InpaintAR

A Comparison of different InPainting Approaches for Real-Time Use

Sebastian Gradwohl

BACHELOR THESIS

submitted to the
Bachelor's Degree Program
Software Engineering

at the
University of Applied Sciences Upper Austria
in Hagenberg

2026

Supervisor: FH-Prof. Dr. Christoph Anthes, MSc

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed copy is identical to the submitted electronic version.

Hagenberg, February 1, 2026

Sebastian Gradwohl

Contents

Preface

This Chapter will be removed eventually and only serves as a disclaimer and information page for the reader on the approach taken for this project

As of now, these few pages are supposed to provide a broad overview of the writing style and approach taken for writing this thesis.

The goal is to write the thesis in parallel to the implementation of the tool, so whenever a new algorithm is analyzed it is first written about in the thesis here, then implemented and evaluated, followed by the documentation in this thesis.

At the end, all the results will be aggregated and compared, to provide some final remarks and provide ideas on how to expand on the current tool

Current Timetable/Milestones:
Mid-October:
- Finish Base-Framework for the tool, to select planes like tables and detect objects above it, for this both the meta passthrough api aswell as the depth-api are going to be used (existing tools for selecting planes and getting a depth-map of the camera feed to detect objects on the selected surfaces)
- Best-Case -> Already black out the found objects, so its basically a black box covering the object, which is updated in real time, should be easy to achieve with the camera coordinates of the objects and a shader (Passthrough API allows Application of Shaders to the Camera Feed, so the concept is to implement the Inpainting as a shader)

Mid-November:
- Finish at least 3 Methodology Evaluations, maybe optimize the tool a bit
- Trial of combining the tool with digital objects, Demo: Interactive Chess Board on a Table with a few notebooks and pens, maybe a glass of water to test reflection behavior

Mid-December:
- Have the Chapters Fundamentals and Concept completely finished
- Best-Case very far with Evaluation

End-December / Early January (Sometime during the Chrismas Holidays):
- Have a first draft to send in for a first review, then there should still be a few weeks time to work in the feedback

Abstract

Due to technological advances in both Hardware and Software-Integration, Augmented Reality is getting more popular with each year.
One of the issues still prevalent in Augmented Reality Software is the distraction caused by clutter surrounding the digital content in use cases like immersive analytics.

This bachelors thesis focuses on the analysis of the different methods and algorithms for the concealment of the clutter to provide a clean and non-distracting environment for the digital object to be displayed on.
To accomplish this, a tool to select planes on which objects are detected to subsequently be hidden through inpainting will be implemented. With this tool, several methods and algorithms will be evaluated in terms of both quality and performance to provide believable and uncluttered environments, whilst at the same time minimizing the use of computational resources and time.
These findings are compared to each other so readers can get an overview on both the advantages and disadvantages each of the analyzed approaches delivers.

Kurzfassung

Aufgrund der technologischen Fortschritte, sowohl in der Hardware, als auch in der Software-Integration, steigt die Beliebtheit von Augmented Reality mit jedem Jahr an. Ein sehr gängiges Problem in aktueller Augmented Reality Software ist die Ablenkung, welche durch physische Objekte, die um den digitalen Inhalten liegen. Beispielsweise bei der immersiven Analyse von Daten.

In dieser Bachelorarbeit liegt der Fokus auf der Analyse verschiedener Methoden und Algorithmen zur Verdeckung dieser Objekte, wodurch eine aufgeräumte, saubere Umgebung für die digitalen Elemente geschaffen wird.
Um dies zu erreichen wird ein Werkzeug entwickelt, welches erlaubt Flächen zu wählen, welche dann dazu verwendet werden, um Objekte auf diesen Flächen zu finden und darauffolgend durch die Nutzung von Inpainting zu verdecken. Mithilfe dieses Werkzeugs werden mehrere Methoden und Algorithmen sowohl auf Qualität, als auch auf Leistung evaluiert, um sowohl die Glaubbarkeit und Ordnung der generierten Umgebungen, während parallel dazu ein Fokus auf die Minimierung von Rechenresourcen und Laufzeit gelegt wird.
Diese Ergebnisse werden zueinander verglichen, sodass Leser sowohl über die Vor- als auch Nachteile der jeweiligen Ansätze einen Überblick bekommen.

Chapter 1

Introduction

The Domain of Augmented Reality (AR) spans all kinds of real-time visualization approaches wherein an otherwise real environment gets augmented with virtual objects [Mil+95] While many still consider Augmented Reality a nieche, the user-numbers tell a different story. In 2023 Statista reported about 983 Million users to be using AR on their mobile devices. This number is set to grow to 1.187 Billion users until 2028 [Int24].

One of the issues still common in AR Applications is the overload of information. Whenever users are supposed to focus on the virtual objects at hand, the visibility of physical objects may serve as a distraction. (cf. Cheng et. al. [Che+22]) An existing solution is the use of Diminished Reality. This Concept aims to hide certain physical objects from the viewport, which can in this case be used to reduce the visual clutter.

1.1 Motivation

While there are many existing solutions for achieving Diminished Reality using Inpainting , the topic of which inpainting method for not only this usecase, but Inpainting in a 3-Dimensional Real-Time Environment has largely been disregarded as of now. Most Papers put the focus on different components of the system like the detection of the objects, user studies or optimizations of smaller visual artifacts. Over the last years there have been numerous achievements in the space of Inpainting methods, ranging from classic, low-cost, algorithms like the fast marching algorithm [SHD21] or the usage of matching textures using gaussian weighting to fill the affected regions [DRR19] to high fidelity machine-learning assisted approaches like the DeepFill Deep-Learning model.[CJL23]

The main goal of this analysis is to provide an easy to reference comparison of the different methodologies to be compared in terms of both performance and quality. Through this analysis, developers can make an educated choice on which methodology to use for their usecase, be it high fidelity in usecases like PC-Powered AR Systems or low energy, low performance systems, more suited for mobile devices, be they mobile phones or standalone Mixed-Reality Head-Mounted Displays. In addition, the tool built for this evaluation should be easy to extend for newer inpainting methods whilst staying easy to integrate into existing applications.

1.2 Overview

This work includes an introduction to the underlying concepts and analyzed inpainting methods, followed by the conceptualization and implementation of the tool as well as the performance and quality evaluation for each of the analyzed inpainting methods. These topics are elaborated upon in the following five chapters:

- **Chapter 2: Fundamentals and Related Work**
  Provides an introduction into visual clutter and why it matters to the domain of Augmented Reality, an introduction to the term of Diminished reality aswell as the underlying logic of each of the inpainting methods.
- **Chapter 3: Concept**
  Elaborates on the architecture and design chosen for this tool and provides an overview on a few example usecases for which the tool might be used
- **Chapter 4: Implementation**
  The focus lies on providing an explanation on each of the classes contained in this tool and how it can be implemented into existing projects to be used. In Addition some of the tools and principles used during the implementation are going to be elaborated on.
- **Chapter 5: Evaluation**
  After providing an overview of the testing setup and how visual clutter and performance are measured, the test scenarios will be presented, followed by the results for each of the analyzed methods.
- **Chapter 6: Conclusion**
  Presents the findings of the evaluation in a summed up manner and goes into detail on how the tool might be expanded in the future to improve the tool to provide an even better user and developer experience

References

Literature

[Che+22] Yi Fei Cheng et al. “Towards Understanding Diminished Reality”. eng. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2022, pp. 1–16 (cit. on p. 1).

[CJL23] Muzi Cui, Hao Jiang, and Chaozhuo Li. “Progressive-Augmented-Based DeepFill for High-Resolution Image Inpainting”. eng. *Information (Basel)* 14.9 (2023), p. 512 (cit. on p. 1).

[DRR19] Ding Ding, Sundaresh Ram, and Jeffrey J. Rodriguez. “Image Inpainting Using Nonlocal Texture Matching and Nonlinear Filtering”. eng. *IEEE transactions on image processing* 28.4 (2019), pp. 1705–1719 (cit. on p. 1).

[Mil+95] Paul Milgram et al. “Augmented reality: a class of displays on the reality-virtuality continuum”. In: *Telemanipulator and Telepresence Technologies*. Ed. by Hari Das. Vol. 2351. International Society for Optics and Photonics. SPIE, 1995, pp. 282–292. DOI: 10.1117/12.197321 (cit. on p. 1).

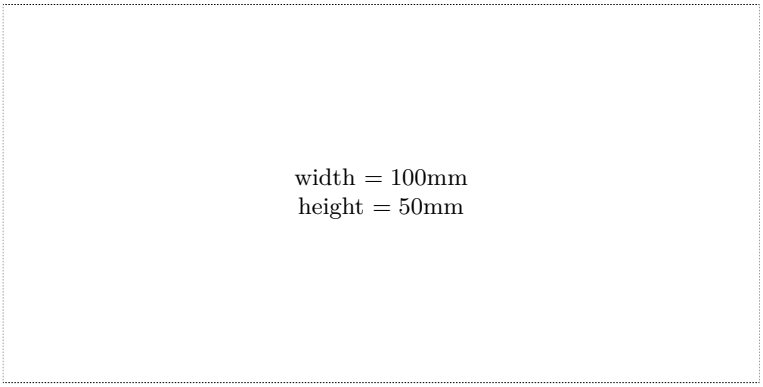
[SHD21] Irawati Nurmala Sari, Emiko Horikawa, and Weiwei Du. “Interactive Image Inpainting of Large-Scale Missing Region”. eng. *IEEE Access* 9 (2021), pp. 56430–56442 (cit. on p. 1).

Online sources

[Int24] ARtillery Intelligence. *Mobile augmented Reality (AR) users worldwide from 2023 to 2028*. 2024. URL: https://www.statista.com/statistics/1098630/global-mobile-augmented-reality-ar-users/ (visited on 08/31/2025) (cit. on p. 1).

Check Final Print Size

— Check final print size! —



— Remove this page after printing! —

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.