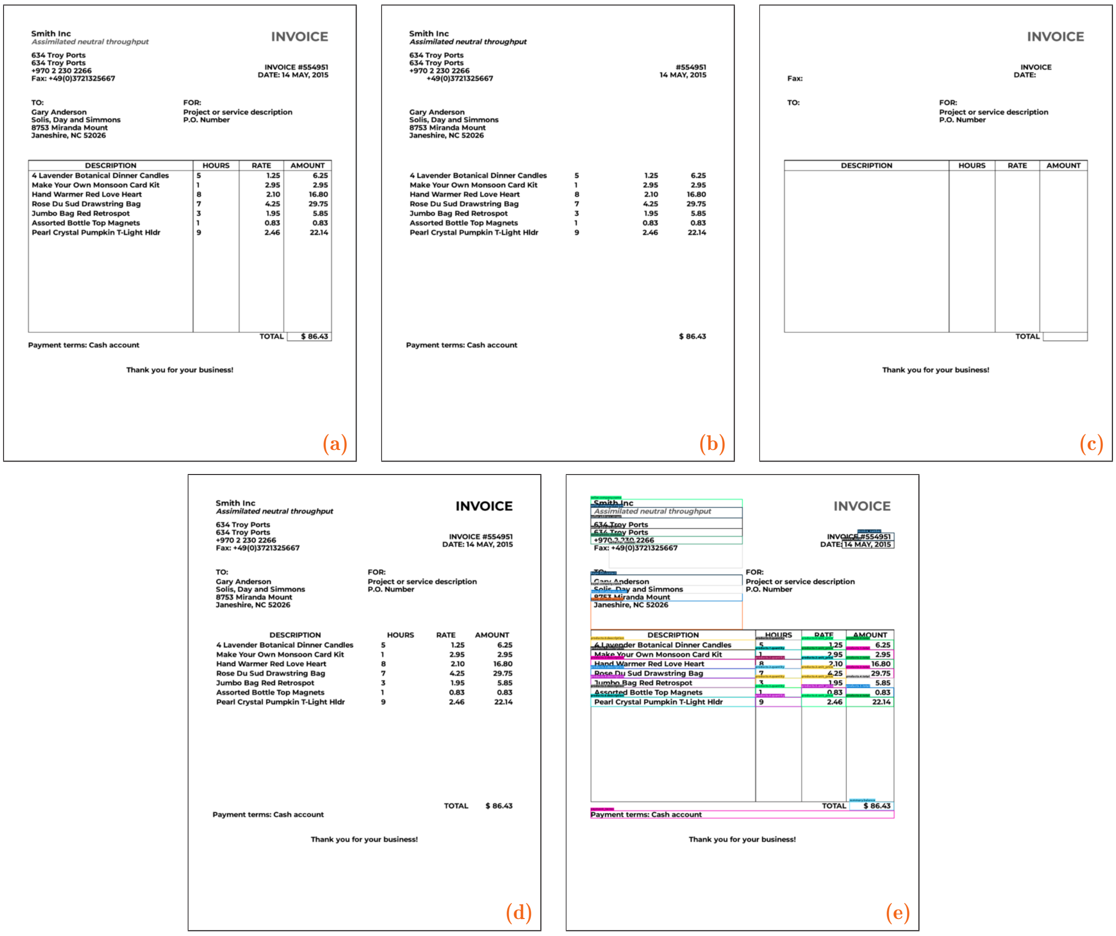
Random content generation. We randomly create fake sales orders and personas that resemble real invoices as closely as possible using existing libraries[1] and the E-Commerce Kaggle dataset [10]. To achieve high realism, we retain the data coherency during the generation process and fit the data to the layout constraints imposed by the web page template, i.e., the number of rows available. Additionally, we generate random representations of the data to increase its variance, e.g., different date formats. Since we provide the generated content in a structured manner, its text representation and position in the document, our dataset can be used for the information extraction task.

Random appearance changes. We increase the visual variance by applying random modifications to the invoice web templates, thus reducing the potential for overfitting. We employ color and font substitutions and random font size scaling. Furthermore, if present, we replaced the logo of the given invoice document with a random logo image from the Large Logo Dataset [74] and altered the document margin.

Rendering. To create a fake invoice sample, the random content is filled into the randomly modified web templates and rendered as an image in A4 format. Furthermore, we create three auxiliary images using JavaScript and CSS manipulations: information delta, template, and text mask. The information delta depicts all randomly generated texts. The template image shows everything except the information delta, hence the overall structure of the given document and static text. The third and last auxiliary images contain all the text in the invoice document. See Figure 5.2 (a) to (d).

Additionally, we provide two types of ground truth information: the true word list and relevant image areas. The latter describes which information is expected to be at which position within the document. Figure 5.2e visualizes the relevant areas. These ground truth annotations are relevant for tasks other than image dewarping, such as information extraction and document understanding.

[1] https://faker.readthedocs.io/en/master/



Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
Fax: +49(0)3721325667

INVOICE
INVOICE #554951
DATE: 14 MAY, 2015

TO:
Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
		TOTAL	$ 86.43

Payment terms: Cash account

Thank you for your business!

(a)

Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
+49(0)3721325667

#554951
14 MAY, 2015

Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
			$ 86.43

Payment terms: Cash account

(b)

INVOICE
Fax:
INVOICE
DATE:

TO:

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
		TOTAL	

Thank you for your business!

(c)

Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
Fax: +49(0)3721325667

INVOICE
INVOICE #554951
DATE: 14 MAY, 2015

TO:
Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
		TOTAL	$ 86.43

Payment terms: Cash account

Thank you for your business!

(d)

Smith Inc
Assimilated neutral throughput
634 Troy Ports
634 Troy Ports
+970 2 230 2266
Fax: +49(0)3721325667

INVOICE
INVOICE #554951
DATE: 14 MAY, 2015

TO:
Gary Anderson
Solis, Day and Simmons
8753 Miranda Mount
Janeshire, NC 52026

FOR:
Project or service description
P.O. Number

DESCRIPTION	HOURS	RATE	AMOUNT
4 Lavender Botanical Dinner Candles	5	1.25	6.25
Make Your Own Monsoon Card Kit	1	2.95	2.95
Hand Warmer Red Love Heart	8	2.10	16.80
Rose Du Sud Drawstring Bag	7	4.25	29.75
Jumbo Bag Red Retrospot	3	1.95	5.85
Assorted Bottle Top Magnets	1	0.83	0.83
Pearl Crystal Pumpkin T-Light Hldr	9	2.46	22.14
		TOTAL	$ 86.43

Payment terms: Cash account

Thank you for your business!

(e)

Figure 5.2: A synthetically generated invoice sample. From a to e: full document, information delta, template, text mask, and relevant area visualization.

5.3 Instance Warping

The next step in the dataset generation process is mapping flat invoices to deformed sheets of paper in 3D and creating 2D renderings (see Figure 5.3a). We project our invoices to the meshes from Doc3D [16] using Blender[2]. For the environment maps, we used the Laval Indoor HDR dataset [25]. In contrast to the Doc3D dataset, we rendered our samples with a considerably higher resolution to represent the real-world scenario better. We chose 1600x1600 instead of the 448x448 pixels used by Doc3D.

This procedure generates the rendering itself (Figure 5.3a) and various additional ground truth maps, namely an albedo map (Figure 5.3b), depth map (Figure 5.3c), normal map (Figure 5.3d), reconstruction map (Figure 5.3e), UV map (Figure 5.3f) and, world coordinate map (Figure 5.3g).

In addition to the previously mentioned ground truth maps, we create and provide four more maps to facilitate our dataset's usage and reduce the need for computation-intensive calculations. The first is a high-resolution backward map (BM) defined in Chapter 2. From Blender, we can directly extract the forward map, also known as the UV map, in the computer graphics community. Since

[2] https://www.blender.org/

the UV map generated by Blender is incomplete in the border region of the texture, we used nearest-neighbor extrapolation to fill the missing pixels in the backward map. The other auxiliary maps are relevant for CREASE [62], namely per-pixel orientation angles (see Figure 5.3h), curvature estimations (see Figure 5.3i), and text masks (see Figure 5.3j).

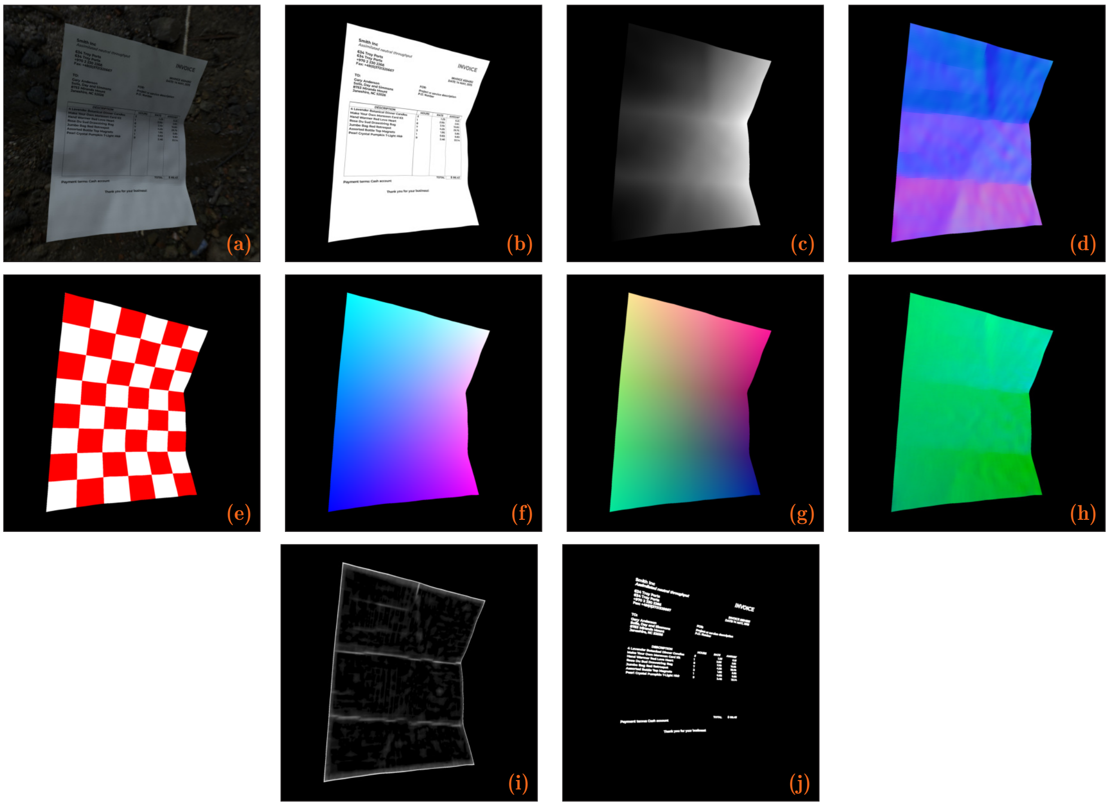


Figure 5.3: The 3D warped document and its nine supervision signals. From (a) to (j): warped document, albedo map, depth map, normal map, reconstruction map, UV map, WC map, warped angles, curvature map, and text mask.

5.4 Design Decisions

Since we use more than one external resource (invoice documents, logos, environments, object meshes, and fonts), we must define the dataset train, validate, and test split before creating the dataset. We split all resources according to our split ratios (66.6% train, 16.7% validation, 16.7% test) and assigned the resource split to each split, respectively. This way, we prevent information leakage between these three splits by using resources in more than one split at a time. Note that the fonts were split on font level instead of style level. Furthermore, we split the meshes according to their generation. Most meshes were created by modifying a recorded parent mesh [16]. We split all meshes by their parent mesh to keep a clear separation between splits.

5.5 Summary and Discussion

In this chapter, we presented our methodology for generating the Inv3D dataset. This dataset fulfills all requirements on a training dataset as listed in the previous chapter Section 4.1. Therefore, this chapter answers the research question one:

RQ1: Data Acquisition

How can we generate a **large-scale, high-resolution dataset** of document images with **ground truth annotations** for geometric dewarping and illumination correction with corresponding **reference templates**?

With its 25,000 samples, our new dataset Inv3D serves as a large-scale dataset of document images $\mathbf{W}$ and its flatbed correspondence $\hat{\mathbf{I}}$. It includes complete ground truth annotations, precisely the true backward map $\hat{\mathbf{B}}$, the true forward map $\hat{\mathbf{F}}$, and additional supervision signals. Lastly and most importantly, each sample comes with a reference template $\mathbf{T}$, which can be used for template-based geometric dewarping and illumination correction.

We published the dataset at: `https://publikationen.bibliothek.kit.edu/1000161884`.

6

Inv3DReal: Real-world evaluation data

In the previous chapter, we introduced the training dataset Inv3D. While this dataset is suitable for training models, we need an independent dataset for evaluation. Crucial for the evaluation dataset is that it is as realistic as possible. In this chapter, we describe the methodology of generating an evaluation dataset called Inv3DReal. Contrary to the training dataset, Inv3DReal is closer to real-world conditions, as it consists of pictures taken by a smartphone camera. We generate the flatbed invoices similarly to the Inv3D invoices, but instead of rendering them, we print them out, apply deformations, and take pictures of them. This way, we avoid any 3D rendering artifacts and can evaluate the performance of dewarping models under realistic conditions.

This chapter is based on the following publication:

Publication

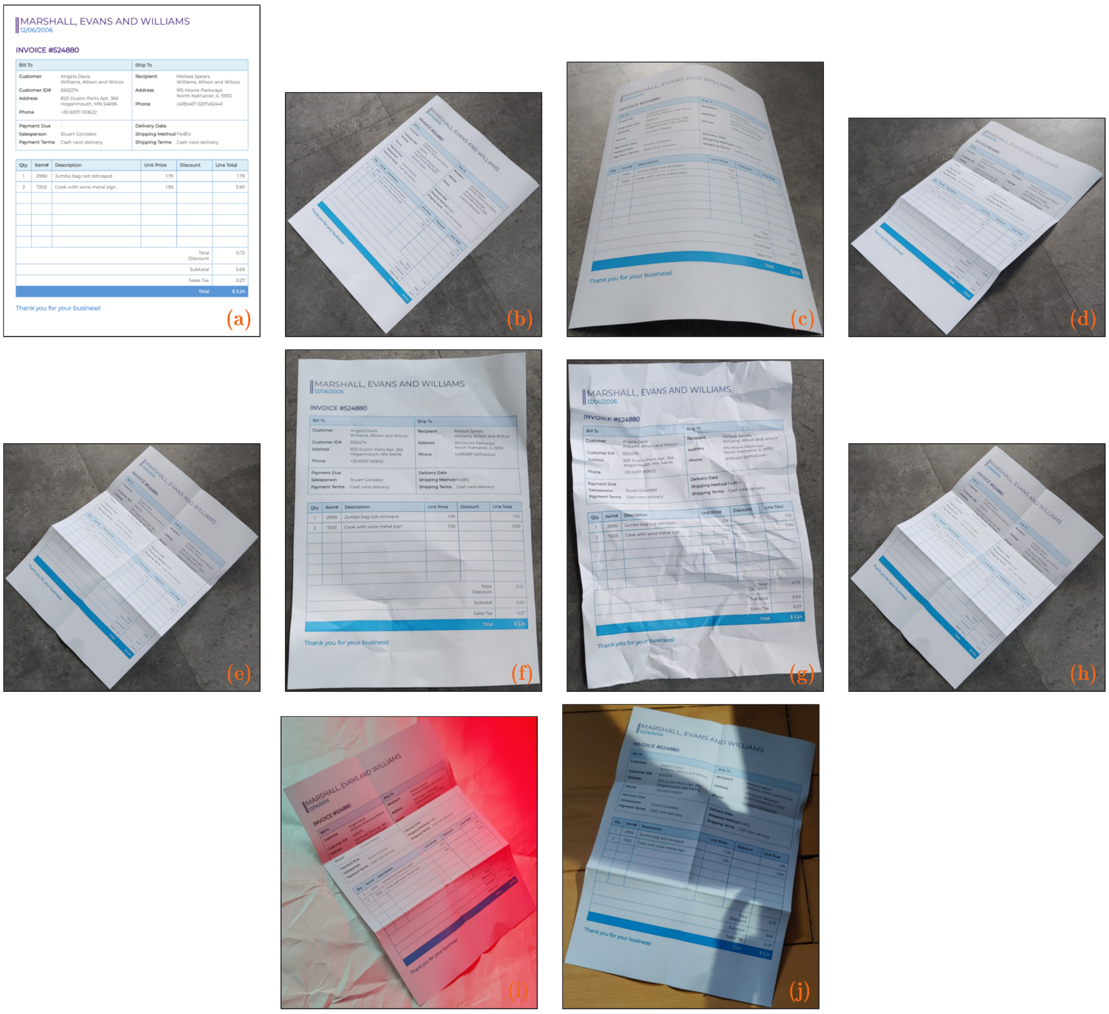
Felix Hertlein et al. "Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping". In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: `10.1007/S10032-023-00434-X`

6.1 Methodology

Inv3DReal consists of 360 pictures displaying printed and altered invoices taken by a smartphone camera under different lighting conditions and backgrounds. We randomly selected 20 samples from the synthetic test dataset as the basis and applied six different deformations (perspective, curled, fewfold, multifold, crumples easy, crumples hard) inspired by Das et al. [16], as well as three different settings (bright, colored, shadow). We provide examples in Figure 6.1. The bright setting (Figure 6.1h) displays the documents on a gray background with daylight incidence. The second setting (Figure 6.1i) displays the document on a white background sheet with RGB lighting. Lastly, we defined the shadow setting (Figure 6.1j) as a document in front of a wooden surface with multiple shadows falling onto the document.

6.2 Summary and Discussion

We complete the data generation task by creating an evaluation dataset called Inv3DReal. Since the dataset was created by taking pictures of warped documents, it does not contain any 3D rendering artifacts and, thus, is closer to the real use case. Because the documents showing Inv3DReal are synthetically generated invoices, there are no data privacy concerns that could hinder a publication.



MARSHALL, EVANS AND WILLIAMS
12/06/2006

INVOICE #524880

Bill To		Ship To	
Customer	Angela Davis Williams, Allison and Wilcox	Recipient	Melissa Spears Williams, Allison and Wilcox
Customer ID#	5502274	Address	915 Moore Parkways North Nathaniel, IL 19312
Address	8121 Dustin Ports Apt. 366 Hoganmouth, MN 54696	Phone	(438)407-3207x62441
Phone	+30 6937 010622		
Payment Due	-	Delivery Date	
Salesperson	Stuart Gonzalez	Shipping Method	FedEx
Payment Terms	Cash next delivery	Shipping Terms	Cash next delivery

Qty.	Item#	Description	Unit Price	Discount	Line Total
1	2990	Jumbo bag red retrospot	1.79		1.79
2	7203	Cook with wine metal sign	1.95		3.90
				Total Discount	0.72
				Subtotal	5.69
				Sales Tax	0.27
				Total	$ 5.24

Thank you for your business!

Figure 6.1: Random sample of the real-world dataset used for benchmarking the Inv3D dataset. First image: flatbed invoice. Images (b) to (g) show the deformations: perspective, curled, fewfold, multifold, crumples easy, and crumples hard. Images (h) to (j) showcase the lighting settings: bright, colored, and shadow.

In contrast to the training dataset Inv3D, there are no ground-truth backward maps $\hat{\mathbf{B}}$ and forward maps $\hat{\mathbf{F}}$, as well as other supplementary pixel-wise annotations. The evaluation of the models will be based on the dewarped image $\mathbf{B}(\mathbf{W})$ and the flatbed image $\hat{\mathbf{I}}$, along with the texts detected in both images.

We published Inv3DReal at: `https://publikationen.bibliothek.kit.edu/1000161884`.

Part IV

Geometric Dewarping

7
Problem Formalization

Given the problem that documents within the camera-captured images are not flat, we need to find a way to dewarp them such that the resulting image resembles the flatbed document images. Since the documents can be arbitrarily deformed, we need to find a way to model this deformation.

In this chapter, we formalize the problem of geometric dewarping and introduce the necessary terminology in Section 7.1 and discuss the evaluation of geometric dewarping approaches in Section 7.2. Additionally, we connect the formal problem definition to the research questions defined in Chapter 1.

7.1 Geometric Dewarping

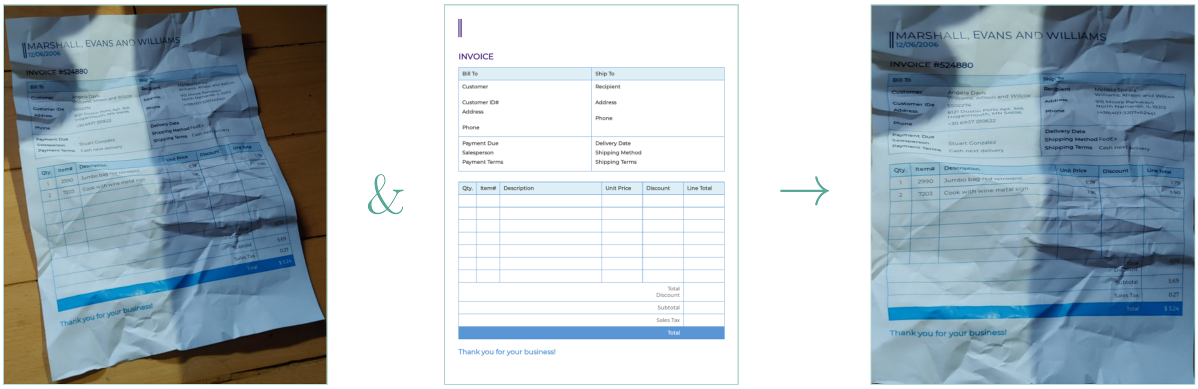
The task of geometric dewarping has two inputs: (1) a RGB image $\mathbf{W} \in \mathbb{R}^{\mathrm{h}_0 \times \mathrm{w}_0 \times 3}$ showing a warped document, and (2) a reference template image $\mathbf{T} \in \mathbb{R}^{\mathrm{h}_1 \times \mathrm{w}_1 \times 3}$ showing the general structure of the flatbed document. Given those images, the task is to find a mapping of pixel locations from the warped image $\mathbf{W}$ to a new image $\mathbf{D} \in \mathbb{R}^{\mathrm{h}_1 \times \mathrm{w}_1 \times 3}$, such that the pixels in $\mathbf{D}$ correspond to the pixels in the flatbed document image $\hat{\mathbf{I}}$ for identical positions. We denote a mapping of pixel locations $\mathbf{B} \in [0,1]^{\mathrm{h}_\mathrm{b} \times \mathrm{w}_\mathrm{b} \times 2}$, thus, the dewarped image can be generated by applying the pixel mapping to the dewarped image $\hat{\mathbf{I}} = \mathbf{B}(\mathbf{W})$. For a visualization of the backward map, see Figure 2.6.

Formally, we can describe the geometric dewarping task as follows:

$$
\begin{gathered}
\mathbb{R}^{\mathrm{h}_0 \times \mathrm{w}_0 \times 3} \times \mathbb{R}^{\mathrm{h}_1 \times \mathrm{w}_1 \times 3} \to [0,1]^{\mathrm{h}_\mathrm{b} \times \mathrm{w}_\mathrm{b} \times 2} \\
(\mathbf{W}, \mathbf{T}) \mapsto \mathbf{B}
\end{gathered}
$$

Note that the task of geometric dewarping only attempts to relocate all pixels from the warped image to the dewarped image but does not change the color value of each individual pixel. Therefore, the optimal result of the geometric dewarping is an image that resembles the flatbed image $\hat{\mathbf{I}}$ closely but is not identical due to the illumination changes by the environment and the falsification of colors by cameras. The example in Figure 7.1 shows a warped document and associated template, along with its (nearly) perfectly dewarped document image.

We can now connect the formal problem definition to the research questions defined in Chapter 1. Two questions (RQ2.1 and RQ2.3) focus on geometrically dewarping document images, which are defined as follows:



Warped document $\mathbf{W}$ Template $\mathbf{T}$ DewarpedImage $\mathbf{D}$

Figure 7.1: Example of a nearly perfect geometric dewarping.

RQ2.1: Implicit Geometric Dewarping

How can we dewarp document images using a **reference template** to improve the quality of the document images?

and

RQ2.3: Explicit Geometric Dewarping

How can we dewarp document images with a reference template by **explicitly leveraging the template information** to improve the quality of the document images?

Both research questions aim to find the backward map $\mathbf{B}$ given the warped image $\mathbf{W}$ and the reference template image $\mathbf{T}$, such that the dewarped image $\mathbf{B}(\mathbf{W})$ resembles the perfect flatbed image $\hat{\mathbf{I}}$. The difference between the two research questions lies in the method employed to perform the dewarping. The former question aims to find a learning-based method which allows the model to leverage the information contained in the reference template image $\mathbf{T}$ freely. The latter question, however, focuses on an explicit approach, i.e., a multi-stage model with explicit intermediate representations. Intermediate stages can be, for example, the extraction of the document contour or the estimation of structural or text lines. This approach is better interpretable, as the model generates humanly understandable intermediate results.

7.2 Evaluation of Geometric Dewarping

While there are a few established metrics for geometric dewarping, they all come with at least one problem, making them unsuitable for measuring the dewarping performance of a given model. Since we cannot use a ground truth backward map $\hat{\mathbf{B}}$ for measuring the model performance, as there is no such information for real-world data samples, we base the evaluation on the comparison of the (partially) dewarped image $\mathbf{I_{dwp}}$ and the flatbed image $\hat{\mathbf{I}}$. The task of finding a suitable metric can, therefore be formalized as the search for a function Ψ with

$$\begin{aligned}\Psi\colon\ \mathbb{R}^{\mathrm{h}_1\times\mathrm{w}_1\times 3}\times\mathbb{R}^{\mathrm{h}_1\times\mathrm{w}_1\times 3} &\to [0,1]\\ (\mathbf{I}_{\mathbf{dwp}},\hat{\mathbf{I}}) &\mapsto \psi\end{aligned}$$

where ψ denotes the real-valued error in geometric dewarping.

In research question RQ2.2, we aim to find such metric for measuring the performance of geometric dewarping approaches, which does overcome the limitations of existing metrics. The research question is defined as follows:

RQ2.2: Dewarping Metric

How can we **evaluate the quality** of the geometric dewarping process with regard to text readability and positional awareness?

8 Implicit Reference Template Leverage

In this chapter, we aim to solve RQ2.1, i.e., the geometric dewarping of document images with additional reference templates using an implicit method. Implicit in this context means we provide the additional information of reference templates to a deep learning model and let the model determine how to leverage the novel information as effectively as possible. We build our approach upon the existing dewarping model GeoTr by Feng et al. [21]. Essentially, they extract visual features from the warped image and employ an attention-based model to combine the information contained in all image regions. We extend this model by extracting visual features from the template image and providing them to the attention model. Thus, the enlarged model can attend in-between warped image and template features and consequently integrate the information contained in the template image.

This chapter is based on the following publication:

Publication

Felix Hertlein et al. "Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping". In: *International Journal on Document Analysis and Recognition* 26.3 [2023], pp. 175–186. DOI: 10.1007/S10032-023-00434-X

We start by introducing the approach in Section 8.1, before explaining the evaluation process in Section 8.2. In Section 8.3, we present the results of our approach and discuss them in Section 8.4.

8.1 Approach

This section presents our novel approach for image dewarping by leveraging structural templates at training and inference time. We extend the transformer-based state-of-the-art model GeoTr introduced by Feng et al. [21] to incorporate the a priori known structural information represented through the invoice templates. In the following, we refer to our new model as GeoTrTemplate and its extension as GeoTrTemplateLarge. See Figure 8.1 for a schematic. Our model receives the warped image $\mathbf{W} \in \mathbb{R}^{h_0 \times w_0 \times 3}$ and the template image $\mathbf{T} \in \mathbb{R}^{h_1 \times w_1 \times 3}$. Both inputs are scaled to a fixed resolution of 288 × 288 for GeoTrTemplate and 600 × 600 for GeoTrTemplateLarge before applying a geometric head H individually to each image. The head H creates deep image representations with 36×36 positional features in a 128-dimensional space. For GeoTrTemplate, we employ the geometric head proposed by Feng et al. [21]. We define the geometric head for our large model as a slice of the EfficientNet B7 noisy student model [94], namely the first four convolutional blocks followed by a custom convolutional layer. The features of the warped image $H(\mathbf{W})$ and the template $H(\mathbf{T})$ are concatenated, forming a combined input representation $R \in \mathbb{R}^{36 \times 36 \times 256}$. We then apply the Transformer Encoder and Decoder from Feng et al. [21] and their Geometric Tail module to upsample the resulting backward map. For details regarding the modules employed, see the original paper. The output is a backward map $\mathbf{B} \in [0, 1]^{288\times288\times2}$. Our loss function is defined as the L1-norm between the output backward map $\mathbf{B}$ and the true backward map $\hat{\mathbf{B}}$.

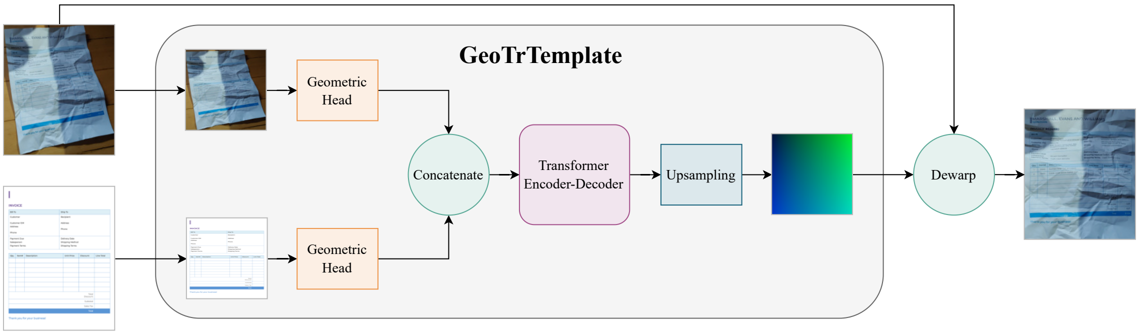


Figure 8.1: Information flow of our model GeoTrTemplate. The architecture extends the GeoTr model by Feng et al. [21].

8.2 Evaluation

In this section, we present details on the evaluation. First, we list and explain the employed metrics. Thereafter, we discuss the baselines, and finally, we give details on the used hyperparameters.

8.2.1 Metrics

To evaluate our approach, we employ the established metrics in the field of document image dewarping, as introduced in Section 2.3. Namely, we use the visual metrics *MS-SSIM* [90], *LPIPS* [104], and *LD* [97], as well as the text-based metrics *ED* [43] and *CER* [1].

8.2.2 Baseline selection

We compare our results to the baselines DewarpNet [18], excluding the refinement network, and GeoTr [21]. Document dewarping can be decomposed into two subtasks, geometric dewarping and illumination correction. The prior remaps all pixel locations whereas the latter alters the per pixel colors to remove shading and environmental light effects. As our model focuses exclusively on geometric dewarping, we selected our baselines to use geometric dewarping only, to make a fair comparison. The usage of an illumination correction model is decoupled from the geometric dewarping, thus can be appended to all baselines and our model. Since the refinement network of DewarpNet [16] and IllTr [21] are illumination correction networks, we argue that omitting these networks is well-founded.

8.2.3 Hyperparameters

We trained all models up to 300 epochs with an early stopping patience of 25 epochs based on the validation mean squared error between $\mathbf{B}$ and $\hat{\mathbf{B}}$. All other hyperparameters depend on the model type. We used the parametrization published by the original authors to reproduce their results as closely as possible. For GeoTrTemplate, we used a batch size of 8, the AdamW optimizer [57] with an initial learning rate of 10^{-3} and the OnceCycleLR scheduler [77] with a maximum learning rate of 10^{-3}. Note that for the training of GeoTr, GeoTrTemplate and GeoTrTemplateLarge we employed gradient clipping to increase the training stability. We clipped the global gradient norm to a value of 1.

DewarpNet [16] and GeoTr [21] employ different background augmentation strategies to boost the model performances. DewarpNet replaces the warped image background with randomly selected textures from the Describable Textures Dataset [13] during training. GeoTr learns a light semantic segmentation network [71] in order to remove the background beforehand. We kept the original backgrounds to enable a fair comparison of both approaches and our models. Furthermore, we augmented all images for training using color jitter with a random change of up to 20 % in brightness, contrast, saturation, and hue, respectively. Note that the input image resolution differs on the individual model due to architectural constraints. In particular, DewarpNet receives images with 128×128 pixels, whereas GeoTr and GeoTrTemplate use images with a resolution of 288×288 pixels and GeoTrTemplateLarge the resolution of 600×600 pixels.

8.3 Results

This section presents and interprets our quantitative and qualitative results, as well as our ablation study on the importance of reference templates.

8.3.1 Quantitative Results

Table 8.1 shows the quantitative results of our approach and the baseline methods evaluated on our new dataset Inv3DReal. We include the identity backward map for reference, thus creating a lower bound for the scores. As expected, all approaches outperform the identity baseline in all metrics by a large margin. Our experiments show that GeoTr is superior to DewarpNet without the refinement network in all metrics and training datasets. When comparing the two models with respect to the used training dataset, we conclude that training on the Inv3D dataset slightly improves the evaluation results compared to training on Doc3D. This effect is expected since Inv3D is more similar to Inv3DReal than Doc3D. The best results by a large margin yielded our models GeoTrTemplate and GeoTrTemplateLarge in all metrics, especially for the visual evaluation metrics. The local distortion improves by 23.4 % and 26.1 %, respectively, compared to the runner-up GeoTr trained on Inv3D. These results show the effectiveness of our approach.

We also train the baseline models on Doc3D and evaluate on the established DocUNet benchmark [61]. The experiment shows that our implementation can closely reproduce the reported results. The difference is likely due to differences in the image augmentation methods that were applied in the original paper, e.g., we omit the random background replacement of DewarpNet in our setting. We choose to apply the exact same image augmentations for all approaches to allow for a fair comparison between them. See Table 8.2 for a direct comparison of our results with the reported numbers.

When comparing the absolute number of the DocUNet benchmark and our Inv3DReal benchmark, we observe that the DocUNet evaluations are closer to the optimum for most metrics, which

Model	Train Dataset	↑*MS-SSIM*	↓*LD*	↓*LPIPS*	↓*ED*	↓*CER*
Identity	-	0.44 (0.10)	36.75 (13.80)	0.60 (0.10)	533 (169)	0.83 (0.21)
DewarpNet (w/o ref)	Doc3D	0.56 (0.11)	26.10 (11.50)	0.42 (0.12)	384 (182)	0.59 (0.25)
DewarpNet (w/o ref)	Inv3D	0.55 (0.11)	25.33 (10.56)	0.43 (0.12)	387 (176)	0.60 (0.24)
GeoTr	Doc3D	0.56 (0.11)	23.27 (10.35)	0.40 (0.12)	357 (185)	0.55 (0.26)
GeoTr	Inv3D	0.56 (0.11)	22.81 (9.98)	0.41 (0.12)	365 (181)	0.57 (0.25)
GeoTrTemplate (ours)	Inv3D	0.64 (0.12)	17.46 (10.62)	0.32 (0.13)	349 (185)	0.54 (0.26)
GeoTrTemplateLarge (ours)	Inv3D	**0.65 (0.12)**	**16.86 (10.46)**	**0.31 (0.13)**	**327 (184)**	**0.51 (0.26)**

Table 8.1: Quantitative evaluation of our new model GeoTrTemplate on Inv3DReal with respect to the training dataset. Values in brackets denote standard deviations.

indicates that our benchmark is more challenging to solve for given approaches. Note that the edit distance on DocUNet is higher on Inv3DReal because DocUNet images contain more text on a large margin. Therefore, the *OCR* engine yields long texts, leading to a high absolute number of insertions, deletions, and replacements for DocUNet.

Model	↑*MS-SSIM*	↓*LD*	↓*LPIPS*	↓*ED*	↓*CER*
DewarpNet (w/o ref)	0.45 (0.12)	10.15 (7.49)	0.32 (0.11)	1180 (1318)	0.30 (0.22)
DewarpNet (w/o ref, orig)	0.47 (-)	8.98 (-)	-	1289 (-)	0.31 (0.25)
GeoTr	0.45 (0.12)	8.65 (6.11)	0.31 (0.10)	892 (1254)	0.23 (0.24)
GeoTr (orig)	-	8.38 (-)	-	935 (-)	0.31 (-)

Table 8.2: Comparison of our implementation with the numbers reported by the original papers trained on Doc3D and evaluated on the DocUNet benchmark [61]. Values in brackets denote standard deviations.

To better understand the characteristics of each approach, we provide an in-depth evaluation of our model GeoTrTemplate based on Inv3DReal. We average the evaluation data per deformation class and per lighting setting. The results are shown in Table 8.3. According to most metrics, the *curled* deformation appears to be the most straightforward task, whereas the *crumples hard* represents the most challenging class to dewarp. Interestingly, the *LD* does not agree with other metrics as stronger deformations lead to better results with regard to the *LD* metric. When we compare the three different environment settings, it appears that *bright* is the easiest, whereas *shadow* is the hardest according to most metrics. Similar to the deformation evaluation, we observe the inverse order according to the local distortion.

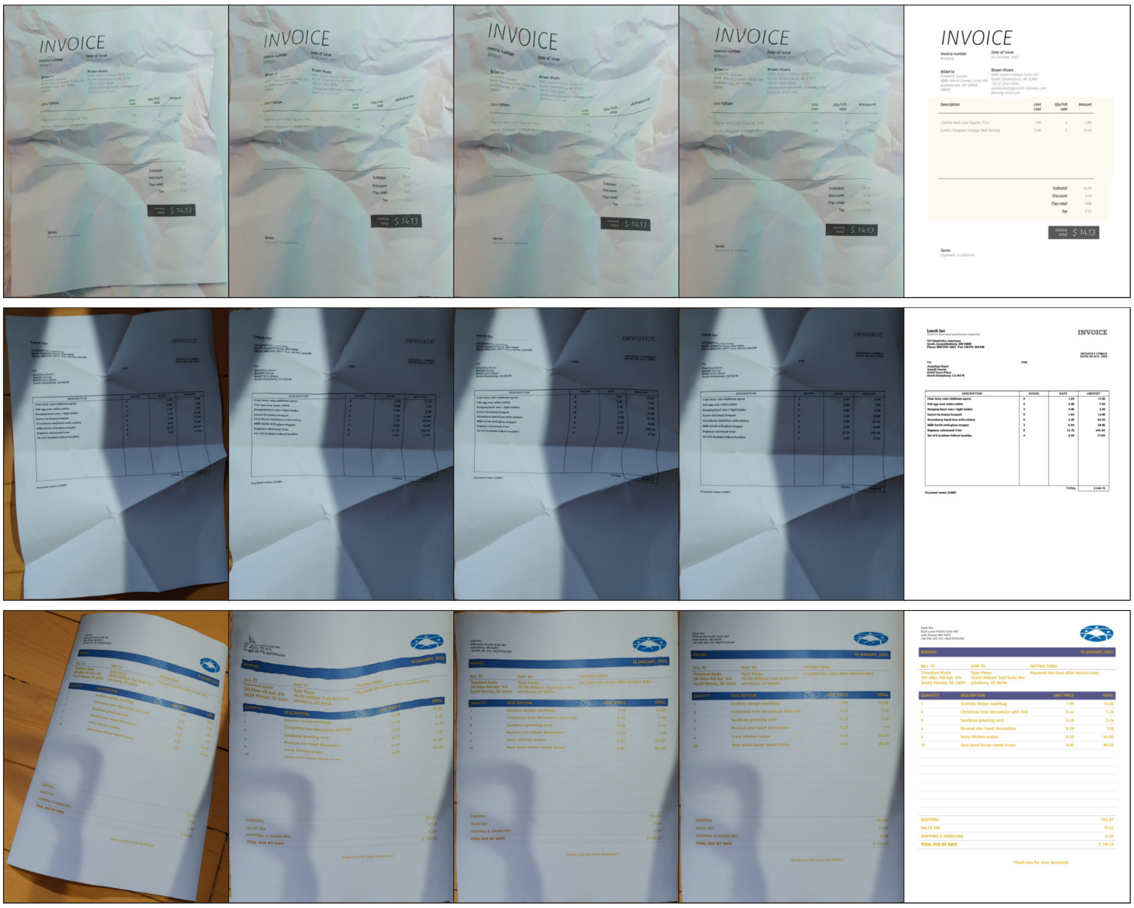
For a quantitative evaluation of DewarpNet [16], and GeoTr [21] on Inv3DReal with regard to the different modifications, see Table 8.3.

8.3.2 Qualitative Results

Figure 8.2 displays three selected samples from Inv3DReal, as well as the dewarping results from DewarpNet [16], GeoTr [21], our model GeoTrTemplateLarge, and the original invoice document. When comparing DewarpNet and GeoTr, we observe that both models have problems correcting

Modification	↑*MS-SSIM*	↓*LD*	↓*LPIPS*	↓*ED*	↓*CER*
perspective	0.69 (0.12)	20.92 (12.15)	0.27 (0.12)	316 (191)	0.49 (0.26)
curled	**0.70 (0.09)**	19.74 (11.75)	**0.26 (0.11)**	**301 (173)**	**0.47 (0.25)**
fewfold	0.66 (0.10)	18.70 (9.67)	0.30 (0.11)	335 (184)	0.52 (0.26)
multifold	0.61 (0.11)	18.84 (11.00)	0.34 (0.12)	367 (165)	0.57 (0.23)
crumples easy	0.64 (0.11)	**12.75 (7.33)**	0.31 (0.11)	357 (206)	0.56 (0.31)
crumples hard	0.52 (0.10)	13.82 (8.65)	0.46 (0.11)	417 (175)	0.64 (0.21)
bright	**0.69 (0.11)**	17.74 (10.39)	**0.25 (0.11)**	**272 (177)**	**0.42 (0.25)**
color	0.66 (0.10)	19.41 (11.20)	0.37 (0.14)	329 (185)	0.51 (0.25)
shadow	0.56 (0.10)	**15.24 (9.88)**	0.35 (0.10)	446 (149)	0.69 (0.20)

Table 8.3: Detailed evaluation of GeoTrTemplate on our new benchmark Inv3DReal split by modification category. Values in brackets denote standard deviations.



(a) Input (b) DewarpNet (w/o ref) (c) GeoTr (d) GeoTrTemplate-Large (ours) (e) Original

Figure 8.2: Qualitative evaluation of DewarpNet [16] without refinement network, GeoTr [21], and our model based on selected samples of Inv3DReal.

the line straightness. A comparison of GeoTrTemplateLarge and GeoTr indicates better global positioning and straighter lines when using the template.

8.3.3 Ablation Study

We conducted an ablation study to measure the influence of different types of structural information on the model performance. For our study, we altered the templates in three different ways and trained our model from scratch using the altered template as input. The modifications are as follows:

White Template. The template image is entirely white and thus contains no additional information over the warped input image.

Text Only. The template image for this ablation contains all texts visible on the original template image (see Figure 5.2c), but no other structures such as lines or images. Note that all texts were converted to black to be visible after removing the background colors. The final template ablation is a black and white image.

Structure Only. We removed all textual information from the original template image (see Figure 5.2c), such that only the structural information remains.

The evaluation results on the Inv3DReal dataset are shown in Table 8.4. The white template ablation performs the poorest of all ablations. The absolute metrics values are comparable to the GeoTr model trained on Inv3D as shown in Table 8.1. This performance is as expected since both experiments receive the same information as input and have a fairly similar network structure. According to the visual metrics, the structure-only ablation and the text-only ablation boost the performance of our model in each case by a few points, but the combination of both yields the best overall performance. This finding indicates a correlation between the model performance and the amount of a priori known information about the target structure. For the text metrics, while there is no significant change of *ED* and *CER* values, the structure-only ablation performs best by a small margin. Overall, we see a more significant improvement in the visual metrics compared to the text-based metrics, which indicates that adding structural information primarily helps to improve global positioning rather than fine-grained details.

To further investigate the template's influence on the performance, we conducted a second ablation test. For this, we trained the GeoTrTemplate model using the Inv3D dataset and selected random templates during inference. The results are shown in Table 8.4. The comparison of choosing a random vs. the correct template shows that the correct template selection is crucial for performance. Indeed, falsely selecting a template results in degraded performance compared to not providing additional information (white template).

Ablation	↑*MS-SSIM*	↓*LD*	↓*LPIPS*	↓*ED*	↓*CER*
White Template	0.56 (0.11)	22.50 (10.34)	0.40 (0.12)	345 (183)	0.54 (0.26)
Structure only	0.61 (0.12)	19.76 (10.09)	0.34 (0.13)	**341 (186)**	**0.53 (0.26)**
Text only	0.60 (0.11)	18.71 (10.32)	0.35 (0.13)	347 (186)	0.54 (0.26)
Full	**0.64 (0.12)**	**17.46 (10.62)**	**0.32 (0.13)**	349 (185)	0.54 (0.26)
Random template	0.54 (0.11)	29.42 (11.65)	0.44 (0.12)	355 (178)	0.55 (0.25)

Table 8.4: Ablation study of GeoTrTemplate on Inv3DReal by gradually removing template information for training and inference. We separately evaluated the importance of correct template choice by randomly selecting templates only during inference. Values in brackets denote standard deviations.

8.4 Summary and Discussion

In this chapter, we presented two novel models, GeoTrTemplate and GeoTrTemplateLarge, which leverage the additional information given by reference templates. They achieve this implicitly using the attention mechanism. We conducted a detailed evaluation study to compare our new models with the state-of-the-art approaches DewarpNet [16] and GeoTr [21]. Our empirical analysis showed that both outperform the baseline methods significantly, in particular, the GeoTrTemplateLarge model improves the local distortion of GeoTr by 26.1 %. Nevertheless, the absolute values for text detection show that further research is needed to solve this task robustly and consistently.

In this chapter, we addressed the RQ2.1:

RQ2.1: Implicit Geometric Dewarping

How can we dewarp document images using a **reference template** to improve the quality of the document images?

The research question is answered as we improved the quality of the dewarped document images over the previous state of the art, GeoTr, achieved through the use of reference templates with our GeoTrTemplate model and its larger version. By ablating our model with reduced template information, we show that the information inside the reference templates is crucial for improving the dewarping results.

9 Matched Normalized Character Error Rate

In this chapter, we address the problem of adequately evaluating the performance of document image dewarping. We start by reviewing the current evaluation metrics and their limitations, then introduce our new metric, the *matched normalized Character Error Rate* (*mnCER*), and explain how it addresses the shortcomings of the previous metrics.

This chapter is based on the following publication:

Publication

Felix Hertlein et al. "DocMatcher: Document Image Dewarping via Structural and Textual Line Matching". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2025

We start by defining the problem and the limitations of the current evaluation metrics in Section 9.1. In Section 9.2, we introduce our new metric *mnCER* and explain how it overcomes the limitations of the current metrics. Finally, we summarize and discuss the new metric in Section 9.3.

9.1 Problem

There are five established metrics for document image dewarping: (1) *Multiscale Structural Similarity* (*MS-SSIM*), (2) *Learned Perceptual Image Patch Similarity* (*LPIPS*), (3) *Local Distortion* (*LD*), (4) *Edit Distance* (*ED*), and (5) *Character Edit Rate* (*CER*) [16, 21]. See Section 2.3 for the definitions.

Since the ground truth backward map $\hat{\mathbf{B}}$ is assumed to be unknown, all metrics are based on the comparison of the dewarped image $\mathbf{D} := \mathbf{B}(\mathbf{W})$ and the true flatbed image $\hat{\mathbf{I}}$. The established metrics can be divided into two categories: (1) visual metrics and (2) text-based metrics.

Visual metrics operate directly on these two images on a pixel level. *MS-SSIM* extracts statistical features from the input images, while *LPIPS* extracts deep-learning-based features that can be compared to assess image similarity. For *LD*, the dense optical flow between both images is calculated and combined using the L_2 metric. The assumption is that an effective image dewarping yields a flow with small displacements and, thus, a small overall metric value. All visual metrics have in common that they only focus on the visual features and disregard the text readability of the documents entirely. Since the text in the images contains most of the document information, evaluating the text readability is crucial for a suitable document image enhancement metric. Neglecting text readability within the visual metrics can result in surprising evaluation scores. For example, a dewarping model that positions pixels roughly in the correct regions but shuffles neighboring pixels might still achieve a decent evaluation score, as the visual features largely align. This outcome is counter-intuitive, as such a model produces images with unreadable text due to the absence of fine-grained details required for text interpretation. Therefore, the visual metrics are not suitable as a single metric to judge the dewarping capabilities of a dewarping approach.

In contrast to visual metrics, the text-based metrics compare the extracted text from the dewarped image $t(\mathbf{D})$ and the true flatbed image $t(\hat{\mathbf{I}})$. While the text-based metrics focus on text readability, they are overly sensitive to the positions of the recognized words. Structured documents, such as invoices, have text spread across the 2D document, and for the text-based metrics, these texts are linearized into a 1D sequence. Therefore, even slight changes in the position of the text can lead to a considerable change in the metric value. Additionally, text-based metrics do not capture the absolute positioning of the text in 2D space, which is crucial for the semantic interpretation of the text, especially if reference templates are available. To address these limitations, we introduce a novel metric that we call the *matched normalized Character Error Rate* (*mnCER*).

9.2 Approach

Given two images, the dewarped image $\mathbf{D}$ and the ground truth image $\hat{\mathbf{I}}$, we first extract the words from both images using the *OCR* engine docTR [65]. Each word w is retrieved with its text t_w and its bounding box b_w in normalized image space $[0,1] \times [0,1]$. This yields two sets of words $W_\mathbf{D}$ and $W_{\hat{\mathbf{I}}}$ for the dewarped and the ground truth image, respectively.

We then match the words from $W_\mathbf{D}$ to $W_{\hat{\mathbf{I}}}$ based on the spatial locality of each word in $W_{FlatImage}$ and each word in $W_\mathbf{D}$. We define the locality measurement as follows:

$$\text{locality}(b_u, b_i) = \begin{cases} IoU(b_u, b_i) & \text{if } IoU(b_u, b_i) > 0 \\ -d(b_u, b_i) & \text{if } IoU(b_u, b_i) = 0 \end{cases} \tag{9.1}$$

where b_u and b_i are the bounding boxes of the words w_u and w_i in $W_\mathbf{D}$ and $W_{\hat{\mathbf{I}}}$, respectively, and $d(b_u, b_i)$ is the minimal euclidean distance between the bounding boxes. The value range of the locality measurement is, therefore, $[-\sqrt{2}, 1]$ since the optimal intersection over union (IoU) is 1 and the largest distance possible is $\sqrt{2}$ within the normalized image domain. We then search a bipartite matching of words from $W_\mathbf{D}$ to $W_{\hat{\mathbf{I}}}$ that maximizes the sum of the locality measurements for each word in $W_\mathbf{D}$ and each word in $W_{\hat{\mathbf{I}}}$ using the Hungarian Assignment algorithm [38]. The assignment is denoted by $M \subseteq W_\mathbf{D} \times W_{\hat{\mathbf{I}}}$. See Figure 9.1 for an example of the word matching.

Given the matched words, we then compute the normalized character error rate (*nCER*) as mentioned in [42] for each matched word w_u and w_i as follows:

$$nCER(w_u, w_i) = \frac{S + I + D}{S + I + D + C} \tag{9.2}$$

where S, I, D, and C are the number of substitutions, insertions, deletions, and correct characters, respectively. In contrast to the *CER*, the *nCER* ranges between 0 and 1, where 0 is the optimal score.

Our new metric *mnCER* is then defined as the average *nCER* overall matched words in M and a penalty value for each unmatched ground truth word in $W_{\hat{\mathbf{I}}}$:

$$mnCER = \frac{1}{|W_{\hat{\mathbf{I}}}|}\left[\sum_{(w_u,w_i)\in M} nCER(w_u,w_i) + |W_{\hat{\mathbf{I}}}| - |M|\right] \tag{9.3}$$

where $|W_{\hat{\mathbf{I}}}|$ is the number of words in $W_{\hat{\mathbf{I}}}$ and $|M|$ is the number of matched words in M. The penalty value for unmatched words equals the worst possible *nCER* value, which is 1. The overall *mnCER* ranges between 0 and 1, where 0 is the optimal score.

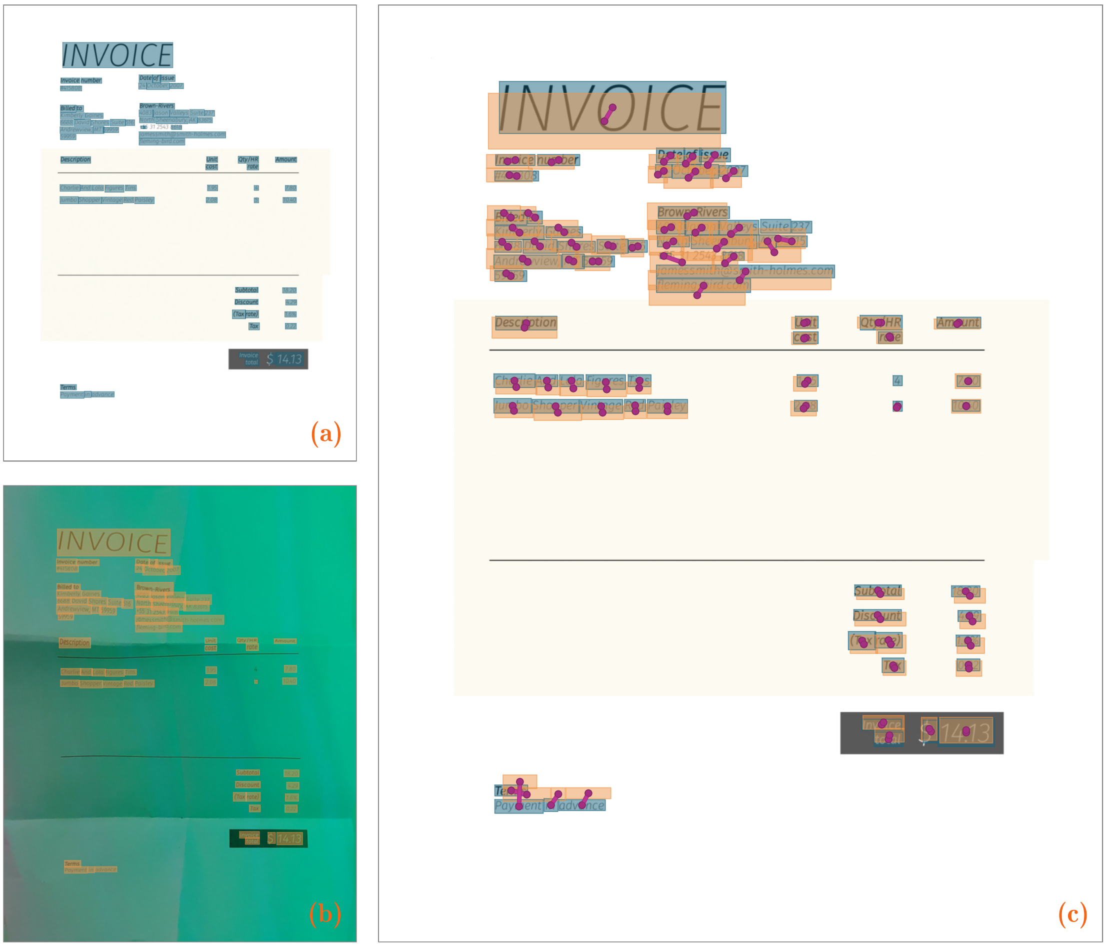


Figure 9.1: Example of the locality-based word matching for the *mnCER* calculation. Images (a) and (b) show the detected bounding boxes $b_{\hat{\mathbf{I}}}$ and $b_{\mathbf{D}}$, respectively. Image (c) displays the word matchings M.

9.3 Summary and Discussion

The new metric *mnCER* combines the advantages of both visual and text-based metrics while also avoiding their respective problems. Since it operates on the detected text in the images, it is sensitive to fine-grained dewarping errors that would make the document unreadable. At the same time, the metric does not suffer from the linearization problem, as there is no need to linearize the detected words. This leads to smooth evaluation scores with regard to minor changes in the word bounding box detections. Since we decided to employ the normalized Character Error Rate instead of the standard version, our metric is bound between 0 and 1, which facilitates the evaluation of document dewarping approaches on an absolute level. Furthermore, the metric behaves as expected for missing text: said text cannot be matched with a ground truth word and, therefore, will be included in the final score by a penalty term of one, i.e., the maximal penalty.

We answer the RQ2.2 on the evaluation of the quality of dewarping approaches with regard to text readability and positional awareness:

RQ2.2: Dewarping Metric

How can we **evaluate the quality** of the geometric dewarping process with regard to text readability and positional awareness?

The presented novel metric *mnCER* fulfills both requirements on a dewarping metric. It is sensitive to text-readability by comparing words from both images. In addition, the metric incorporates the positions of the words in the dewarped image $\mathbf{D}$ and the flatbed reference image $\hat{\mathbf{I}}$, which are crucial for the downstream task of information extraction. The accurate dewarping of words and correct matching with the ground truth words allows for a semantic interpretation of the dewarped words based on their location within the template.

10

Explicit Reference Template Leverage

In this chapter, we approach the problem of geometric dewarping from another angle. Our first solution - presented in Chapter 8 - takes an implicit approach to this problem, i.e., letting the model figure out how to leverage the information inside a reference template. Contrary to this approach, we seek a novel method of utilizing this information for geometric dewarping, using a multistage pipeline, where we can interpret all intermediate results. Unlike the implicit approach, we explicitly guide the model to use the reference template information for dewarping.

When comparing a warped document and the corresponding reference template as humans, we intuitively make connections between the visual elements like logos, words, or structural lines seen in both images. Using these connections, we can deduce where the warped elements must move and how they must be deformed to align with the reference template. To make that possible for AI, we need to solve a series of tasks: (1) find the document in the warped image, (2) detect all lines in the warped image and the template, (3) match the lines based on their positions and visual appearance and (4) utilize the knowledge of the matched lines to generate a dense backward mapping. Given the matches of visual elements in both images, we can create a backward map that straightens lines from the warped domain and moves them to the correct location within the flatbed domain. See Figure 10.1 for an illustration of this idea.

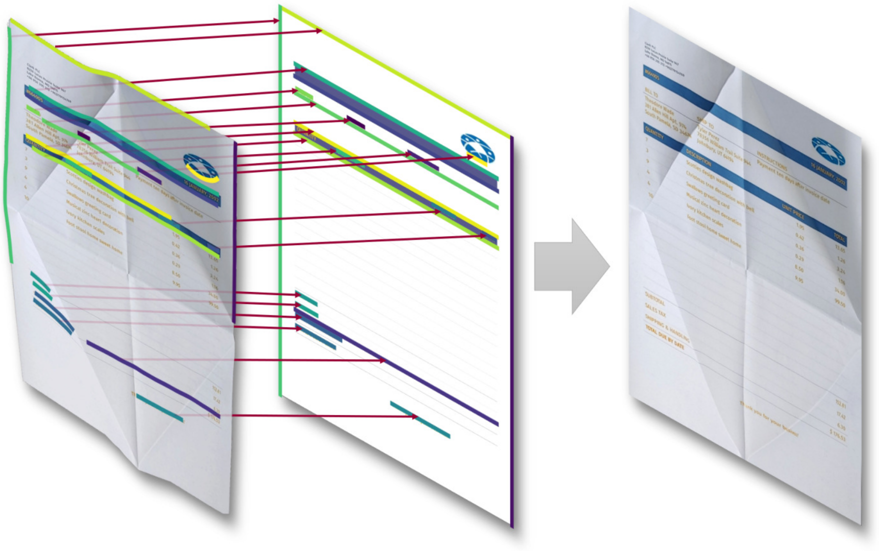


Figure 10.1: Illustration demonstrating the idea of line detection and matching for document image dewarping.

This chapter is based on the following publication:

Publication

Felix Hertlein et al. "DocMatcher: Document Image Dewarping via Structural and Textual Line Matching". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2025

We begin by introducing the methodology of our approach in Section 10.1, followed by a section on the evaluation details (Section 10.2). In Section 10.3, we present the results of our approach and compare them to the state-of-the-art methods. Finally, we conclude the chapter with a summary and a discussion of the results in Section 10.4.

10.1 Approach

Our approach consists of several stages, as illustrated in Figure 10.2. First, the background is removed from the input image. Subsequently, we detect the structural and textual lines in the warped document image and use them to pre-dewarp the document using a homography transformation. We then employ our document dewarping model GeoTrTemplateLarge twice to initially dewarp the document before we apply our new line-based dewarping method. This method detects structural and textual lines in the pre-dewarped document and matches them to the template lines. Given these line correspondences, we construct a dense transformation map and apply it to the pre-dewarped document to obtain the final dewarped document. We describe each stage in detail below.

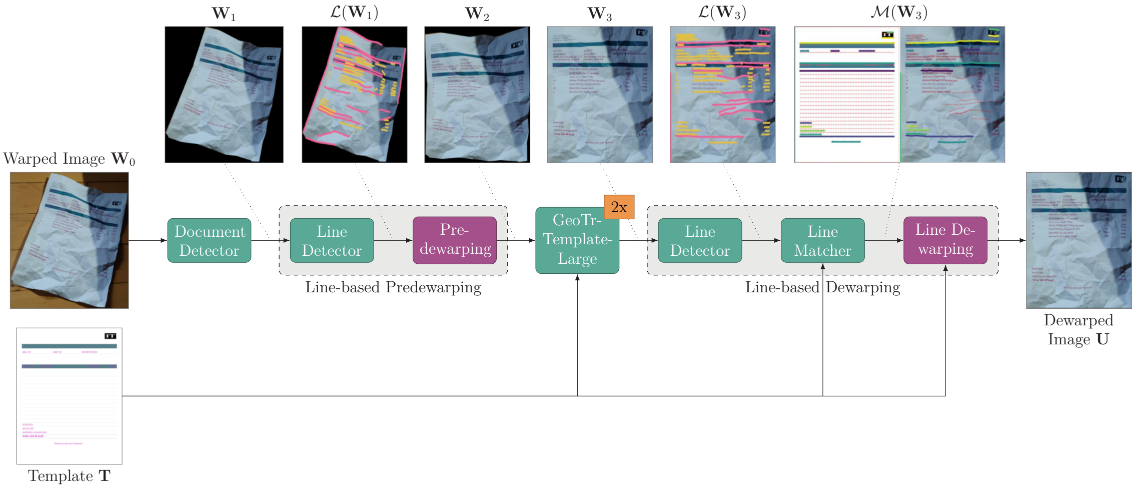


Figure 10.2: Overview of the proposed approach. First, the background is removed from the input image. Then, the structural and text lines are extracted from the background-removed image. We use the detected lines to pre-dewarp the image and then match the detected lines to the template lines. Given the line matches, we compute the dense backward map and apply it to the input image.

10.1.1 Document Detector

Given a warped document image $\mathbf{W}_0 \in \mathbb{R}^{h \times w \times 3}$, the first step is to remove the background from the image. The subsequent steps do not have to deal with the noise in the background. We achieve this by fine-tuning the state-of-the-art segmentation model Segment Anything (SAM) [36] on Inv3D for the semantic segmentation task. We set a total of two classes: the document class and the background class. For fine-tuning, we freeze SAM's image and prompt encoder and only train the mask decoder of the `ViT-L` model. The image input resolution is set to 1024 × 1024 pixels. The output of the model is a binary mask that segments the document from the background. We then apply the mask to the input image to remove the background. Let $\mathbf{W}_1 \in \mathbb{R}^{h \times w \times 3}$ represent the resulting image with the background removed.

10.1.2 Line Detector

Our line detector follows the approach of Lal et al. [40]. They treat the line detection task as an instance segmentation problem and propose a transformer-based model using masked attention called LineFormer. We train our own version of the LineFormer model on the Inv3D dataset for the task of structural and textual line detection. Since the Inv3D dataset does not provide structural line annotations, we generate the line annotations based on the optimal image $\hat{\mathbf{I}}$ using the Canny edge detector [9]. For training the LineFormer model, our lines need to be given a thickness. Structural lines are given a thickness of 3 pixels as proposed by the original work, while the thickness of textual lines is determined automatically by the area covered by the text. We extend the augmentations of LineFormer by adding photometric distortions, random mesh distortion, and randomly overlaying images of the Describable Textures Dataset [13] from the categories *stained* and *wrinkled* to improve the transfer from simulated to real-world images. In contrast to the original LineFormer model, we extract 2D line paths instead of mathematical functions. Given the binary mask for each object candidate, we apply the skeletonization algorithm [105] to limit the line thickness to a maximum of one pixel. We find our 2D line path by converting the binary map to a graph, where each line pixel is a node and two nodes are connected if they are neighboring and search for the longest path within said graph. Figure 10.3 visualizes the 2D line path detection.

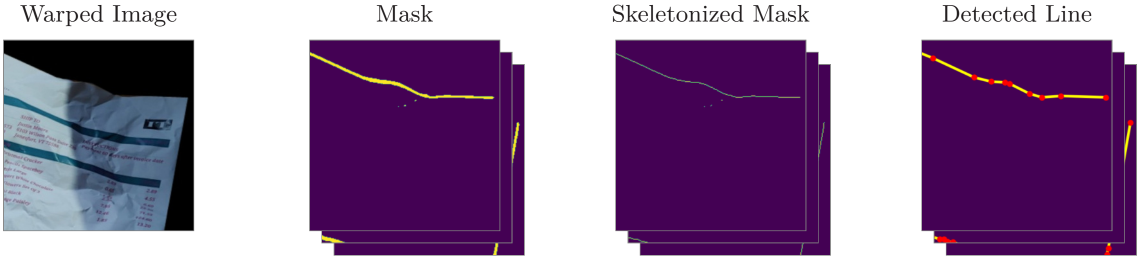


Figure 10.3: Visualization of the 2D line path detection. For better visibility, we display only a cutout of the warped image and the line detection artifacts.

In a post-processing step, we remove duplicate lines and lines with a length below a certain threshold. We denote the detected lines as $\mathcal{L}(\mathbf{W}_i)$ for $\mathbf{W}_i$ being the input image.

10.1.3 Pre-Dewarping

To simplify the problem, we pre-dewarp the image using the detected lines by estimating a homography transformation. We base the pre-dewarping on the detected structural and textual lines instead of the document outline, as the document outline is not always correctly detected in complex cases. The idea behind our homography estimation is to axis align all detected lines as precisely as possible. For structural text lines, any axis is suitable, while for textual lines, we need to consider the horizontal axes.

Given a set of detected lines $\mathcal{L}$, we estimate the homography matrix $\mathbf{H}$ by minimizing the following objective function:

$$\arg\min_{\mathbf{H}} \sum_{l \in \mathcal{L}} \Big[\|p(l, \mathbf{H})\|_2 \cdot \min_{r \in \mathcal{R}} \operatorname{angle}(p(l, \mathbf{H}), r) \Big]^2 \tag{10.1}$$

with

$$p(l, \mathbf{H}) = \operatorname{project}_{\mathbf{H}}(\operatorname{approximate}(l)) \tag{10.2}$$

where approximate$(\cdot)$ maps a 2D line path the best fitting line using linear regression and $\mathcal{R}$ are the axis-aligned unit vectors to compare to. For text lines, $\mathcal{R}$ are the two horizontal unit vectors, while for structural lines, $\mathcal{R}$ are the four unit vectors. See Figure 10.4 for a visualization of the pre-dewarping optimization.

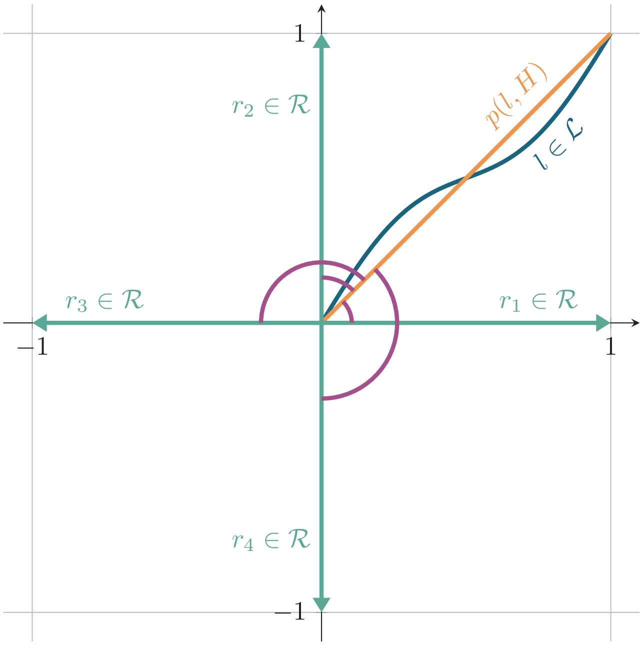


Figure 10.4: Visualization of the pre-dewarping optimization.

The homography matrix is constructed as follows:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{RS}} & 0 \\ & & 0 \\ v_0 & v_1 & 1 \end{bmatrix} \tag{10.3}$$

$$\mathbf{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \tag{10.4}$$

Thereby, it allows for rotation by an angle α, shear by a scalar s, and projection using the scalars v_0 and v_1. Since our objective function contains multiple local minima with respect to the rotation α, we search for a suitable initial value denoted as $\alpha_0 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ by densely evaluating the objective function with fixed shear and projection.

For stability, we minimize the objective function using constrained optimization, where the variables are restricted within the bounds: $\alpha \in \left[\alpha_0 - \frac{\pi}{4}, \alpha_0 + \frac{\pi}{4}\right]$, and $v_0, v_1 \in \left[-1^{-3}, 1^{-3}\right]$. Given the minimized homography matrix $\hat{\mathbf{H}}$, we set the scaling and translation values such that the whole document is still inside the image after applying the homography transformation. We apply the final transformation to the background-removed image $\mathbf{W}_1$, which results in $\mathbf{W}_2$. See Figure 10.2 for an example.

10.1.4 Template-based Document Dewarping

We use the previously introduced template-based document dewarping model GeoTrTemplateLarge to dewarp the image $\mathbf{W}_2$. It is based on an encoder-decoder architecture on top of deep embeddings of the input and template image. The attention mechanism allows the model to connect template information with the dewarped image. Thereby, it integrates the template information implicitly. Our approach leverages the dewarping capabilities of GeoTrTemplateLarge before we add our more explicit line-based dewarping approach. We apply GeoTrTemplateLarge twice to the pre-dewarped image $\mathbf{W}_2$ to obtain the initial dewarped image $\mathbf{W}_3$.

10.1.5 Line Matcher

Given the initially dewarped image $\mathbf{W}_3$ and the template image $\mathbf{T}$, we aim to match the detected lines $\mathcal{L}(\mathbf{W}_3)$ to the template lines. To achieve this, we encode the line features and match them using the state-of-the-art local feature matcher LightGlue [49]. The line encoding consists of multiple parts: a visual encoding, a positional encoding, and an optional text encoding. See Figure 10.6 for a visualization of the line encoding. For the prior two encodings, we extend the input image $\mathbf{W}_3$ with a n-dimensional sinusoidal position encoding [89]. In our case, we set the dimensionality to 8, which results in an 11-dimensional input image. For each line $l \in \mathcal{L}(\mathbf{W}_3)$ and its width, we transform the line mask from Euclidean to Frenet coordinates [91]. The Frenet coordinates of a given pixel (x, y) are given by the distance d along the line and the distance o orthogonal to the line. When transforming the line mask to Frenet coordinates, we obtain a compact representation of the line $r_l \in \mathbb{R}^{256\times8\times11}$, which entails information about the line shape, position, and visual appearance.

Figure 10.6 shows an example of the visual and positional line features after transforming them from Euclidean to Frenet coordinates.

We split the Frenet line encoding into eight chunks of 8×8 pixels and apply a vision transformer [19] to the patch sequence, yielding a deep line embedding with 256 dimensions. For text lines, we use a

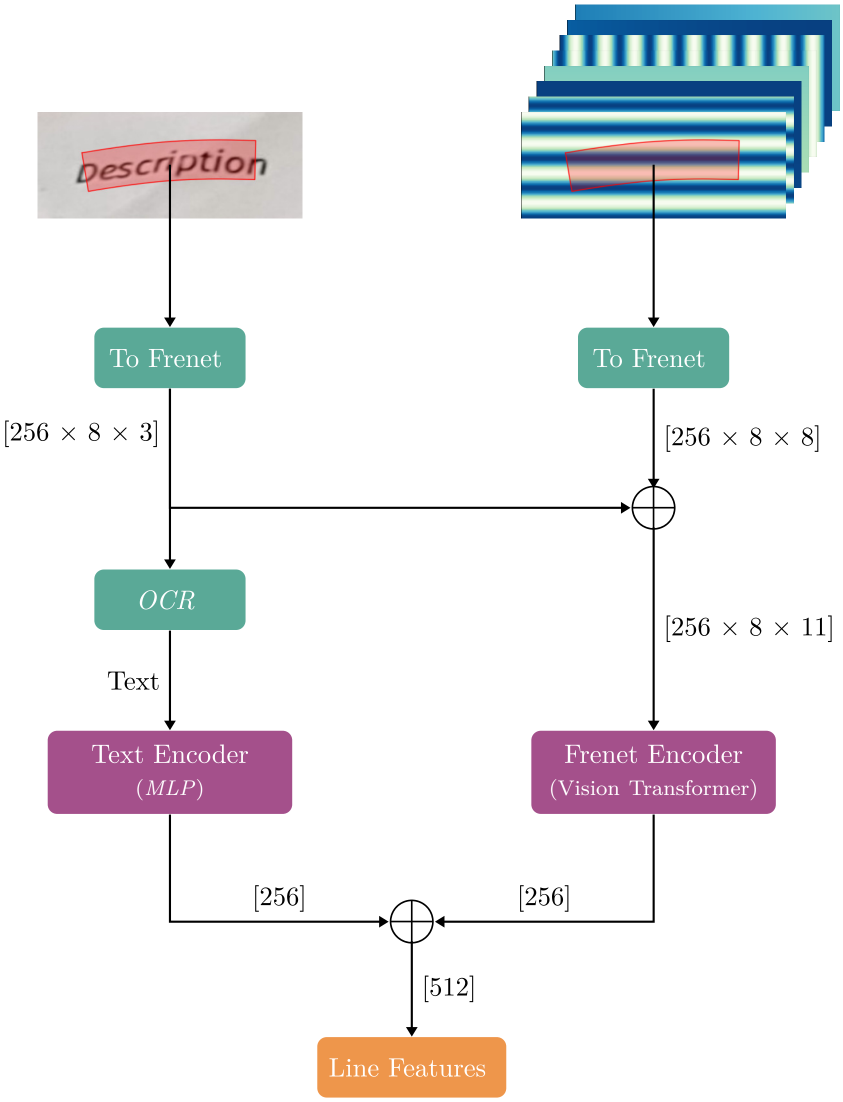


Figure 10.5: Visualization of the line encoding.

pre-trained *OCR* model called docTR [65] to extract the text within the Frenet space and encode it with a simple *MLP* with two hidden layers, also resulting in a 256-dimensional vector. The concatenation of the vision transformer and the text encoding then gives the full line representation.

To match warped and template line embeddings, we use the local feature matcher LightGlue [49]. We train our model on the Inv3D dataset with the same hyperparameters as in the original paper. At inference time, we remove false matches by filtering out matches with a log assignment probability below the threshold t, matches without matching line types, and text matches without a common substring of at least three characters. We set the log assignment probability threshold to $t = -1$. This post-processing step reduces the number of false matches and thus improves the final dewarping result. We denote the matches as $\mathcal{M}(\mathbf{W}_3)$. See Figure 10.2 for an example.

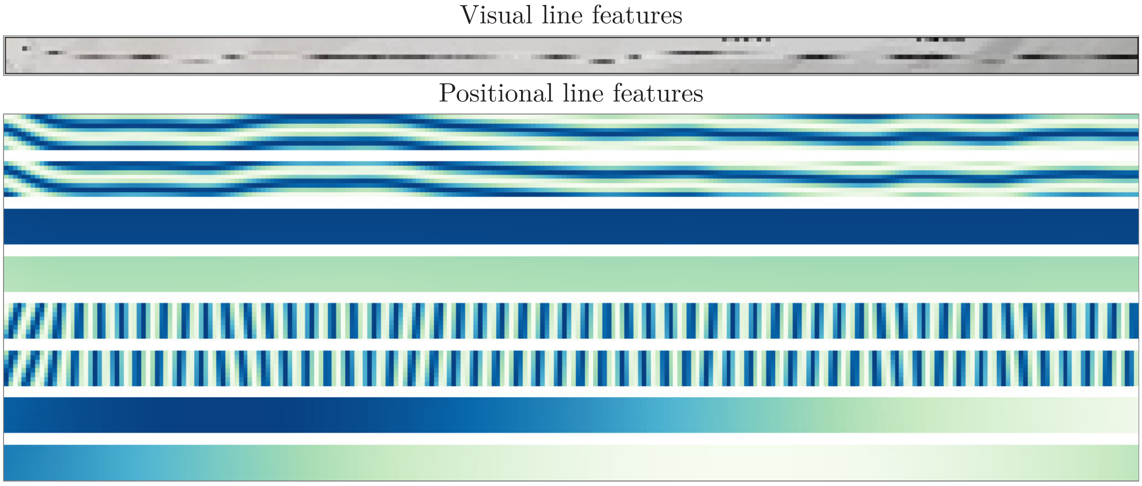


Figure 10.6: Visualization of a line encoding in Frenet space.

10.1.6 Line Dewarping

Given the matches $\mathcal{M}(\mathbf{W}_3)$, we construct a dense coordinate mapping to dewarp the image. We denote this mapping, called forward map, as $\mathbf{F} \in [0,1]^{512\times512\times2}$. We start with the identity transformation and iteratively enhance the forward map by incorporating line correspondences. This is done by adding point correspondences and interpolating the missing vectors using Delaunay Triangulation [41]. The construction of the forward map $\mathbf{F}$ consists of three stages: known matches, support points, and unmatched lines. Figure 10.7 shows the construction of the forward map at each stage.

1. Project matches. In the first stage, we consider the known matches in descending order of assignment probability. For each match, we densely sample point correspondences along the warped and template line and add them to the forward map, thus forcing the pixels at the point correspondences to their location on the template. For text lines, we generate the point correspondences for both x and y coordinates. As for the structural lines, we only add either the x- or the y-coordinates to the forward map, depending on the template line direction. This is because structural lines only constrain the position in one direction. For example, if the structural line is horizontal, we only add the y-coordinates to the forward map.

2. Support points. For the second stage, we add support points outside the convex hull of all correspondences to reduce the interpolation error. Delaunay triangulation outside the convex hull of the correspondences (and within the image bounds) creates pixel shifts due to the triangular nature of the interpolation. Our goal with the support points is to retain the straightness of the image in those regions. To achieve this, we project the convex hull points to the left/right or top/bottom of the image space, depending on the gradient direction of each channel. Figure 10.7 shows an example of the support points.

3. Unmatched lines. In stage three, we want to utilize the knowledge about all detected but unmatched lines. To achieve this, we project the unmatched lines with the current forward map and straighten the lines in the dewarped space. This allows us to project the unmatched lines similar to stage 1 and, thus, enforce the straightness of the unmatched lines in the final projection.

Since the line matcher occasionally produces false matches, we need to ensure that these false matches do not influence the final dewarping. To achieve this, we check the validity of the forward map during the iterative construction and reject all matches or lines that contradict our invariant. We define the invariant as follows: for each position in the forward map, the gradient must not exceed a range of valid values. We set the lower bound to 0 and the upper bound to 0.0025, respectively. This effectively limits the maximum compression and stretching for each part of the image and, thus, disregards extreme cases. Due to the minor imprecisions of the line detector, our constructed forward map contains minor artifacts that can lead to text distortion. In the final step, we apply a mean smoothing kernel to the backward map to retain the text readability.

We denote the final forward map as $\mathbf{F}$. By inverting the forward map, we obtain the backward map $\mathbf{B}$ and apply it to the pre-dewarped image $\mathbf{W}_3$. This yields the final dewarped image $\mathbf{U}$.

10.2 Evaluation

We evaluate our new approach, DocMatcher, on Inv3DReal (see Chapter 6) on the established metrics *MS-SSIM*, *LD*, *LPIPS*, *ED*, *CER*, as well as our novel metric *mnCER*. See Section 2.3 for details on the established metrics and Chapter 9 for the novel metric. Note that we changed the *OCR* engine from Tesseract [78] to docTR [65] for the evaluation of the text-based metrics, as docTR outperforms Tesseract by a large margin.

We compare DocMatcher with our previous work GeoTrTemplateLarge and other related works on geometric dewarping, namely DewarpNet [16] (w/o refinement network), GeoTr [21], as well as the baseline method identity, where no geometric dewarping was applied to the warped input image. Please note that we employed the DewarpNet without the refinement network, as it is designed to perform illumination correction. Since geometric dewarping and illumination correction are separate problems that can be applied independently, leaving the refinement network out yields a fair comparison of the methods.

10.3 Results

This section presents the results of our approach evaluated on the Inv3DReal dataset. We compare our approach to the identity mapping as a baseline, as well as our previously described document dewarping model GeoTrTemplateLarge, and the related works GeoTr [21] and the DewarpNet [16] without refinement network. Note that the latter two papers present an approach for document image dewarping in combination with illumination correction. Since our paper focuses on document image dewarping, we only compare against the dewarping part of the models. Furthermore, it is important to note that of our baselines, only the model GeoTrTemplateLarge utilizes a template image for dewarping. The other models are template-free and learn the dewarping based on the input image only. To the best of our knowledge, no other recent model uses a template image for document image dewarping.

The following section presents the quantitative and qualitative results, as well as an ablation study.

10.3.1 Quantitative Results

The quantitative results of our approach are shown in Figure and Table 10.1. When comparing our approach to the state-of-the-art models - GeoTr, DewarpNet, and GeoTrTemplateLarge and the

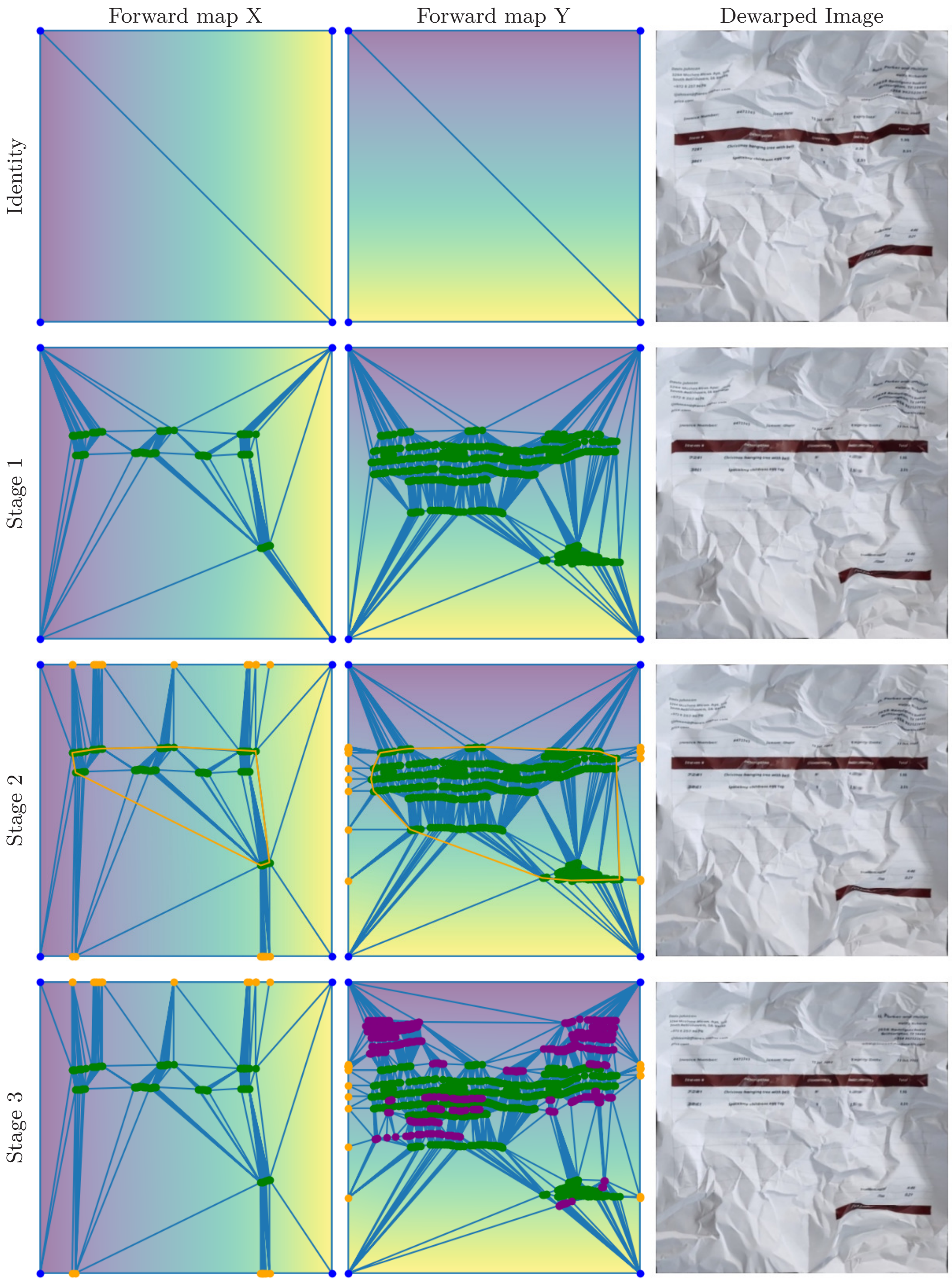


Figure 10.7: Construction of the forward map **F** for the line-based dewarping shown at multiple stages.

baseline identity mapping, we can observe that our approach outperforms all models in all metrics. The comparison with our previous approach GeoTrTemplateLarge shows a significant improvement of 9.6 % in *MS-SSIM*, 11.8 % in *LPIPS*, 32.6 % in *LD*, 40.2 % in *mnCER*, 12.87 % in *ED*, and 11.68 % in *CER*. In particular, the metrics *LD* and *mnCER* show the most significant improvements. Both metrics include the positioning of the elements in the dewarped image, indicating that the positioning of the text and visual elements improved significantly. The visual metrics *MS-SSIM* and *LPIPS* also show a considerable improvement, indicating that the dewarped image looks closer

	MS-SSIM ↑	*LPIPS* ↓	*LD* ↓	*mnCER* ↓	*ED* ↓	*CER* ↓
Identity	0.44 (0.10)	0.6 (0.10)	36.77 (13.54)	0.78 (0.14)	394.64 (140.81)	0.63 (0.19)
DewarpNet (w/o ref) [16]	0.55 (0.11)	0.42 (0.12)	25.27 (10.37)	0.58 (0.17)	254.92 (131.39)	0.41 (0.19)
GeoTr [21]	0.56 (0.11)	0.41 (0.12)	23.25 (10.16)	0.52 (0.18)	192.46 (118.88)	0.31 (0.18)
GeoTrTemplate-Large (previous)	0.65 (0.12)	0.31 (0.13)	16.82 (10.31)	0.28 (0.18)	128.33 (99.09)	0.20 (0.15)
DocMatcher (ours)	**0.71 (0.12)**	**0.27 (0.12)**	**11.34 (8.26)**	**0.17 (0.13)**	**111.81 (79.56)**	**0.18 (0.13)**

Table 10.1: Quantitative results of our approach DocMatcher in comparison to the state of the art on the Inv3DReal dataset. All models were trained on Inv3D and evaluated on Inv3DReal. The values are given as mean and standard deviation.

to the original image. For the text-based metrics *ED* and *CER*, we also observe an improvement, but it should be noted that these metrics suffer from the linearization problem explained above and thus are not as reliable as the *mnCER* metric. Notably, the absolute numbers for the text-based metrics are significantly lower than those reported in our previous work, GeoTrTemplateLarge. This is due to the change of the *OCR* engine from Tesseract to docTR, which handles challenging conditions better.

Overall, the results indicate that our approach is capable of dewarping document images more accurately than the previous state-of-the-art models.

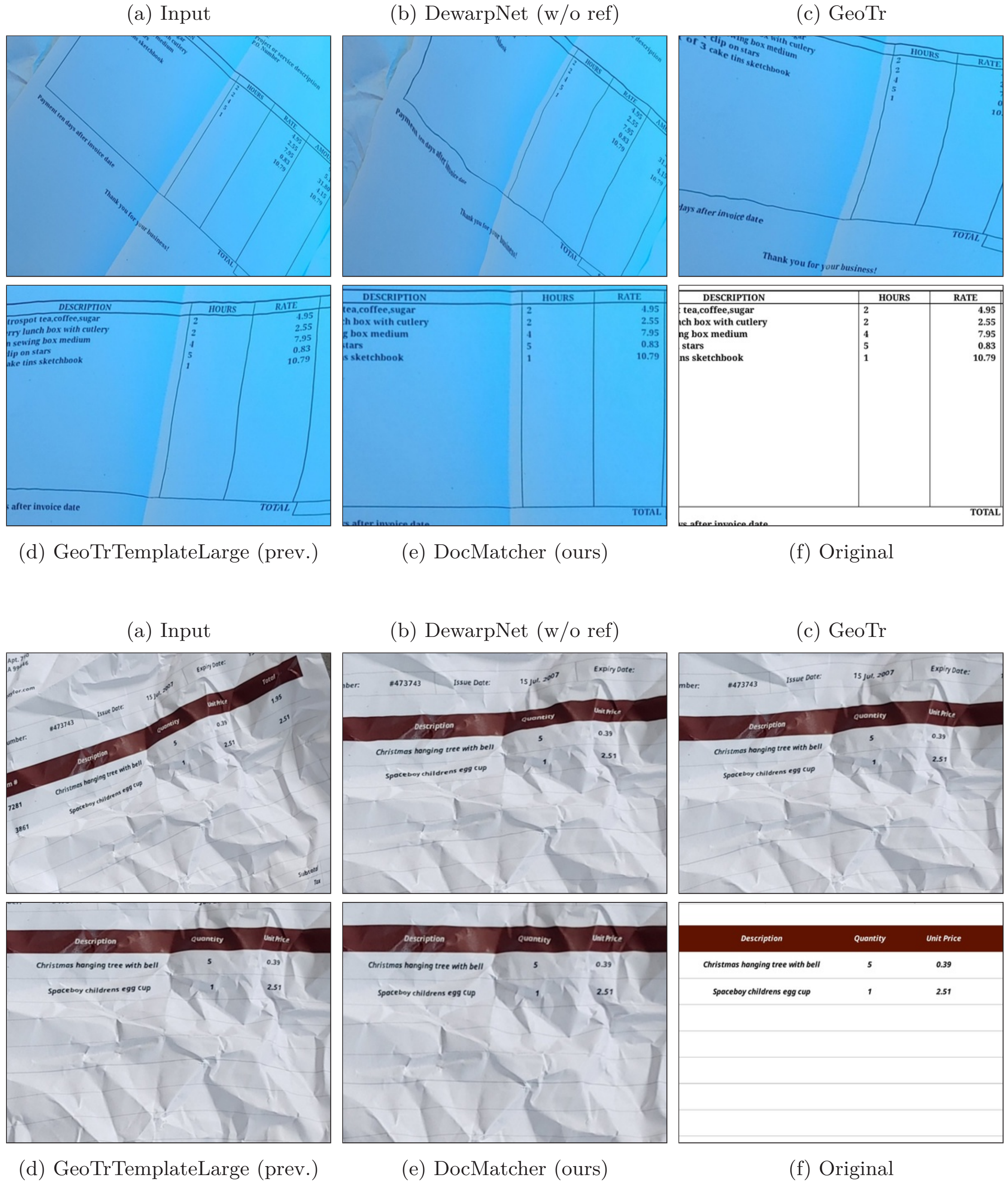
10.3.2 Qualitative Results

We show the qualitative results of our approach compared to the state-of-the-art models for several selected samples in Figure 10.8. The first column displays the input image, whereas the last column presents the original image. The second to fifth columns show the results of the models DewarpNet (w/o ref) [16], GeoTr [21], GeoTrTemplateLarge, and our new approach DocMatcher, respectively. We cropped the same region for all images to simplify the comparison. Thereby, it is easier to see the differences in the positioning of the text and visual elements.

In these examples, we can observe that this approach yields dewarped images, which are visually closer to the original image than the other models. The overall positioning of the elements and the straightness are improved, even under challenging lighting conditions (top row) or strong deformations (bottom row).

We also show the intermediate stages of our approach in Figure 10.9. From image (a) to (f), we display the input image, the background-removed image, the homography transformation, the GeoTrTemplateLarge dewarping, the line matcher (2x), and the final dewarped image, respectively. In this example, the background removal did not correctly remove all of the background since the differentiation between the document foreground and background is challenging. Since the pre-dewarping step is not dependent on accurate background removal, the following steps are not overly affected. This demonstrates the advantage of content-based dewarping as in our approach compared to outline-based dewarping.

We want to point out that the line-based dewarping stage can generate black artifacts at the border of the dewarped images. That effect can occur when the line-based dewarping moves the pixels closer to the center, and no pixels outside the image boundary can fill the gap. We experimented with filling these artifacts by the color value of the nearest valid pixel. This variant yields a *MS-SSIM* of 0.72, a *LPIPS* of 0.26, a *LD* of 15.04, a *mnCER* of 0.17, an *ED* of 111.74 and a *CER* of



(a) Input (b) DewarpNet (w/o ref) (c) GeoTr

(d) GeoTrTemplateLarge (prev.) (e) DocMatcher (ours) (f) Original

(a) Input (b) DewarpNet (w/o ref) (c) GeoTr

(d) GeoTrTemplateLarge (prev.) (e) DocMatcher (ours) (f) Original

Figure 10.8: Qualitative evaluation of the state of the art and our approach based on selected samples of Inv3DReal.

0.18. When comparing it to our proposed variant, we observe a significant change in *LD*, while the other metrics remain roughly the same. This sensitivity to the border artifacts indicates that the metric *LD* is not robust to small changes in the image content.

We also show the least successful dewarpings per metric in Figure 10.10. The reasons for the bad metrics differ depending on the metric. Heavy crumples or blurry images appear to be the most

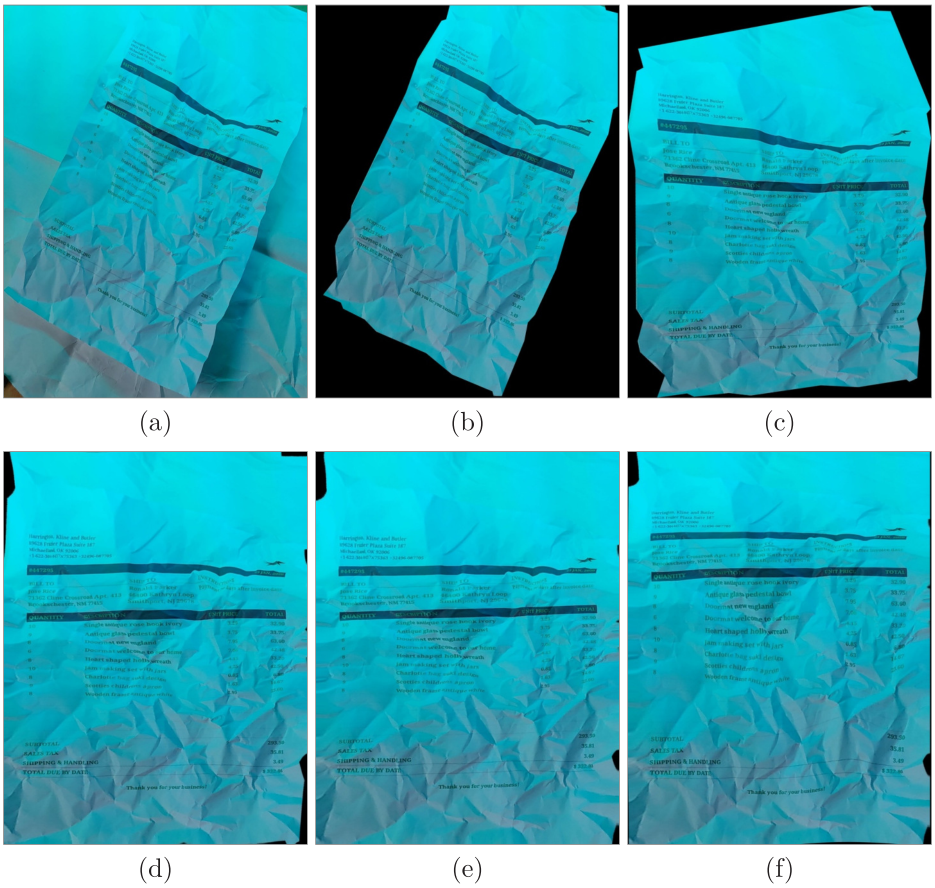


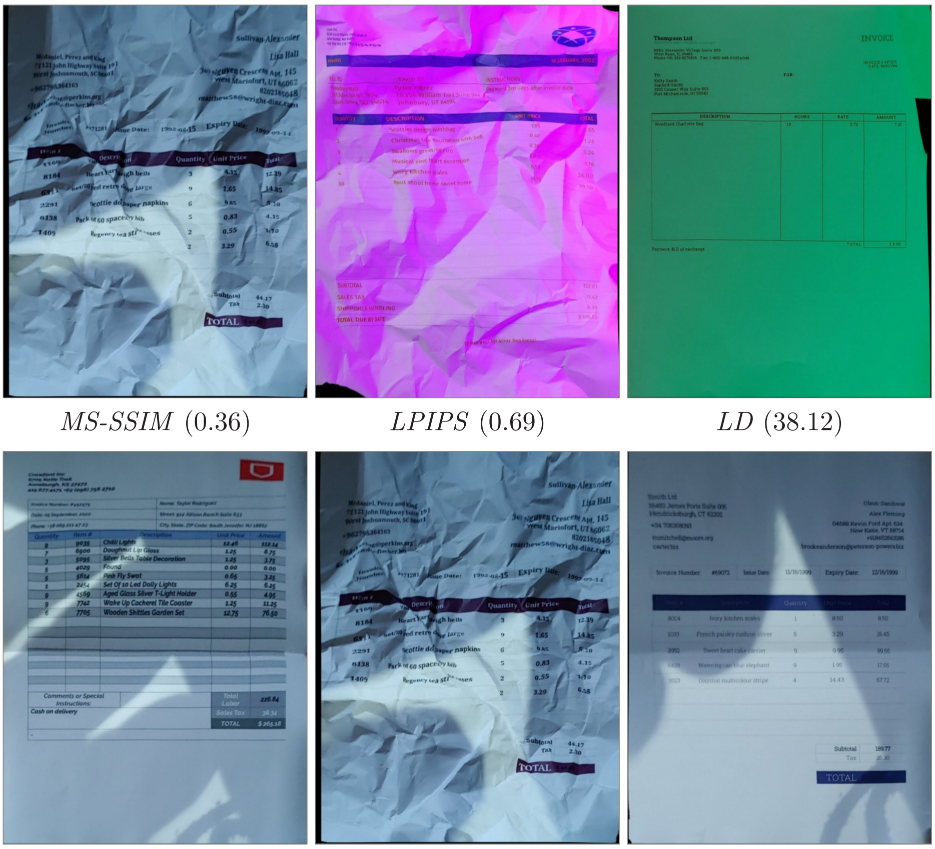
Figure 10.9: Visualization of intermediate dewarping stages within our proposed approach. From (a) to (f): input image $\mathbf{W}_0$, background-removed image $\mathbf{W}_1$, pre-dewarped image $\mathbf{W}_3$, initially dewarped using GeoTrTemplateLarge (2x) $\mathbf{W}_4$, and the final output $\mathbf{U}$

challenging for our approach. Overall, the dewarped images still appear reasonably well dewarped, even in the worst cases. This indicates that there are no extreme failure cases.

10.3.3 Ablation Study

To evaluate the impact of the individual components of our approach, we conduct an ablation study. We ablate the full model piece by piece until we reach the base model GeoTrTemplateLarge [31] and evaluate the performance on the Inv3DReal dataset. We compare the full model to the following ablations: (1) without the line-based dewarping, (2) without the second inference of GeoTrTemplateLarge, (3) without the line-based pre-dewarping and finally without the background removal (4). The last ablation is equal to the model GeoTrTemplateLarge itself.

Figure 10.11 presents the ablation results. The first ablation, *w/o line-based dewarping*, shows significant degradation in *LD* and more minor degradation in *mnCER*. All other metric values are roughly the same as the full model. For the second ablation, *w/o second GeoTrTemplateLarge*, we find the most prominent degradation in *mnCER*. The third ablation, *w/o line-based pre-dewarping*, shows significant degradation in all metrics, especially in *mnCER*. Finally, the last ablation, *w/o background removal*, shows minor degradation in most metrics. While this step appears to have the least effect, it still contributes to the model's overall performance. This indicates that all



MS-SSIM (0.36) *LPIPS* (0.69) *LD* (38.12)

mnCER (0.74) *ED* (359) *CER* (0.59)

Figure 10.10: Least successful dewarpings in Inv3DReal with our approach per metric.

components are relevant to the full model. Considering the varying degrees of degradation among the metrics and ablations, it is reasonable to assume that each component has its own merits.

10.3.4 Susceptibility to Errors

In this section, we investigate the susceptibility of our approach to errors in the intermediate stages of our pipeline. To that end, we design two experiments: The first experiment examines the influences of incorrect mask detection by the document detector. During inference, we rotate the detected document mask around its center by a random angle drawn uniformly from the interval $[-\alpha, \alpha]$. For the second experiment, we investigate the effect of incorrect line detections on the subsequent stages. Therefore, we randomly generate distractor lines during line detection by merging existing lines and adding them to the detected lines.

The results of the error susceptibility experiment are shown in Table 10.2. The rotation of the document masks and the addition of distractor lines show little effect for small changes. For significant changes, the impact of the added errors differs on the kind of error. Distractor lines seem to hardly influence the overall performance, while the mask rotation shows a decline in all metrics. However, it should be noted that even at the maximum rotation, the visual metrics and *mnCER* are still better than our previous approach, GeoTrTemplateLarge, and significantly better than all other benchmarks. The degradation in the text metrics might be caused by loss of

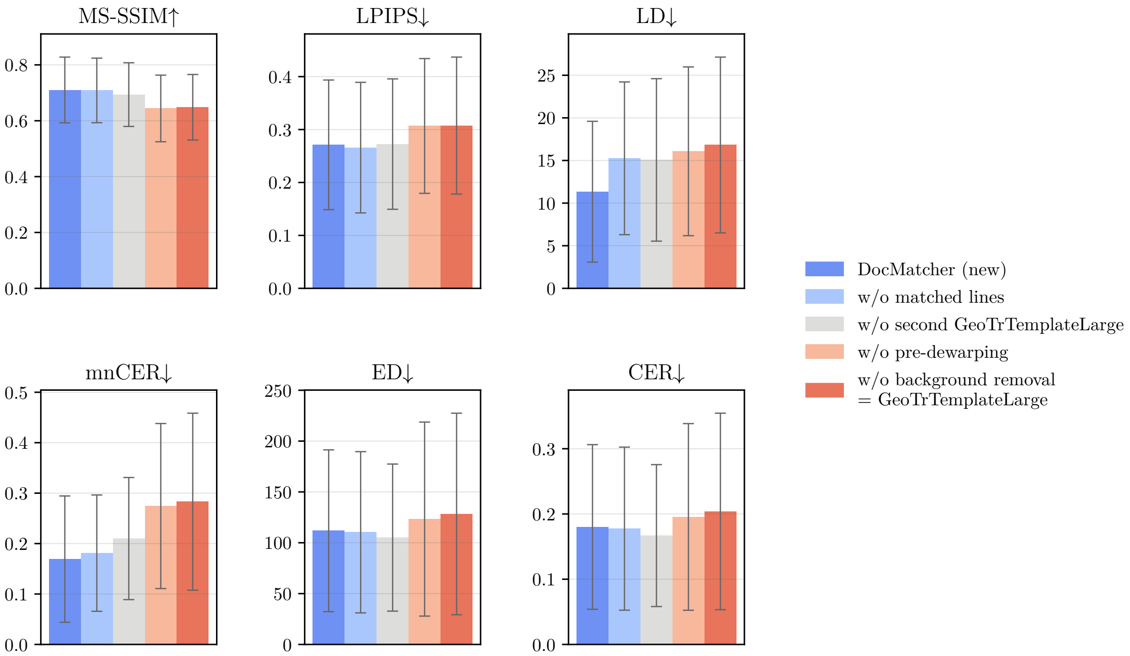


Figure 10.11: Ablation study for our proposed approach DocMatcher. We remove parts of the model piece by piece until we have only the GeoTrTemplateLarge [31] remaining.

		MS-SSIM↑	*LPIPS*↓	*LD*↓	*mnCER*↓	*ED*↓	*CER*↓
Rotate	$\alpha = 25°$	0.68	0.30	12.0	0.23	145	0.23
	$\alpha = 20°$	0.68	0.29	11.6	0.21	141	0.23
	$\alpha = 15°$	0.69	0.28	11.6	0.18	129	0.21
	$\alpha = 10°$	0.70	0.28	**11.3**	**0.17**	124	0.20
	$\alpha = 5°$	**0.71**	**0.27**	**11.3**	**0.17**	116	0.19
	ours	**0.71**	**0.27**	**11.3**	**0.17**	112	**0.18**
Distractors	5%	**0.71**	**0.27**	11.5	**0.17**	115	**0.18**
	10%	0.70	0.28	11.5	**0.17**	113	**0.18**
	15%	0.70	0.28	11.5	0.18	**111**	**0.18**
	20%	0.70	0.28	11.8	0.18	112	**0.18**
	25%	0.70	0.28	11.7	0.19	117	0.19

Table 10.2: Results of the error susceptibility experiment.

information through incorrect masking. Overall, our approach shows robustness against errors in intermediate stages.

10.4 Summary and Discussion

In this chapter, we proposed a novel approach for geometric dewarping using reference templates that explicitly leverage the additional information in the reference template.

It utilizes a template image to guide the dewarping process and achieves state-of-the-art results on the Inv3DReal dataset. We leverage the line orientation for a robust, content-based image pre-dewarping stage to reduce the pixel shift distance. Furthermore, we associate the lines between the warped and template image and leverage these associations to construct a dense transformation field that is capable of moving the document pixels to their correct position. We evaluated our approach on the Inv3DReal dataset and compared it to the state-of-the-art models. The results show that our approach outperforms the previous best model, GeoTrTemplateLarge, in all metrics, especially in *LD* and *mnCER*. We conducted an ablation study to evaluate the impact of the individual components of our approach. Furthermore, we performed two experiments on the susceptibility of intermediate errors and showed that our approach is robust against these errors.

In this chapter, we addressed the research question RQ2.3:

RQ2.3: Explicit Geometric Dewarping

How can we dewarp document images with a reference template by **explicitly leveraging the template information** to improve the quality of the document images?

Our approach, DocMatcher, answers this question since it explicitly leverages the reference template information in the form of detected and matched lines to geometrically correct the warped document images. Our results show that the explicit approach outperforms our implicit geometric dewarping model proposed in RQ2.1.

Part V

Illumination Correction

11 Problem Formalization

In the previous part, Geometric Dewarping, we searched for a method to remove the geometric distortions induced by photographing the document using a smartphone. This effort tackled the challenge categories *Camera Distortions* and *Paper Distortions* as defined in Section 1.1.

Building upon the results of the previous part, we address the remaining challenge group: *Lighting*. In this group, we face the problems of ambient light and shadows, which are also captured by the smartphone. For an optimal representation of the original document, we are interested in a version of the captured and dewarped document that resembles the pristine document as closely as possible. To this end, we formulate the research question as follows:

RQ3: Illumination Correction

Given the partially dewarped document images, how can we **correct the illumination** to improve the quality of the document images?

Note that the problem of illumination correction is usually approached after geometrically dewarping a document image. Despite all advances in geometric dewarping, the results of that stage still contain warpings since the dewarping approaches are not perfect. Crucially, our investigation on illumination correction needs to cope with imperfectly geometrically dewarped images.

This chapter is based on the following publication:

Publication

Felix Hertlein and Alexander Naumann. “Template-guided Illumination Correction for Document Images with Imperfect Geometric Reconstruction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. IEEE, 2023, pp. 904–913. DOI: `10.1109/ICCVW60793.2023.00097`

Since we assume to have reference templates available for geometric dewarping, we can also leverage this a priori knowledge for illumination correction. The template image contains information about the document’s true colors, easing the illumination correction task since a correction model does not need to guess the output colors. Figure 11.1 gives a brief overview of the task of illumination correction of imperfectly dewarped document images using reference templates.

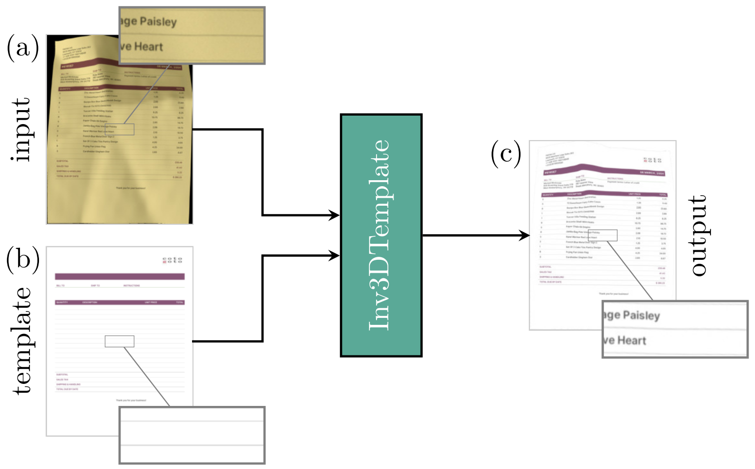


Figure 11.1: Overview of the illumination correction task of imperfectly dewarped document images using reference templates.

Given an imperfectly geometrically dewarped document image $\mathbf{I_{dwp}} := \mathbf{B}(\mathbf{W})$, where $\mathbf{B}$ is an imperfect backward mapping as defined in [31] and a warped document image $\mathbf{W}$, our goal is to find a mapping ϕ from $\mathbf{I_{dwp}}$ to $\mathbf{I_{ill}}$ such that all illumination effects are removed from the image:

$$\begin{aligned} \phi\colon \mathbb{R}^{\mathrm{h}\times \mathrm{w}\times 3} &\to \mathbb{R}^{\mathrm{h}\times \mathrm{w}\times 3} \\ \mathbf{I_{dwp}} &\mapsto \mathbf{I_{ill}} \end{aligned}$$

More specifically, our model is supposed to predict the (partially) dewarped albedo map $\mathbf{I_{ill}} := \mathbf{B}(\mathbf{A})$ for an albedo image $\mathbf{A}$ which corresponds pixel-wise to the warped image $\mathbf{W}$. Note that the goal for our model is to learn solely the illumination correction task and not to complete the partial geometric dewarping. Therefore, we define the ground truth image as the partially dewarped albedo map $\mathbf{B}(\mathbf{A})$ instead of the perfect flat document.

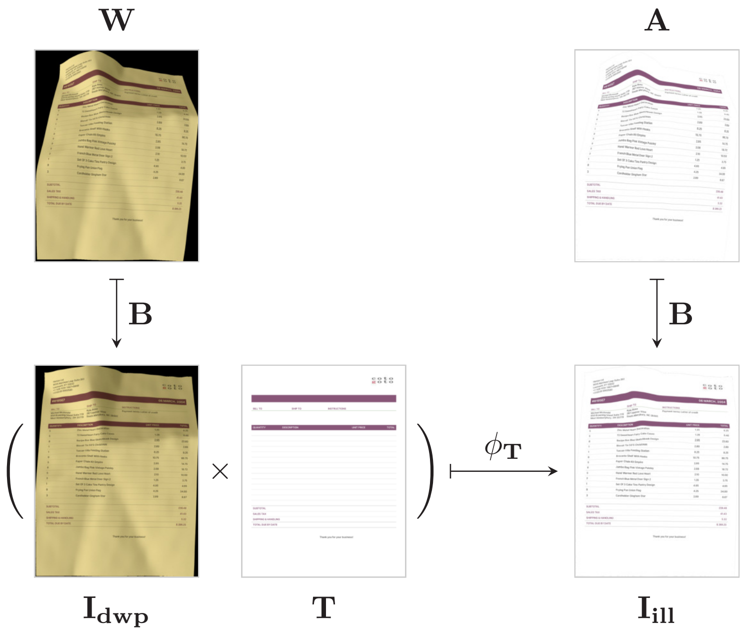


Figure 11.2: Visualization of the illumination correction problem.

To facilitate the learning task for our model, we leverage a priori known information about the image structure given as a template image $\mathbf{T} \in \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times 3}$. Formally, this is described as follows:

$$\begin{aligned} \phi_{\mathbf{T}} \colon\ \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times 3} \times \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times 3} &\to \mathbb{R}^{\mathrm{h}\times\mathrm{w}\times 3} \\ (\mathbf{I_{dwp}}, \mathbf{T}) &\mapsto \mathbf{I_{ill}} \end{aligned}$$

See Figure 11.2 for a visualization of the illumination correction problem.

12 Template Leverage

Given the problem of illumination correction for imperfectly dewarped document images, we propose a novel approach to leverage a priori known information about the image structure in this chapter. For this purpose, we introduce a model that can exploit the visual cues provided by a reference template image to improve the illumination correction performance. Since there are multiple ways to incorporate the template information, we will investigate two different approaches with a total of four variants.

This chapter is based on the following publication:

Publication

Felix Hertlein and Alexander Naumann. "Template-guided Illumination Correction for Document Images with Imperfect Geometric Reconstruction". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.* IEEE, 2023, pp. 904–913. DOI: 10.1109/ICCVW60793.2023.00097

We begin by proposing our novel approach to the problem of illumination correction in Section 12.1. Following up, we present our evaluation details in Section 12.2, including the training and evaluation datasets, as well as the employed metrics. In Section 12.3, we show the quantitative and qualitative results of our approach, as well as three ablation studies. We conclude this chapter with a summary and discussion in Section 12.4.

12.1 Approach

We base our architecture on the state-of-the-art document image enhancement model IllTr [21]. It was published by Feng et al. [21] as part of the image dewarping and illumination correction model named DocTr. We briefly summarize the model architecture of IllTr in the following:

Since the partially dewarped documents $\mathbf{I_{dwp}}$ contain high-frequency signals, IllTr avoids scaling the input to a fixed size. Instead, the document is split into slightly overlapping patches with a fixed size of $p \times p$ pixels and processed individually before being stitched together afterwards. Each patch $\mathbf{P_{img}} \in \mathbb{R}^{p \times p \times 3}$ is then preprocessed by a convolutional module called *Illumination Head*. This module extracts visual features from a single patch $\mathbf{P_{img}}$ by convoluting and downsampling. The resulting feature vector is then flattened into a sequence of tokens $f_i \in \mathbb{R}^{N \times c}$ with $c = 512$ and $N = \frac{p}{8} * \frac{p}{8}$. Using a transformer encoder-decoder structure, IllTr encodes the global relationship between the features f_i and generates global-aware representations before decoding them to a low-resolution prediction $f_j \in \mathbb{R}^{\frac{p}{8} \times \frac{p}{8} \times c}$. Finally, a learnable module called *Illumination Tail* upsamples the low-resolution features f_j to generate the final high-resolution patch prediction $f_k \in \mathbb{R}^{p \times p \times 3}$. For more details see the original work by Feng et al. [21].

Our architecture processes the input image similarly to IllTr in patches of size $p \times p$ before stitching them together in the end. In contrast to the prior work, we have a priori visual information in templates $\mathbf{T}$ available. To exploit this information, we propose two different variants:

1. The template $\mathbf{T}$ is scaled to a fixed size of $p \times p$ pixels. We refer to this template representation as $\mathbf{TP_{full}}$. Since $\mathbf{TP_{full}}$ is created using the full template, it contains all low-frequency signals but misses the fine-grained details.

2. We crop a window of $(p + 2m) \times (p + 2m)$ pixels from the template $\mathbf{T}$ for a margin of m pixels such that the $p \times p$ center region corresponds to the image patch region $\mathbf{P_{img}}$. We then scale the cropped window to our fixed size of $p \times p$ pixels. We refer to the scaled patch as $\mathbf{TP_{pad=m}}$. This cropping method captures the local template context for a given image patch. Since the alignment of $\mathbf{I_{dwp}}$ and $\mathbf{T}$ is not pixel-wise correct due to the imperfect dewarping $\mathbf{B}$, cropping at the exact same coordinates might not contain the relevant visual features for the given image patch. We introduced the margin m to tackle this problem.

Each variant encodes information about the visual template structure in a $p \times p$ patch referred to as $\mathbf{TP_x}$. See Figure 12.1 for a visualization of the patch extraction variants.

Given an image patch $\mathbf{P_{img}}$ and a template patch $\mathbf{TP_x}$, we apply one independent *Illumination Head* per patch, which yields two sequences of features $f_i \in \mathbb{R}^{N \times c}$ and $t_i \in \mathbb{R}^{N \times c}$. We concatenate the sequences f_i and t_i to $c_i \in \mathbb{R}^{2N \times c}$ before applying the same encoder structure as IllTr. This allows the model to attend between the image features, as well as cross relations between image and template features. That way, the model is conceptually capable of integrating prior visual cues provided by the template image in the intermediate feature representation.

Before applying the decoder module, as in IllTr, we discard half of the intermediate features from the encoder since we are only interested in predicting a single image patch. Finally, we apply the *Illumination Tail* similar to IllTr to upsample the output features.

As for the loss function, we adhere to the original and combine the L1 loss with the perceptual loss, also known as VGG loss, [76] using a weighting factor λ.

$$\mathcal{L}_{total} = \mathcal{L}_1 + \lambda \mathcal{L}_{VGG} \tag{12.1}$$

12.2 Evaluation

In this section, we explain how we train and evaluate our model. First, we discuss the datasets we use, as described in Section 12.2.1. Then, we review the evaluation metrics we employ, which are explained in Section 12.2.2. Finally, we provide details on the implementation of our approach, as outlined in Section 12.2.3.

12.2.1 Datasets

In the following, we present both our training and evaluation dataset.

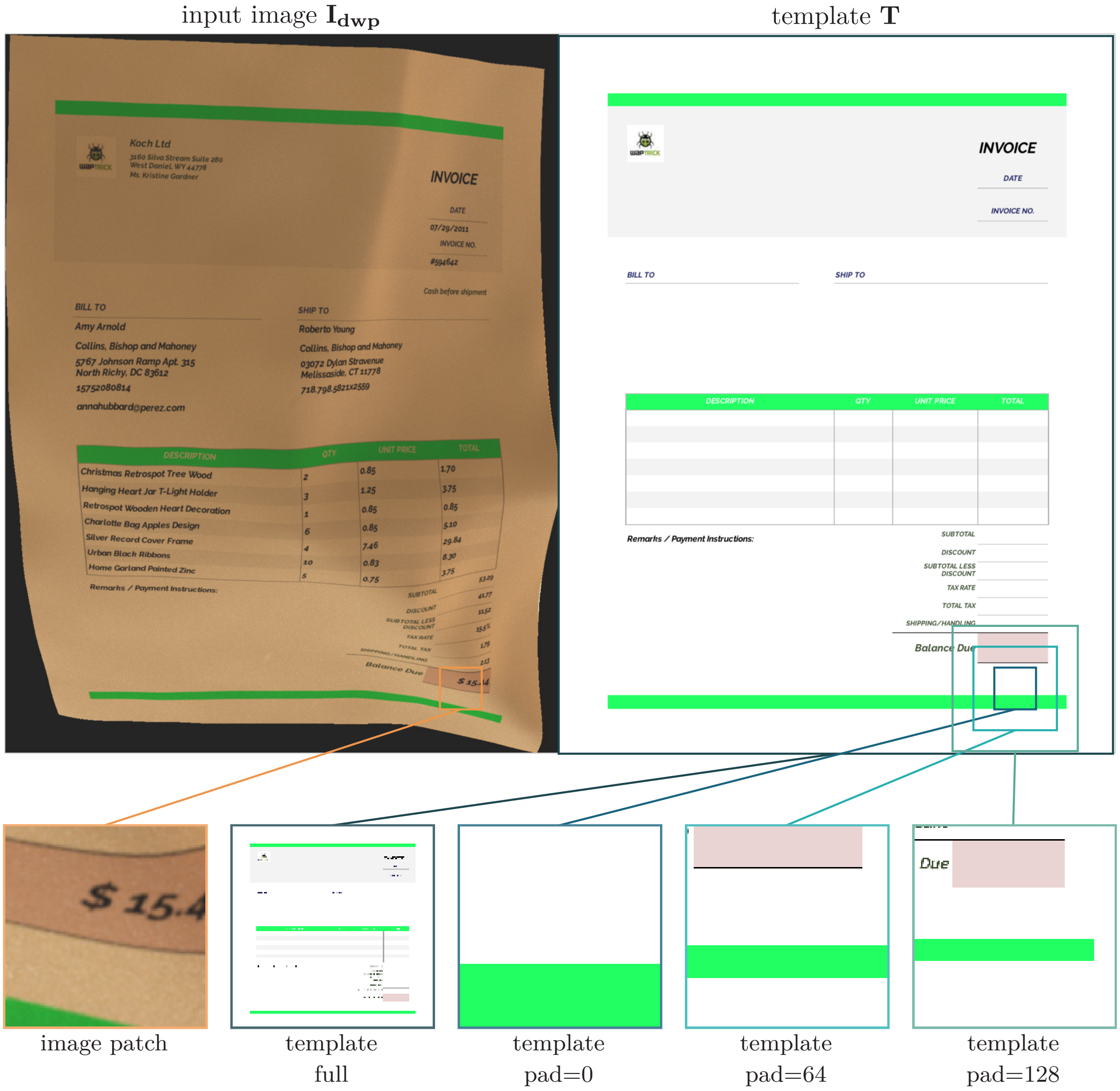


Figure 12.1: Visualization of a single image patch and the corresponding template patches for the different variants. The graphic shows three template patches for the padded crop variant with margin m = {0, 64, 128}, respectively.

12.2.1.1 Training Dataset

We train the models on our Inv3D dataset (see Chapter 5). In Inv3D, each sample has its distinct template, leading to a one-to-one match of warped images and templates during training. Since the dataset contains fully warped document images and our target domain is partially dewarped document images, we apply the ground truth backward transformation $\hat{\mathbf{B}}$ partially to the warped image $\mathbf{W}$ and the warped albedo map $\mathbf{A}$. The parameter $\alpha \in [0, 1]$ scales the amplitude of the backward map. $\hat{\mathbf{B}}_0(\mathbf{W})$ corresponds to the input image $\mathbf{W}$ and $\hat{\mathbf{B}}_1(\mathbf{W})$ equates to the perfectly dewarped document image containing solely illumination effects. See Figure 12.2 for an example of various dewarping progressions. Note that we explicitly use random values α drawn from a uniform distribution between 0 and 1 during training to simulate the imperfect geometric dewarping from the preceding dewarping stage instead of using perfectly dewarped documents. Training on partially dewarped documents is crucial to ensure the model's robustness to imperfect geometric dewarping.

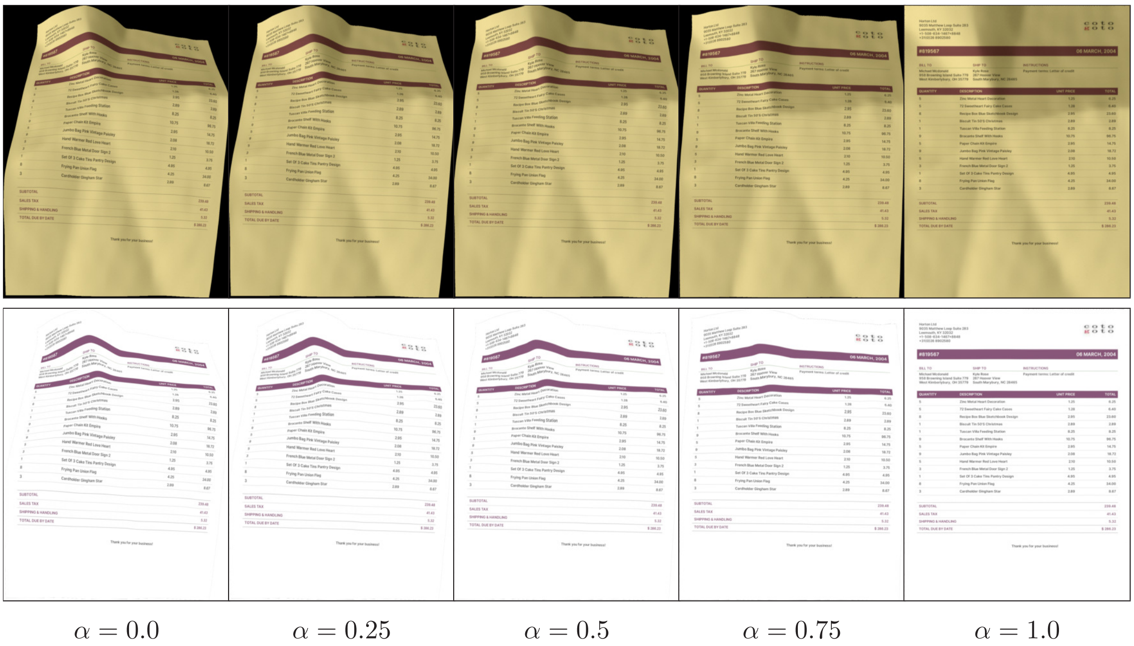


Figure 12.2: Depiction of the Inv3D dataset with partial dewarpings $\hat{\mathbf{B}}_\alpha$. The upper row shows the input images, while the lower row illustrates the corresponding ground truth illumination-corrected counterparts.

12.2.1.2 Evaluation Dataset

For evaluation, we use the real-world dataset Inv3DReal (see Chapter 6) and geometrically dewarp it using our model GeoTrTemplateLarge (see Chapter 8). This way, we can evaluate our models in a realistic setting, as our models are intended to be applied after the geometric dewarping step. We refer to the dewarped dataset as Inv3DRealDewarp in the following.

Additionally, we use the test split of our synthetic dataset Inv3D to conduct ablation studies, as it offers a more controlled environment for evaluation. In the following, we refer to this dataset as Inv3DTest.

12.2.2 Evaluation Metrics

All metrics employed compare the model output $\mathbf{I}_{\text{ill}}$ with a reference image of identical resolution. For the synthetic evaluations, the reference image is the identically warped ground truth albedo image $\hat{\mathbf{I}}$. Due to the absence of a ground truth backward map for the real-world dataset Inv3DRealDewarp, we cannot compare the model output against a pixel-aligned and perfectly illuminated image. Instead, we evaluate the model's performance by comparing it to the ground truth flatbed image, which serves as the closest available approximation. All images have a resolution of 2200 × 1700 pixels.

We assess the models using four metrics, namely *MS-SSIM* [90], *LPIPS* [104], *ED*, and *CER*. See Section 2.3 for the definitions.

12.2.3 Implementation Details

We attempt to keep the hyperparameters as close as possible to the original work IllTr [21]. The patch size p is set to 128 pixels, and the overlap between two patches to 16 pixels, similar to IllTr. The loss weight λ from Equation 12.1 is 10^{-5} and we set the batch size to 24. For training, we employed the AdamW optimizer [57] with an initial learning rate of 10^{-4} and a StepLR scheduler[1] with a step size of 20 and a gamma of 0.3. Contrary to IllTr, we do not stop training after 35 epochs and continue it until there is no further improvement for 25 continuous epochs measured by the loss $\mathcal{L}_{total}$ in Equation 12.1 on the validation data split. To improve the resilience to different lighting variations, we employ a random color jitter during training with random brightness, contrast, saturation, and hue.

12.3 Results

In this section, we present the results of our experiments. The sections 12.3.1 and 12.3.2 compare our models with the state-of-the-art quantitatively and qualitatively, respectively. Section 12.3.3 presents three ablation studies for detailed insights into the best-performing model.

12.3.1 Quantitative Results

Table 12.1 lists the quantitative results of our approach in comparison to the state-of-the-art model IllTr [21] and the identity baseline. First, we observe that all models trained on Inv3D outperform the baseline in all metrics. The IllTr model trained on DocProj yields a lower *MS-SSIM* value than the baseline method and, thus, indicates a degradation in visual similarity. When comparing IllTr trained on DocProj [46] with the same model trained on Inv3D, it is apparent that the training on Inv3D is superior in all metrics. This could be attributed to the smaller domain gap for Inv3D to our evaluation dataset and the training process with partially dewarped documents. When comparing IllTr to our model IllTrTemplate, we find that IllTrTemplate surpasses IllTr in all variants and metrics. Within our four variants, there is no clear best model based on the set of all metrics. The visual metrics *MS-SSIM* and *LPIPS* indicate that a padding of 128 pixels works best, while the text metrics *ED* and *CER* favor a padding of 0 pixels as the most favorable choice. With an *LPIPS* of 0.221, the variant with 128 pixel padding achieves a 15 % relative improvement in contrast to the original model Inv3D trained on the same data. Thus, this model is recommended for document archival and retrieval. For the text metrics, the variant with 0 pixels padding achieves a relative improvement of 6.3 % for *ED* and *CER* and is therefore beneficial for information extraction.

12.3.2 Qualitative Results

Figure 12.3 shows randomly selected images from the evaluation dataset Inv3DRealDewarp. Looking at the illumination correction results, we observe that all models, IllTr and IllTrTemplate, generate patchy artifacts to some degree. More precisely, the individual patches do not always agree on a common background color, which leads to visible patches within the stitched image.

[1] `https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html`

Model	Template	Train Dataset	↑*MS-SSIM*	↓*LPIPS*	↓*ED*	↓*CER*
Identity	—	—	0.711 (0.094)	0.324 (0.111)	329.1 (184.6)	0.512 (0.264)
IllTr [21]	—	DocProj [46]	0.651 (0.133)	0.306 (0.118)	306.4 (151.2)	0.477 (0.213)
IllTr [21]	—	Inv3D [31]	0.718 (0.154)	0.260 (0.095)	264.6 (161.0)	0.412 (0.229)
IllTrTemplate (ours)	full	Inv3D [31]	0.736 (0.109)	0.234 (0.086)	257.9 (159.7)	0.402 (0.227)
IllTrTemplate (ours)	pad=0	Inv3D [31]	0.731 (0.107)	0.231 (0.085)	**247.9 (156.3)**	**0.386 (0.221)**
IllTrTemplate (ours)	pad=64	Inv3D [31]	0.760 (0.144)	0.226 (0.082)	251.4 (161.7)	0.391 (0.226)
IllTrTemplate (ours)	pad=128	Inv3D [31]	**0.762 (0.137)**	**0.221 (0.082)**	251.3 (159.8)	0.392 (0.227)

Table 12.1: Evaluation of our model IllTrTemplate on the Inv3DRealDewarp dataset. Values in brackets denote the standard deviation across all test samples.

This finding is likely due to the independent illumination correction for each patch before stitching them together in the end. The illumination corrections of IllTr trained on DocProj [46] (column (b)) show stronger artifacts around the shadow borders than those trained on Inv3D, which indicates a lack of hard shadows in the DocProj dataset.

The comparison of IllTr and IllTrTemplate demonstrates the effectiveness of adding template information for color reconstruction. All IllTrTemplate variants seem to incorporate the original colors provided by the templates in their illumination correction output. Thus, the reconstructed images appear to be more similar to the original. Note, since no ground truth backward mappings $\hat{\mathbf{B}}$ are available for the real-world dataset Inv3DReal, the reference image does not contain any warping.

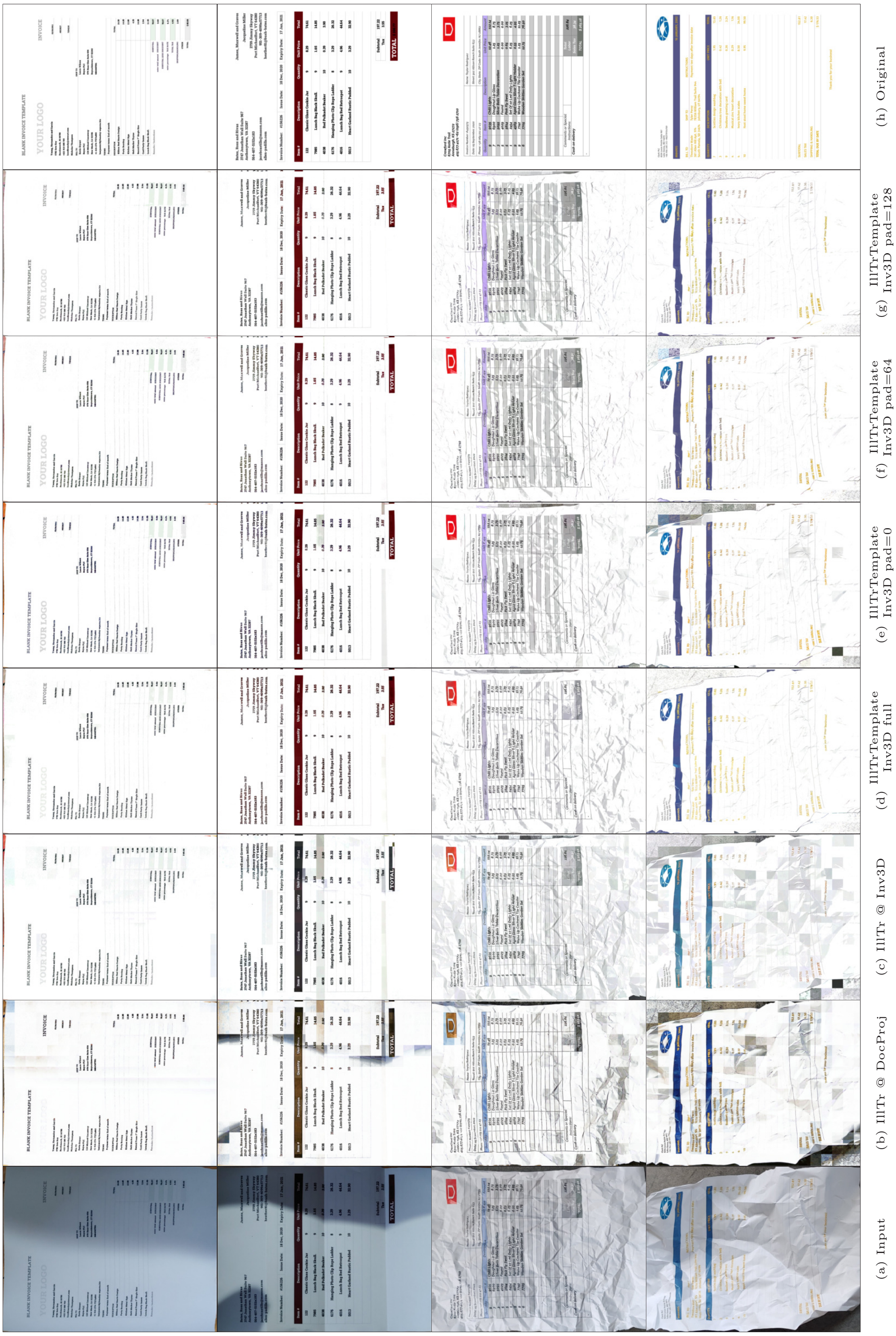
When considering the last two rows, we observe that all models struggle with removing the fine-grained creases. This is likely caused by the domain gap between the synthetically generated dataset Inv3D and the real-world evaluation dataset Inv3DReal.

12.3.3 Ablation Studies

In the following, we conduct a series of ablation studies. We consider only the model IllTrTemplate with a padding of 128 pixels as it is the best-performing model according to the *LPIPS* metric.

12.3.3.1 Ablation 1: Categorization

We split the Inv3DRealDewarp dataset samples into their different categories depending on the type of document sheet modification and environment setting during recording. Table 12.2 shows the results for IllTrTemplate with 128 pixels padding trained on Inv3D and evaluated on Inv3DRealDewarp. For the document modification type, we observe that *crumpleseasy* improves most according to all metrics. *Crumpleshard* seems to be the hardest modification type, which coincides with the qualitative findings that hard creases are not corrected properly. When considering the dataset split by environment setting, it becomes apparent that the majority of metrics, except for *MS-SSIM*, collectively affirm that the *color* environment is comparatively less challenging, whereas the *shadow* setting poses the greatest difficulty. The latter also aligns with the observations of the qualitative analysis, wherein the presence of harsh shadows resulted in the generation of more pronounced artifacts.



(a) Input

(b) IllTr @ DocProj

(c) IllTr @ Inv3D

(d) IllTrTemplate Inv3D full

(e) IllTrTemplate Inv3D pad=0

(f) IllTrTemplate Inv3D pad=64

(g) IllTrTemplate Inv3D pad=128

(h) Original

Figure 12.3: Qualitative results of state-of-the-art IllTr [21] and our model IllTrTemplate. The samples were drawn randomly from Inv3DRealDewarp. The left column shows the input images $\mathbf{I_{dwp}}$. The rightmost column shows the optimal image $\hat{\mathbf{B}}(\mathbf{A})$. The center columns depict the illumination corrected images per model $\mathbf{B}(\mathbf{W})$.

Model	↑*MS-SSIM*	↓*LPIPS*	↓*ED*	↓*CER*
perspective	0.770 (0.153)	0.210 (0.089)	244.9 (163.9)	0.381 (0.227)
curled	0.768 (0.124)	0.199 (0.073)	239.9 (175.4)	0.380 (0.264)
fewfold	0.770 (0.139)	0.209 (0.070)	259.4 (170.4)	0.402 (0.233)
multifold	0.757 (0.151)	0.230 (0.088)	256.9 (157.5)	0.398 (0.221)
crumpleseasy	**0.797 (0.115)**	**0.190 (0.055)**	**225.5 (166.0)**	**0.352 (0.235)**
crumpleshard	0.711 (0.124)	0.289 (0.075)	281.2 (120.7)	0.440 (0.172)
bright	0.751 (0.142)	0.221 (0.088)	251.6 (163.9)	0.392 (0.230)
color	0.766 (0.137)	**0.213 (0.083)**	**229.1 (163.0)**	**0.355 (0.225)**
shadow	**0.770 (0.131)**	0.230 (0.075)	273.3 (150.5)	0.430 (0.222)

Table 12.2: Ablation 1: The Inv3DRealDewarp dataset is partitioned into categories based on their respective modifications (upper part) and environment settings (lower part). The depicted results have been generated by our model IllTrTemplate with a padding of 128 pixels. Values in brackets denote the standard deviation within each category.

12.3.3.2 Ablation 2: Dewarping Importance

To gain insights into the importance of the quality of the preceding geometric dewarping step on the illumination correction, we evaluate the test split of Inv3D with varying degrees of dewarping. See Figure 12.2 for an example of various dewarping progressions.

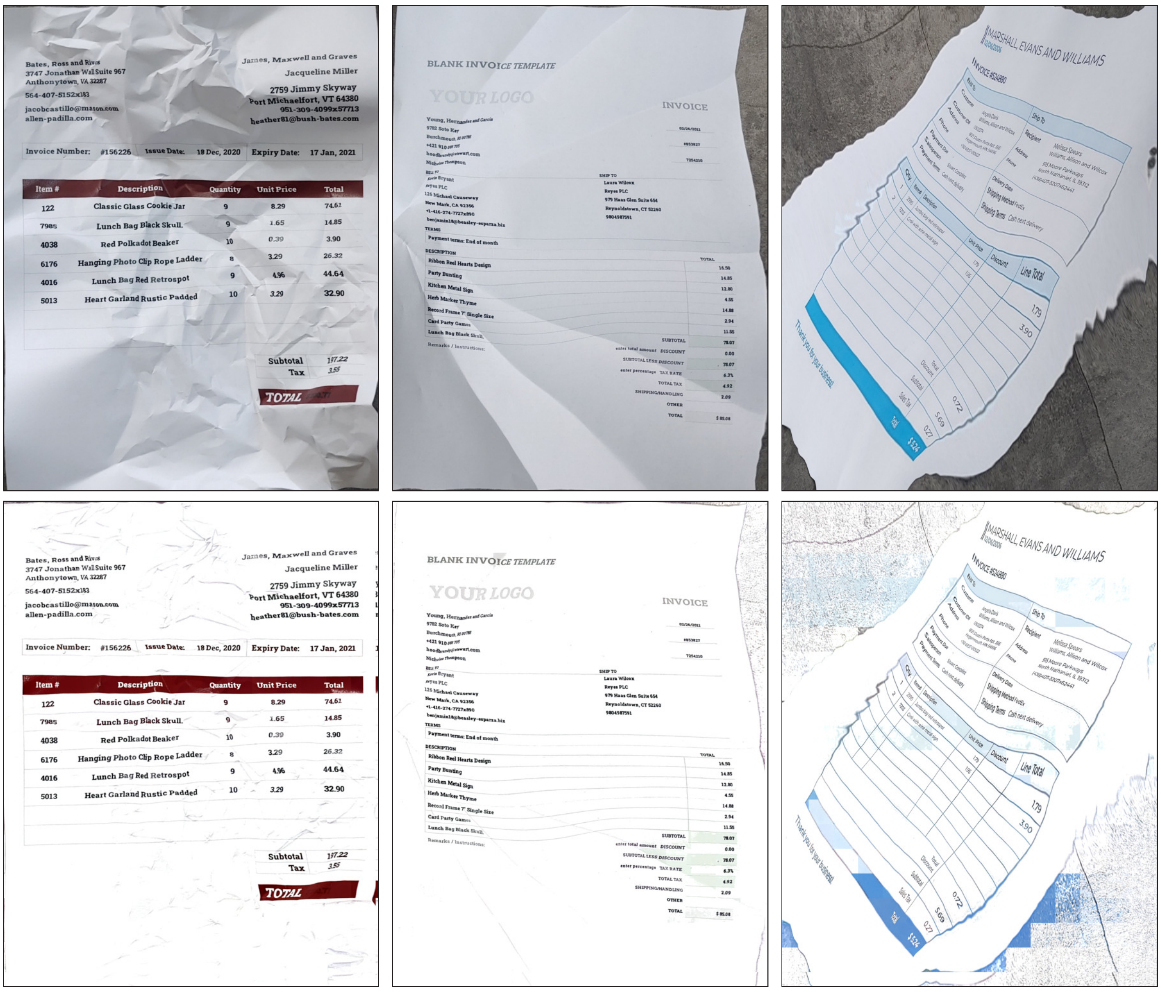
Table 12.3 shows the results of this ablation study. The absolute values of the visual metrics *MS-SSIM* and *LPIPS* exhibit a remarkable closeness to their respective optimum. This indicates the near-perfect illumination correction of the test split of Inv3D. Meanwhile, the text metrics *ED* and *CER* continue to exhibit considerably high values. This implies an imprecise reconstruction of high-frequent signals within the image since the fine-grained details are crucial for text recognition. In all metrics except for *MS-SSIM*, the best results were achieved using the perfect geometric dewarping with $\alpha = 1$. Since the visual metrics are already near their optimum, there is no steep decrease in performance when considering $\alpha < 1$. Note that the remarkably high *CER* values for low α values are due to the limited *OCR* performance of Tesseract in the reference image $\hat{\mathbf{I}}^{\alpha}$.

Dewarp factor α	↑*MS-SSIM*	↓*LPIPS*	↓*ED*	↓*CER*
0.0 (fully warped)	**0.992 (0.038)**	0.058 (0.019)	202.8 (149.2)	2.725 (17.091)
0.2	0.981 (0.016)	0.046 (0.019)	207.9 (145.4)	3.952 (21.332)
0.4	0.979 (0.015)	0.045 (0.020)	216.4 (137.4)	1.637 (7.349)
0.6	0.978 (0.014)	0.043 (0.021)	220.2 (150.2)	1.245 (7.544)
0.8	0.978 (0.014)	0.040 (0.022)	212.0 (158.8)	0.673 (3.270)
1.0 (fully dewarped)	0.982 (0.014)	**0.030 (0.026)**	**194.9 (171.9)**	**0.472 (1.104)**

Table 12.3: Ablation 2: We investigate the importance of the dewarp factor α. All results were obtained by our model IllTrTemplate with 128 pixel padding on a subset of 360 samples of Inv3DTest. Values in brackets denote the standard deviation across all test samples.

12.3.3.3 Ablation 3: Error Distribution

In a third ablation study, we examine the error distribution over all samples in the evaluation dataset Inv3DRealDewarp. Since the imperfections in the geometric dewarping step affect the illumination step, we classified all input samples in three categories *severely flawed*, *flawed*, and *moderate* depending on the severity of the dewarping errors. Images in the first category contain large areas showing the background instead of the document. In the *flawed* category, there is less background visible, but the document still does not cover the entire image. The last category includes all samples where, at minimum, the outline has been accurately mapped. Figure 12.4 shows examples of the categories.



Moderate Flawed Severely flawed

Figure 12.4: Samples of Inv3DRealDewarp with the highest *LPIPS* value per imperfection category for IllTrTemplate with 128 pixel padding. The upper row depicts input images, and lower row shows the results after illumination correction.

Figure 12.5 plots the distribution of *LPIPS* values over the samples of Inv3DRealDewarp after the illumination correction. The histogram shows that severely flawed geometric dewarpings give the highest metric values. Since our IllTrTemplate model solely corrects the illumination and does not complete the partial geometric dewarping, this finding is to be expected.

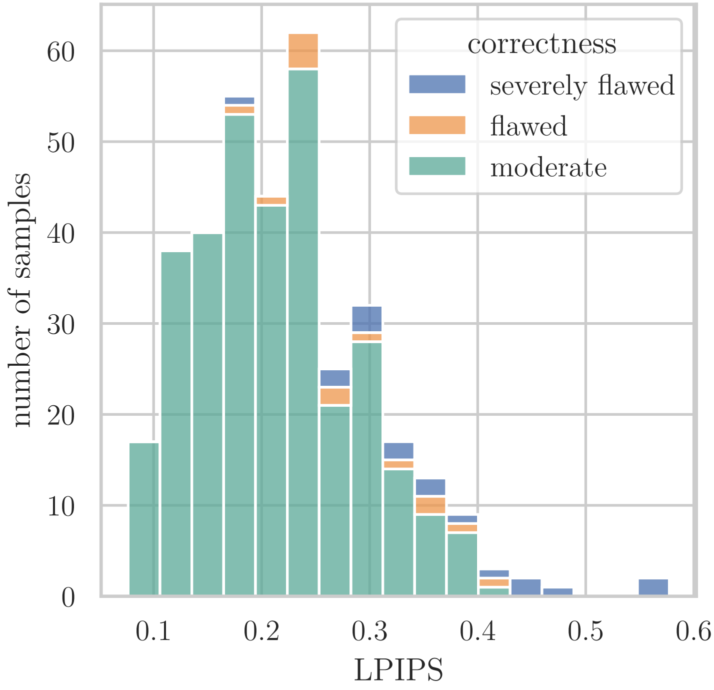


Figure 12.5: Distribution of the *LPIPS* error over all Inv3DRealDewarp samples given by the IllTrTemplate model with a padding of 128 pixels. All samples are classified depending on the severity of the dewarping errors.

12.4 Summary and Discussion

In this chapter, we tackled the problem of illumination correction for imperfectly dewarped document images using reference templates. This question was formulated in RQ3:

> RQ3: Illumination Correction
>
> Given the partially dewarped document images, how can we **correct the illumination** to improve the quality of the document images?

We presented two methods for incorporating the template information using a transformer encoder-decoder architecture. To evaluate the effectiveness of additional template images, we conducted a comparative analysis against the state-of-the-art model IllTr [21]. We assessed a total of four new template-based models, specifically the full template model and the cropped template models with paddings of 0, 64, and 128 pixels. We measured the performance using multiple metrics and observed a relative improvement of 15 % *LPIPS* and 6.3 % *CER* error compared to IllTr. A series of ablation studies were conducted that revealed a domain gap between the synthetically generated dataset, Inv3D, and the evaluation dataset, Inv3DRealDewarp.

With the proposed model IllTrTemplate, we can answer the RQ3. All four variants surpass the performance of the previous state-of-the-art model IllTr [21]. Thus, we found a method to improve illumination correction for partially dewarped document images with the help of reference templates.

Part VI

Synthesis

13 Conclusion and Outlook

In this thesis, we investigated document image enhancement methods using reference templates. Specifically, we addressed the tasks of geometric dewarping and illumination correction, i.e., the removal of pixel distortions and correction of illumination changes induced by the camera-based image-capturing process. To this end, we created a training and evaluation dataset in Part III and conducted an investigation on geometric dewarping in Part IV. In Part V, we tackled the problem of illumination correction under the assumption of imperfect geometric dewarping. In this chapter, we summarize and discuss the thesis (Section 13.1) before concluding the work with an outlook over future work in Section 13.2.

13.1 Summary and Discussion

Since business workflows in many countries still rely on printed document sheets, there is a need to digitize them in order to transition from a physical to a digital workflow. Example domains are the retail industry, insurance sector, healthcare system, etc. In order to digitize paper sheets, there are three fundamental ways: (1) human labor, (2) scanners combined with AI, and (3) cameras combined with AI. The third option, cameras combined with AI, is the least expensive option and offers great flexibility at the same time. However, using a camera as the image capturing method introduces inherent challenges in AI processing and interpretation compared to the scanner-based approach. These challenges comprise camera and paper deformations, as well as external lighting influences. In this thesis, we explored methods to overcome these challenges by transforming camera-captured images into scan-like images, enabling more effective downstream analysis of the documents.

Our approach to improving the automated geometric dewarping and illumination correction capabilities is based on the key idea that the human mind leverages prior knowledge about the expected document structure to perform these tasks. In order to provide additional information to a machine learning model, we introduced the concept of reference templates, i.e., RGB images displaying the expected structure of the document without any instance-specific information. We formulated our hypothesis as follows:

Hypothesis: Reference Templates

Additional information about the expected document structure and visual appearance can be leveraged to improve the document image enhancement process.

We structured our research into six parts: (1) introduction (2) preliminaries, (3) data generation, (4) geometric dewarping, (5) illumination correction, and (6) synthesis.

Part I introduced the topic by discussing the motivation and the research goal first. We then analyzed the challenges in the research topic before presenting the key idea and hypothesis. Furthermore, we presented and explained the research questions, followed by an overview of the contributions made in this work.

Part II presented the preliminaries of our work, which include the foundational concepts and related research necessary for understanding our work. The foundations are divided into three sections: First, we discuss the various methods of coordinate transformations employed. Second, we outline the deep learning architectures used throughout this work. Third, we provide details on the evaluation metrics established in prior research. In the related work, we provide an overview of the research area grouped into three subareas: available datasets, geometric dewarping, and illumination correction. Furthermore, we set the prior work in context with our contributions and discuss the value added.

Part III focuses on the problem of the unavailability of a suitable dataset for this research (RQ1). We formalize the requirements for a dataset for geometric dewarping and illumination correction using reference templates. Based on these requirements, we present our approach to generating a synthetic, large-scale, and high-resolution training dataset called Inv3D. It comprises three steps: base template generation, instance generation, and instance warping. In addition, we present a second dataset, Inv3DReal, which is used for evaluation. Both datasets, as well as the generation code, are publicly available.[1]

Part IV investigates the problem of correcting geometric distortions in document images. The document image-capturing process using a camera introduces geometric distortions due to the camera's projection and the physical deformations of the document sheet. We explored two methods for document image dewarping leveraging reference templates: an implicit (RQ2.1) approach and an explicit (RQ2.3) approach. We proposed a model called GeoTrTemplate, which utilizes an attention mechanism to determine how to leverage the information within the reference templates and showed that it outperforms all previous dewarping models without reference templates. The second approach explicitly exploits the knowledge of the reference templates by extracting visual and textual lines from the input images and the template images, and thereafter matches both sets of lines in order to infer the geometric dewarping map. By using explicit representations in intermediate steps, this approach decomposes the complex task of geometric dewarping into a series of simpler tasks. Our new model, DocMatcher, outperforms all previous models without reference templates, as well as our first model, GeoTrTemplate, across all metrics by a significant margin. We published the code and weights for both models online.[2]

Finally, we answered the question of how to evaluate the quality of geometric dewarping approaches with regard to text readability and positional awareness (RQ2.2). Since all previously employed metrics either suffer from the text linearization problem or the insensitivity of text readability, we proposed a novel metric called *mnCER*, which addresses both problems.

[1] Datasets: `https://publikationen.bibliothek.kit.edu/1000161884`
Code: `https://github.com/FelixHertlein/inv3d-generator`

[2] GeoTrTemplate: `https://github.com/FelixHertlein/inv3d-model`
DocMatcher: `https://felixhertlein.github.io/doc-matcher`

Part V is concerned with the correction of the illumination in document images, assuming they have been partially dewarped beforehand (RQ3). The document capturing process using a camera leads to undesirable illumination artifacts like ambient light and shadows, which need to be removed to improve text readability using *AI*. We introduced a new model, IllTrTemplate, designed to integrate reference template information using a transformer encoder-decoder architecture. To explore different approaches for integrating reference templates, we proposed four variants of IllTrTemplate. We show that our proposed model improves the automated text readability compared to the previous state-of-the-art model IllTr, which does not leverage template information. Our code and model weights are available online.[3]

In conclusion, our hypothesis on the benefit of reference templates for geometric dewarping and illumination correction of document images is confirmed. We presented several methods for both tasks, each improving its respective task by utilizing reference templates. Our experiments show that adding information about the expected document structure and visual appearance can be leveraged to improve the document image enhancement process.

13.2 Future Work

Our research has shown that reference templates can be beneficial for document image dewarping and illumination correction. In future research, the information provided in the reference templates could be leveraged not only to improve the enhancement of document images but also for downstream tasks such as information extraction. Given the knowledge about the expected structure of a document, one could exploit it by creating a semantically biased *OCR* engine for highly accurate text detection. For example, given the information that a date is expected in a specific area, a semantically biased *OCR* approach could reject invalid dates such as the 71st day of a month and instead parse it as the next likely one, i.e., the 11th. Thus, using the reference template information might improve the robustness of information extraction systems.

Another research direction could focus on the applicability of document image enhancement systems using reference templates. For now, a potential user has to select the template associated with a given document from a pool of available document types by hand. A machine learning model could automate this task by selecting the correct template given the warped document image as input. This would increase the usability of template-based approaches, as it provides a faster and more convenient experience for the user.

As stated in the introduction (Section 1.1), we focused in this work on the challenges *Camera Distortions*, *Paper Distortions*, and *Lighting*. In addition to these challenges, further influences could affect the document sheet and image-capturing process. These include partially destroyed documents, stains of additional substances, e.g., coffee, and blur while image capturing. Future work could investigate whether the current models can handle these additional influences sufficiently well. In case the results are not sufficient, there is a need for more research in this direction.

Lastly, we want to highlight the potential for integrating multimodal foundation models like CLIP [72] in the image dewarping and illumination correction process. Their implicit general knowledge of documents and (limited) logical reasoning might benefit geometric dewarping and

[3] IllTrTemplate: `https://github.com/FelixHertlein/illtrtemplate-model`

illumination correction. Since these models are prone to hallucination, they are unsuitable for direct information extraction. Nevertheless, one could leverage their general knowledge to generate a better backward map, dewarp the image, and then use *OCR* for the text extraction. Since the foundation model is used exclusively for backward map generation and not text extraction, the risk of hallucination is significantly reduced.

List of Figures

List of Tables

List of Abbreviations

A

AE *Auto Encoder* 30, 33, 35
AI *Artificial Intelligence* 3, 4, 6, 39, 75, 109, 111

C

CER *Character Error* Rate 9, 28, 64, 66–68, 71, 72, 82–84, 87, 88, 100–102, 104, 106
CGAN *Conditional Generative Adversarial Network* 30, 33, 35
CNN *Convolutional Neural Network* 24, 30, 31, 33–36
ConvGRU *Convolutional Gated Recurrent, Network* 30

D

DIB *Document Image Binarization* 34
DIE *Document Image Enhancement* 34–36
DL *Deep Learning* 3, 4, 26, 45

E

ED *Edit Distance* 28, 64, 66–68, 71, 82–84, 87, 88, 100–102, 104

F

FPN *Feature Pyramid Network* 30

G

GAN *Generative Adversarial Network* 34, 35
GRU *Gated Recurrent Unit* 32

I

IE *Information Extraction* 4
IoU *Intersection over Union* 72

L

LD *Local Distortion* 9, 27, 28, 64, 66–68, 71, 82–89
LPIPS *Learned Perceptual Image Patch Similarity* 9, 27, 28, 64, 66–68, 71, 82–84, 87, 88, 100–102, 104–106, 114

M

MLP *Multi-Layer Perceptron* 30, 80
mnCER *matched normalized Character Error Rate* 9, 10, 71–74, 82–84, 86–89, 110, 113
MS-SSIM *Multi-Scale Structural Similarity* 27, 28, 64, 66–68, 71, 82–84, 87, 88, 100–102, 104

N

nCER *normalized Character Error Rate* 72, 73
NLP *Natural Language Processing* 21

O

OCR *Optical Character Recognition* 27, 28, 66, 72, 80, 82, 84, 104, 111, 112

S

SAM *Segment Anything* 21, 24, 25
SIFT *Scale-Invariant Feature Transform* 27
SSIM *Structural Similarity* 27

V

ViT *Vision Transformer* 21, 23, 24

Bibliography

[1] Lightning AI. *Character Error Rate*. 2024. URL: `https://lightning.ai/docs/torchmetrics/stable/text/char_error_rate.html` (visited on 12/22/2024).

[2] Anas M. Ali; Bilel Benjdira; Anis Koubaa; Walid El Shafai; Zahid Khan; Wadii Boulila. "Vision Transformers in Image Restoration: A Survey". In: *Sensors* 23.5 (2023), p. 2385. DOI: `10.3390/S23052385`.

[3] Marcos Martins de Almeida; Rafael Dueire Lins; Rodrigo Barros Bernardino; Darlisson Marinho de Jesus; Bruno Lima. "A New Binarization Algorithm for Historical Documents". In: *Journal of Imaging* 4.2 (2018), p. 27. DOI: `10.3390/JIMAGING4020027`.

[4] Hmrishav Bandyopadhyay; Tanmoy Dasgupta; Nibaran Das; Mita Nasipuri. "A Gated and Bifurcated Stacked U-Net Module for Document Image Dewarping". In: *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 10548–10554. DOI: `10.1109/ICPR48806.2021.9413001`.

[5] Hmrishav Bandyopadhyay; Tanmoy Dasgupta; Nibaran Das; Mita Nasipuri. "RectiNet-v2: A stacked network architecture for document image dewarping". In: *Pattern Recognition Letters* 155 (2022), pp. 41–47. DOI: `10.1016/J.PATREC.2022.01.014`.

[6] Derek Bradley; Gerhard Roth. "Adaptive Thresholding using the Integral Image". In: *Journal of Graphics Tools* 12.2 (2007), pp. 13–21. DOI: `10.1080/2151237X.2007.10129236`.

[7] Alexander Burden; Melissa Cote; Alexandra Branzan Albu. "Rectification of Camera-Captured Document Images with Mixed Contents and Varied Layouts". In: *Proceedings of the 16th Conference on Computer and Robot Vision*. IEEE, 2019, pp. 33–40. DOI: `10.1109/CRV.2019.00013`.

[8] Jorge Calvo-Zaragoza; Antonio Javier Gallego. "A selectional auto-encoder approach for document image binarization". In: *Pattern Recognition* 86 (2019), pp. 37–47. DOI: `10.1016/J.PATCOG.2018.08.011`.

[9] John F. Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (1986), pp. 679–698. DOI: `10.1109/TPAMI.1986.4767851`.

[10] Daqing Chen. *E-Commerce Data*. 2017. URL: `https://www.kaggle.com/carrie1/ecommerce-data` (visited on 04/11/2022).

[11] Siu-Wing Cheng; Tamal K. Dey; Jonathan Richard Shewchuk. *Delaunay Mesh Generation*. Chapman and Hall / CRC computer and information science series. CRC Press, 2013. ISBN: 978-1-584-88730-0.

[12] Kyunghyun Cho; Bart van Merrienboer; Çaglar Gülçehre; Dzmitry Bahdanau; Fethi Bougares; Holger Schwenk; Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Ed. by Alessandro Moschitti; Bo Pang; Walter Daelemans. ACL, 2014, pp. 1724–1734. DOI: `10.3115/V1/D14-1179`.

[13] Mircea Cimpoi; Subhransu Maji; Iasonas Kokkinos; Sammy Mohamed; Andrea Vedaldi. "Describing Textures in the Wild". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2014, pp. 3606–3613. DOI: `10.1109/CVPR.2014.461`.

[14] Beiya Dai; Xing li; Qunyi Xie; Yulin Li; Xiameng Qin; Chengquan Zhang; Kun Yao; Junyu Han. "MataDoc: Margin and Text Aware Document Dewarping for Arbitrary Boundary". In: *arXiv* (2023). DOI: 10.48550/ARXIV.2307.12571.

[15] Sagnik Das; Ke Ma; Zhixin Shu; Dimitris Samaras. "Learning an Isometric Surface Parameterization for Texture Unwrapping". In: *Proceedings of the 17th European Conference on Computer Vision*. Ed. by Shai Avidan; Gabriel J. Brostow; Moustapha Cissé; Giovanni Maria Farinella; Tal Hassner. Vol. 13697. Lecture Notes in Computer Science. Springer, 2022, pp. 580–597. DOI: 10.1007/978-3-031-19836-6_33.

[16] Sagnik Das; Ke Ma; Zhixin Shu; Dimitris Samaras; Roy Shilkrot. "DewarpNet: Single-Image Document Unwarping With Stacked 3D and 2D Regression Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 131–140. DOI: 10.1109/ICCV.2019.00022.

[17] Sagnik Das; Hassan A. Sial; Ke Ma; Ramón Baldrich; María Vanrell; Dimitris Samaras. "Intrinsic Decomposition of Document Images In-the-Wild". In: *Proceedings of the 31st British Machine Vision Conference*. BMVA Press, 2020.

[18] Sagnik Das; Kunwar Yashraj Singh; Jon Wu; Erhan Bas; Vijay Mahadevan; Rahul Bhotika; Dimitris Samaras. "End-to-end Piece-wise Unwarping of Document Images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2021, pp. 4248–4257. DOI: 10.1109/ICCV48922.2021.00423.

[19] Alexey Dosovitskiy; Lucas Beyer; Alexander Kolesnikov; Dirk Weissenborn; Xiaohua Zhai; Thomas Unterthiner; Mostafa Dehghani; Matthias Minderer; Georg Heigold; Sylvain Gelly; Jakob Uszkoreit; Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *Proceedings of the 9th International Conference on Learning Representations*. OpenReview.net, 2021.

[20] Hao Feng; Shaokai Liu; Jiajun Deng; Wengang Zhou; Houqiang Li. "Deep Unrestricted Document Image Rectification". In: *IEEE Transactions on Multimedia* 26 (2024), pp. 6142–6154. DOI: 10.1109/TMM.2023.3347094.

[21] Hao Feng; Yuechen Wang; Wengang Zhou; Jiajun Deng; Houqiang Li. "DocTr: Document Image Transformer for Geometric Unwarping and Illumination Correction". In: *Proceedings of the 29th ACM International Conference on Multimedia*. Ed. by Heng Tao Shen; Yueting Zhuang; John R. Smith; Yang Yang; Pablo César; Florian Metze; Balakrishnan Prabhakaran. ACM, 2021, pp. 273–281. DOI: 10.1145/3474085.3475388.

[22] Hao Feng; Wengang Zhou; Jiajun Deng; Qi Tian; Houqiang Li. "DocScanner: Robust Document Image Rectification with Progressive Learning". In: *arXiv* (2021). URL: https://arxiv.org/abs/2110.14968.

[23] Hao Feng; Wengang Zhou; Jiajun Deng; Yuechen Wang; Houqiang Li. "Geometric Representation Learning for Document Image Rectification". In: *Proceedings of the 17th European Conference on Computer Vision*. Ed. by Shai Avidan; Gabriel J. Brostow; Moustapha Cissé; Giovanni Maria Farinella; Tal Hassner. Vol. 13697. Lecture Notes in Computer Science. Springer, 2022, pp. 475–492. DOI: 10.1007/978-3-031-19836-6_27.

[24] Arpan Garai; Samit Biswas; Sekhar Mandal. "A theoretical justification of warping generation for dewarping using CNN". In: *Pattern Recognition* 109 (2021), p. 107621. DOI: 10.1016/J.PATCOG.2020.107621.

[25] Marc-André Gardner; Kalyan Sunkavalli; Ersin Yumer; Xiaohui Shen; Emiliano Gambaretto; Christian Gagné; Jean-François Lalonde. “Learning to predict indoor illumination from a single image”. In: *ACM Transactions on Graphics* 36.6 (2017), 176:1–176:14. DOI: 10.1145/3130800.3130891.

[26] Sarayut Gonwirat; Olarik Surinta. “DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters”. In: *IEEE Access* 10 (2022), pp. 90133–90148. DOI: 10.1109/ACCESS.2022.3201560.

[27] Andrew Harltey; Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)* Cambridge University Press, 2006. ISBN: 978-0-521-54051-3.

[28] Kaiming He; Xiangyu Zhang; Shaoqing Ren; Jian Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[29] Federal Ministry of Health (Germany). *Federal Act on Registration*. 2024. URL: https://www.gesetze-im-internet.de/englisch_bmg/englisch_bmg.html (visited on 09/20/2024).

[30] Felix Hertlein; Alexander Naumann. “Template-guided Illumination Correction for Document Images with Imperfect Geometric Reconstruction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. IEEE, 2023, pp. 904–913. DOI: 10.1109/ICCVW60793.2023.00097.

[31] Felix Hertlein; Alexander Naumann; Patrick Philipp. “Inv3D: a high-resolution 3D invoice dataset for template-guided single-image document unwarping”. In: *International Journal on Document Analysis and Recognition* 26.3 (2023), pp. 175–186. DOI: 10.1007/S10032-023-00434-X.

[32] Felix Hertlein; Alexander Naumann; York Sure-Vetter. “DocMatcher: Document Image Dewarping via Structural and Textual Line Matching”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2025.

[33] Sonain Jamil; Md Jalil Piran; Oh-Jin Kwon. “A Comprehensive Survey of Transformers for Computer Vision”. In: *Drones* 7.5 (2023), p. 287. DOI: 10.3390/drones7050287.

[34] Xiangwei Jiang; Rujiao Long; Nan Xue; Zhibo Yang; Cong Yao; Gui-Song Xia. “Revisiting Document Image Dewarping by Grid Regularization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 4533–4542. DOI: 10.1109/CVPR52688.2022.00450.

[35] Seokjun Kang; Brian Kenji Iwana; Seiichi Uchida. “Complex image processing with less data - Document image binarization by integrating multiple pre-trained U-Net modules”. In: *Pattern Recognition* 109 (2021), p. 107577. DOI: 10.1016/J.PATCOG.2020.107577.

[36] Alexander Kirillov; Eric Mintun; Nikhila Ravi; Hanzi Mao; Chloé Rolland; Laura Gustafson; Tete Xiao; Spencer Whitehead; Alexander C. Berg; Wan-Yen Lo; Piotr Dollár; Ross B. Girshick. “Segment Anything”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2023, pp. 3992–4003. DOI: 10.1109/ICCV51070.2023.00371.

[37] Alex Krizhevsky; Ilya Sutskever; Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*. Ed. by Peter L. Bartlett; Fernando C. N. Pereira; Christopher J. C. Burges; Léon Bottou; Kilian Q. Weinberger. 2012, pp. 1106–1114.

[38] Harold W. Kuhn. “The Hungarian method for the assignment problem”. In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

[39] Pooja Kumari; Sukhendu Das. “Am I readable? Transfer learning based document image rectification”. In: *International Journal on Document Analysis and Recognition* 27.3 (2024), pp. 433–446. DOI: 10.1007/S10032-024-00476-9.

[40] Jay Lal; Aditya Mitkari; Mahesh Bhosale; David S. Doermann. “LineFormer: Line Chart Data Extraction Using Instance Segmentation”. In: *Proceedings of the International Conference on Document Analysis and Recognition.* Ed. by Gernot A. Fink; Rajiv Jain; Koichi Kise; Richard Zanibbi. Vol. 14191. Lecture Notes in Computer Science. Springer, 2023, pp. 387–400. DOI: 10.1007/978-3-031-41734-4_24.

[41] Der-Tsai Lee; Bruce J. Schachter. “Two algorithms for constructing a Delaunay triangulation”. In: *International Journal of Parallel Programming* 9.3 (1980), pp. 219–242. DOI: 10.1007/BF00977785.

[42] Kenneth Leung. *Evaluate OCR Output Quality with Character Error Rate (CER) and Word Error Rate (WER).* 2021. URL: https://medium.com/p/853175297510 (visited on 03/15/2024).

[43] Vladimir I. Levenshtein. “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet Physics Doklady* 10 (1965), pp. 707–710.

[44] Heng Li; Xiangping Wu; Qingcai Chen; Qianjin Xiang. “Foreground and Text-lines Aware Document Image Rectification”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* IEEE, 2023, pp. 19517–19526. DOI: 10.1109/ICCV51070.2023.01793.

[45] Pu Li; Weize Quan; Jianwei Guo; Dong-Ming Yan. “Layout-aware Single-image Document Flattening”. In: *ACM Transactions on Graphics* 43.1 (2024), 9:1–9:17. DOI: 10.1145/3627818.

[46] Xiaoyu Li; Bo Zhang; Jing Liao; Pedro V. Sander. “Document rectification and illumination correction using a patch-based CNN”. In: *ACM Transactions on Graphics* 38.6 (2019), 168:1–168:11. DOI: 10.1145/3355089.3356563.

[47] Yanghao Li; Hanzi Mao; Ross B. Girshick; Kaiming He. “Exploring Plain Vision Transformer Backbones for Object Detection”. In: *Proceedings of the 17th European Conference on Computer Vision.* Ed. by Shai Avidan; Gabriel J. Brostow; Moustapha Cissé; Giovanni Maria Farinella; Tal Hassner. Vol. 13669. Lecture Notes in Computer Science. Springer, 2022, pp. 280–296. DOI: 10.1007/978-3-031-20077-9_17.

[48] Yun-Hsuan Lin; Wen-Chin Chen; Yung-Yu Chuang. “BEDSR-Net: A Deep Shadow Removal Network From a Single Document Image”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* Computer Vision Foundation / IEEE, 2020, pp. 12902–12911. DOI: 10.1109/CVPR42600.2020.01292.

[49] Philipp Lindenberger; Paul-Edouard Sarlin; Marc Pollefeys. “LightGlue: Local Feature Matching at Light Speed”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* IEEE, 2023, pp. 17581–17592. DOI: 10.1109/ICCV51070.2023.01616.

[50] Rafael Dueire Lins; Rodrigo Barros Bernardino; Darlisson Marinho de Jesus; José Mário Oliveira. “Binarizing Document Images Acquired with Portable Cameras”. In: *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition.* IEEE, 2017, pp. 45–50. DOI: 10.1109/ICDAR.2017.348.

[51] Rafael Dueire Lins; Ergina Kavallieratou; Elisa H. Barney Smith; Rodrigo Barros Bernardino; Darlisson Marinho de Jesus. “ICDAR 2019 Time-Quality Binarization Competition”. In: *Proceedings of the International Conference on Document Analysis and Recognition.* IEEE, 2019, pp. 1539–1546. DOI: 10.1109/ICDAR.2019.00248.

[52] Ce Liu; Jenny Yuen; Antonio Torralba. “SIFT Flow: Dense Correspondence across Scenes and Its Applications”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 978–994. DOI: `10.1109/TPAMI.2010.147`.

[53] Shaokai Liu; Hao Feng; Wengang Zhou. “Rethinking Supervision in Document Unwarping: A Self-Consistent Flow-Free Approach”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 34.6 (2024), pp. 4817–4828. DOI: `10.1109/TCSVT.2023.3336068`.

[54] Shaokai Liu; Hao Feng; Wengang Zhou; Houqiang Li; Cong Liu; Feng Wu. “DocMAE: Document Image Rectification via Self-supervised Representation Learning”. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 2023, pp. 1613–1618. DOI: `10.1109/ICME55011.2023.00278`.

[55] Shenlu Liu; Kangliang Liu. “ADIU:An Antiquarian Document Image Unwarping Dataset”. In: *Proceedings of the IEEE International Conference on Big Data*. Ed. by Shusaku Tsumoto; Yukio Ohsawa; Lei Chen; Dirk Van den Poel; Xiaohua Hu; Yoichi Motomura; Takuya Takagi; Lingfei Wu; Ying Xie; Akihiro Abe; Vijay Raghavan. IEEE, 2022, pp. 4181–4186. DOI: `10.1109/BIGDATA55660.2022.10020521`.

[56] Xiyan Liu; Gaofeng Meng; Bin Fan; Shiming Xiang; Chunhong Pan. “Geometric rectification of document images using adversarial gated unwarping network”. In: *Pattern Recognition* 108 (2020), p. 107576. DOI: `10.1016/J.PATCOG.2020.107576`.

[57] Ilya Loshchilov; Frank Hutter. “Fixing Weight Decay Regularization in Adam”. In: *arXiv* (2017). URL: `https://arxiv.org/abs/1711.05101`.

[58] Shijian Lu; Bolan Su; Chew Lim Tan. “Document image binarization using background estimation and stroke edges”. In: *International Journal on Document Analysis and Recognition* 13.4 (2010), pp. 303–314. DOI: `10.1007/S10032-010-0130-8`.

[59] Dong Luo; Pengbo Bo. “Geometric Rectification of Creased Document Images based on Isometric Mapping”. In: *arXiv* (2022). DOI: `10.48550/ARXIV.2212.08365`.

[60] Ke Ma; Sagnik Das; Zhixin Shu; Dimitris Samaras. “Learning From Documents in the Wild to Improve Document Unwarping”. In: *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference*. Ed. by Munkhtsetseg Nandigjav; Niloy J. Mitra; Aaron Hertzmann. ACM, 2022, 34:1–34:9. DOI: `10.1145/3528233.3530756`.

[61] Ke Ma; Zhixin Shu; Xue Bai; Jue Wang; Dimitris Samaras. “DocUNet: Document Image Unwarping via a Stacked U-Net”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 4700–4709. DOI: `10.1109/CVPR.2018.00494`.

[62] Amir Markovitz; Inbal Lavi; Or Perel; Shai Mazor; Roee Litman. “Can You Read Me Now? Content Aware Rectification Using Angle Supervision”. In: *Proceedings of the 16th European Conference on Computer Vision*. Ed. by Andrea Vedaldi; Horst Bischof; Thomas Brox; Jan-Michael Frahm. Vol. 12357. Lecture Notes in Computer Science. Springer, 2020, pp. 208–223. DOI: `10.1007/978-3-030-58610-2\_13`.

[63] Gaofeng Meng; Chunhong Pan; Shiming Xiang; Jiangyong Duan. “Metric Rectification of Curved Document Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.4 (2012), pp. 707–722. DOI: `10.1109/TPAMI.2011.151`.

[64] Gaofeng Meng; Yuanqi Su; Ying Wu; Shiming Xiang; Chunhong Pan. "Exploiting Vector Fields for Geometric Rectification of Distorted Document Images". In: *Proceedings of the 15th European Conference on Computer Vision*. Ed. by Vittorio Ferrari; Martial Hebert; Cristian Sminchisescu; Yair Weiss. Vol. 11220. Lecture Notes in Computer Science. Springer, 2018, pp. 180–195. DOI: `10.1007/978-3-030-01270-0\_11`.

[65] Mindee. *docTR: Document Text Recognition*. 2021. URL: `https://github.com/mindee/doctr` (visited on 12/19/2024).

[66] C. H. Nachappa; N. Shobha Rani; Peeta Basa Pati; M. Gokulnath. "Adaptive dewarping of severely warped camera-captured document images based on document map generation". In: *International Journal on Document Analysis and Recognition* 26.2 (2023), pp. 149–169. DOI: `10.1007/S10032-022-00425-4`.

[67] Hala Neji; Mohamed Ben Halima; Javier Nogueras-Iso; Tarek M. Hamdani; Javier Lacasta; Habib Chabchoub; Adel M. Alimi. "Doc-Attentive-GAN: attentive GAN for historical document denoising". In: *Multimedia Tools and Applications* 83.18 (2024), pp. 55509–55525. DOI: `10.1007/S11042-023-17476-2`.

[68] Zhaoyang Niu; Guoqiang Zhong; Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62. DOI: `10.1016/J.NEUCOM.2021.03.091`.

[69] Edwin Juma Omol. "Organizational digital transformation: from evolution to future trends". In: *Digital Transformation and Society* 3.3 (2024), pp. 240–256.

[70] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 9.1 (1979), pp. 62–66. DOI: `10.1109/TSMC.1979.4310076`.

[71] Xuebin Qin; Zichen Zhang; Chenyang Huang; Masood Dehghan; Osmar R. Zaïane; Martin Jägersand. "U^2-Net: Going deeper with nested U-structure for salient object detection". In: *Pattern Recognition* 106 (2020), p. 107404. DOI: `10.1016/J.PATCOG.2020.107404`.

[72] Alec Radford; Jong Wook Kim; Chris Hallacy; Aditya Ramesh; Gabriel Goh; Sandhini Agarwal; Girish Sastry; Amanda Askell; Pamela Mishkin; Jack Clark; Gretchen Krueger; Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila; Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763.

[73] Vijaya Kumar Bajjer Ramanna; Syed Saqib Bukhari; Andreas Dengel. "Document Image Dewarping using Deep Learning". In: *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods*. Ed. by Maria De Marsico; Gabriella Sanniti di Baja; Ana L. N. Fred. SciTePress, 2019, pp. 524–531. DOI: `10.5220/0007368405240531`.

[74] Alexander Sage; Eirikur Agustsson; Radu Timofte; Luc Van Gool. *LLD - Large Logo Dataset - version 0.1*. 2017. URL: `https://data.vision.ee.ethz.ch/cvl/lld` (visited on 04/11/2022).

[75] Gilles Simon; Salvatore Tabbone. "Generic Document Image Dewarping by Probabilistic Discretization of Vanishing Points". In: *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 2344–2351. DOI: `10.1109/ICPR48806.2021.9412649`.

[76] Karen Simonyan; Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the 3rd International Conference on Learning Representations*. Ed. by Yoshua Bengio; Yann LeCun. 2015.

[77] Leslie N. Smith; Nicholay Topin. “Super-convergence: Very fast training of neural networks using large learning rates”. In: *Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Vol. 11006. International Society for Optics and Photonics. 2019, p. 1100612.

[78] Ray Smith. “An Overview of the Tesseract OCR Engine”. In: *Proceedings of the 9th International Conference on Document Analysis and Recognition*. IEEE Computer Society, 2007, pp. 629–633. DOI: 10.1109/ICDAR.2007.4376991.

[79] Mohamed Ali Souibgui; Sanket Biswas; Sana Khamekhem Jemni; Yousri Kessentini; Alicia Fornés; Josep Lladós; Umapada Pal. “DocEnTr: An End-to-End Document Image Enhancement Transformer”. In: *Proceedings of the 26th International Conference on Pattern Recognition*. IEEE, 2022, pp. 1699–1705. DOI: 10.1109/ICPR56361.2022.9956101.

[80] Mohamed Ali Souibgui; Sanket Biswas; Andrés Mafla; Ali Furkan Biten; Alicia Fornés; Yousri Kessentini; Josep Lladós; Lluís Gómez; Dimosthenis Karatzas. “Text-DIAE: A Self-Supervised Degradation Invariant Autoencoder for Text Recognition and Document Enhancement”. In: *Proceedings of the 37th Conference on Artificial Intelligence*. Ed. by Brian Williams; Yiling Chen; Jennifer Neville. AAAI Press, 2023, pp. 2330–2338. DOI: 10.1609/AAAI.V37I2.25328.

[81] Mohamed Ali Souibgui; Yousri Kessentini. “DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2022), pp. 1180–1191. DOI: 10.1109/TPAMI.2020.3022406.

[82] Bolan Su; Shijian Lu; Chew Lim Tan. “Robust Document Image Binarization Technique for Degraded Document Images”. In: *IEEE Transactions on Image Processing* 22.4 (2013), pp. 1408–1417. DOI: 10.1109/TIP.2012.2231089.

[83] Matthew Tancik; Pratul P. Srinivasan; Ben Mildenhall; Sara Fridovich-Keil; Nithin Raghavan; Utkarsh Singhal; Ravi Ramamoorthi; Jonathan T. Barron; Ren Ng. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*. Ed. by Hugo Larochelle; Marc’Aurelio Ranzato; Raia Hadsell; Maria-Florina Balcan; Hsuan-Tien Lin. 2020.

[84] Ashish Vaswani; Noam Shazeer; Niki Parmar; Jakob Uszkoreit; Llion Jones; Aidan N. Gomez; Lukasz Kaiser; Illia Polosukhin. “Attention is All you Need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Ed. by Isabelle Guyon; Ulrike von Luxburg; Samy Bengio; Hanna M. Wallach; Rob Fergus; S. V. N. Vishwanathan; Roman Garnett. 2017, pp. 5998–6008.

[85] Floor Verhoeven; Tanguy Magne; Olga Sorkine-Hornung. “UVDoc: Neural Grid-based Document Unwarping”. In: *SIGGRAPH Asia 2023 Conference Papers*. Ed. by June Kim; Ming C. Lin; Bernd Bickel. ACM, 2023, 110:1–110:11. DOI: 10.1145/3610548.3618174.

[86] H. C. Vinod; S. K. Niranjan. “Camera Captured Document De-Warping and De-Skewing”. In: *Journal of Computational and Theoretical Nanoscience* 17.9-10 (2020), pp. 4398–4403.

[87] Qiang Wang; Bei Li; Tong Xiao; Jingbo Zhu; Changliang Li; Derek F. Wong; Lidia S. Chao. “Learning Deep Transformer Models for Machine Translation”. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics*. Ed. by Anna Korhonen; David R. Traum; Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 1810–1822. DOI: 10.18653/V1/P19-1176.

[88] Yonghui Wang; Wengang Zhou; Zhenbo Lu; Houqiang Li. "UDoc-GAN: Unpaired Document Illumination Correction with Background Light Prior". In: *Proceedings of the 30th ACM International Conference on Multimedia*. Ed. by João Magalhães; Alberto Del Bimbo; Shin'ichi Satoh; Nicu Sebe; Xavier Alameda-Pineda; Qin Jin; Vincent Oria; Laura Toni. ACM, 2022, pp. 5074–5082. DOI: 10.1145/3503161.3547916.

[89] Zelun Wang; Jyh-Charn Liu. "Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training". In: *International Journal on Document Analysis and Recognition* 24.1 (2021), pp. 63–75. DOI: 10.1007/S10032-020-00360-2.

[90] Zhou Wang; Eero P. Simoncelli; Alan C. Bovik. "Multiscale structural similarity for image quality assessment". In: *Proceedings of the 37th Asilomar Conference on Signals, Systems Computers*. Vol. 2. IEEE. 2003, pp. 1398–1402.

[91] Moritz Werling; Julius Ziegler; Sören Kammel; Sebastian Thrun. "Optimal trajectory generation for dynamic street scenarios in a Frenét Frame". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993. DOI: 10.1109/ROBOT.2010.5509799.

[92] Guo-Wang Xie; Fei Yin; Xu-Yao Zhang; Cheng-Lin Liu. "Dewarping Document Image by Displacement Flow Estimation with Fully Convolutional Network". In: *Proceedings of the 14th IAPR International Workshop, Document Analysis Systems 2020*. Ed. by Xiang Bai; Dimosthenis Karatzas; Daniel Lopresti. Vol. 12116. Lecture Notes in Computer Science. Springer, 2020, pp. 131–144. DOI: 10.1007/978-3-030-57058-3_10.

[93] Guo-Wang Xie; Fei Yin; Xu-Yao Zhang; Cheng-Lin Liu. "Document Dewarping with Control Points". In: *Proceedings of the 16th International Conference on Document Analysis and Recognition*. Ed. by Josep Lladós; Daniel Lopresti; Seiichi Uchida. Vol. 12821. Lecture Notes in Computer Science. Springer, 2021, pp. 466–480. DOI: 10.1007/978-3-030-86549-8_30.

[94] Qizhe Xie; Minh-Thang Luong; Eduard H. Hovy; Quoc V. Le. "Self-Training With Noisy Student Improves ImageNet Classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2020, pp. 10684–10695. DOI: 10.1109/CVPR42600.2020.01070.

[95] Zhen Xu; Fei Yin; Peipei Yang; Cheng-Lin Liu. "Document Image Rectification in Complex Scene Using Stacked Siamese Networks". In: *Proceedings of the 26th International Conference on Pattern Recognition*. IEEE, 2022, pp. 1550–1556. DOI: 10.1109/ICPR56361.2022.9956331.

[96] Chuhui Xue; Zichen Tian; Fangneng Zhan; Shijian Lu; Song Bai. "Fourier Document Restoration for Robust Document Dewarping and Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 4563–4572. DOI: 10.1109/CVPR52688.2022.00453.

[97] Shaodi You; Yasuyuki Matsushita; Sudipta N. Sinha; Yusuke Bou; Katsushi Ikeuchi. "Multiview Rectification of Folded Documents". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.2 (2018), pp. 505–511. DOI: 10.1109/TPAMI.2017.2675980.

[98] Fangchen Yu; Yina Xie; Lei Wu; Yafei Wen; Guozhi Wang; Shuai Ren; Xiaoxin Chen; Jianfeng Mao; Wenye Li. "DocReal: Robust Document Dewarping of Real-Life Images via Attention-Enhanced Control Point Prediction". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE, 2024, pp. 654–663. DOI: 10.1109/WACV57701.2024.00072.

[99] Syed Waqas Zamir; Aditya Arora; Salman Khan; Munawar Hayat; Fahad Shahbaz Khan; Ming-Hsuan Yang. “Restormer: Efficient Transformer for High-Resolution Image Restoration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 5718–5729. DOI: 10.1109/CVPR52688.2022.00564.

[100] Jiaxin Zhang; Bangdong Chen; Hiuyi Cheng; Lianwen Jin; Kai Ding; Fengjun Guo. “DocAligner: Annotating Real-world Photographic Document Images by Simply Taking Pictures”. In: *arXiv* (2023). DOI: 10.48550/ARXIV.2306.05749.

[101] Jiaxin Zhang; Lingyu Liang; Kai Ding; Fengjun Guo; Lianwen Jin. “Appearance Enhancement for Camera-Captured Document Images in the Wild”. In: *IEEE Transactions on Artificial Intelligence* 5.5 (2024), pp. 2319–2330. DOI: 10.1109/TAI.2023.3321257.

[102] Jiaxin Zhang; Canjie Luo; Lianwen Jin; Fengjun Guo; Kai Ding. “Marior: Margin Removal and Iterative Content Rectification for Document Dewarping in the Wild”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. Ed. by João Magalhães; Alberto Del Bimbo; Shin’ichi Satoh; Nicu Sebe; Xavier Alameda-Pineda; Qin Jin; Vincent Oria; Laura Toni. ACM, 2022, pp. 2805–2815. DOI: 10.1145/3503161.3548214.

[103] Jiaxin Zhang; Dezhi Peng; Chongyu Liu; Peirong Zhang; Lianwen Jin. “DocRes: A Generalist Model Toward Unifying Document Image Restoration Tasks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2024, pp. 15654–15664. DOI: 10.1109/CVPR52733.2024.01482.

[104] Richard Zhang; Phillip Isola; Alexei A. Efros; Eli Shechtman; Oliver Wang. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 586–595. DOI: 10.1109/CVPR.2018.00068.

[105] Tongjie Y. Zhang; Ching Y. Suen. “A Fast Parallel Algorithm for Thinning Digital Patterns”. In: *Communications of the ACM* 27.3 (1984), pp. 236–239. DOI: 10.1145/357994.358023.

[106] Weiguang Zhang; Qiufeng Wang; Kaizhu Huang. “Polar-Doc: One-Stage Document Dewarping with Multi-Scope Constraints under Polar Representation”. In: *arXiv* (2023). DOI: 10.48550/ARXIV.2312.07925.

[107] Weiguang Zhang; Qiufeng Wang; Kaizhu Huang; Xiaomeng Gu; Fengjun Guo. “Coarse-to-Fine Document Image Registration for Dewarping”. In: *Proceedings of the 18th International Conference on Document Analysis and Recognition*. Ed. by Elisa H. Barney Smith; Marcus Liwicki; Liangrui Peng. Vol. 14807. Lecture Notes in Computer Science. Springer, 2024, pp. 343–358. DOI: 10.1007/978-3-031-70546-5_20.

[108] Weiguang Zhang; Qiufeng Wang; Kaizhu Huang; Xiaowei Huang; Fengjun Guo; Xiaomeng Gu. “Document Registration: Towards Automated Labeling of Pixel-Level Alignment Between Warped-Flat Documents”. In: *Proceedings of the 32nd ACM International Conference on Multimedia*. Ed. by Jianfei Cai; Mohan S. Kankanhalli; Balakrishnan Prabhakaran; Susanne Boll; Ramanathan Subramanian; Liang Zheng; Vivek K. Singh; Pablo César; Lexing Xie; Dong Xu. ACM, 2024, pp. 9933–9942. DOI: 10.1145/3664647.3681548.