TSBB11 - Images and Graphics, Project Course CDIO - Re-Imagining Historic Art

Viktor Bergkvist, Alfred Johannesson, Louise Poteau, and Robin Wibom

Abstract—This project investigates the possibility of using modern tools and methods to both create digital copies of lost art pieces as well as finding new ways of interacting with art. The main investigated areas were creating 3D models from limited amounts of photos and using machine learning to color old grayscale photos. The primary methods used for the 3D modeling were *structure from motion*, *monocular depth estimation*, and *dense feature matching*. The Recoloring was done by finetuning a base model from *DeOldify*. Everything was combined into a virtual museum. More improvements could be made, however the museum shows the viability of creating modern digital copies of art.

1 Introduction

This project is a collaboration between Linköping Municipality, the Computer Vision Laboratory (CVL) at Linköping University, and AI Sweden. It aims to revive the historical mural painted on the wall of "Röda Magasinet," which was covered a hundred years ago, using cutting-edge technology. The project employs archival photos, recoloring techniques, and 3D reconstruction to recover the mural and the history behind this museum. The ultimate goal is to make it accessible and engaging for as many people as possible.

In this project, we explored several approaches to address the challenges of reconstructing historical artwork from limited material. *Structure from Motion (SfM)* was chosen for its ability to perform full 3D reconstruction, allowing flexibility in selecting viewpoints and creating detailed visualizations. However, SfM depends on having many overlapping images, which is often not available for historical objects. To handle this, we also looked into monocular depth estimation, which is used to predict 3D geometry from a single image. In addition, we tested dense feature matching, as it is good at finding correspondences between images even when input data is sparse. These methods complement each other, providing a balance between how many images are needed, the level of detail in the reconstruction, and robustness to different kinds of input.

1.1 Structure from Motion

Structure from Motion (SfM) is a technique used for reconstructing 3D geometry from a series of 2D images. The process identifies and matches key features across multiple overlapping images, estimates the camera positions and orientations, and triangulates corresponding points in 3D space to produce a sparse point cloud (Schönberger and Frahm, 2016).

COLMAP is an open-source SfM tool (Schönberger, 2024) that streamlines this process. The reconstruction steps consist of:

1) **Feature Detection:** Keypoints are detected in the input images using algorithms such as *SIFT*, which extracts distinctive and repeatable features.
2) **Feature Matching:** Correspondences between features in overlapping images are established through exhaustive pairwise matching, with geometric verification ensuring robust matches.
3) **Sparse Reconstruction:** Using the verified matches, camera positions and orientations are estimated. Corresponding points are triangulated to produce a sparse 3D point cloud.
4) **Dense Reconstruction:** Once a sparse model is established, COLMAP generates a dense point cloud by computing depth maps for each image. This involves undistorting images, performing stereo matching, and fusing depth maps.
5) **Surface Meshing:** The dense point cloud is further processed into a mesh. COLMAP supports two methods:
   - **Poisson Meshing:** Produces a smooth, watertight surface.
   - **Delaunay Meshing:** Creates a triangulated mesh optimized for visualization.

This streamlined pipeline allows for automatic 3D reconstruction and is a key part of this project.

1.2 Monocular Depth Estimation

Estimating the 3D geometry of general scenes is a core challenge in computer vision. While significant advancements have been made in reconstructing 3D structures from multiple images using approaches like SfM, (Schönberger and Frahm, 2016 Agarwal et al., 2011), inferring 3D geometry from a single image across arbitrary domains remains a highly ill-posed and complex problem.

As an alternative to SfM we explored the implementation of monocular depth estimation. We believe this approach offers a solution for making history more interactive, which aligns well with the goals of this project. The reason for this is that in many cases historical objects that have been "captured" by cameras might only have very few good images of them, making other reconstruction methods ineffective.

Monocular geometry estimation (MGE) typically involves first predicting a depth map, which is scaled and shifted by an unknown factor. This depth map is then combined with camera intrinsics to reconstruct the 3D shape through unprojection, (Yin et al., 2020). Recent advancements in *monocular depth estimation (MDE)* have been driven by training on large-scale datasets, (Yang et al., 2024). However, accurately determining camera intrinsics, such as focal length, from a single image remains difficult due to significant ambiguity in the absence of strong geometric cues.

In our project, we utilized *MoGe*, *(Monocular Geometry Estimation)*, a method for monocular depth and geometry estimation. MoGe is designed to recover 3D geometry from monocular open-domain images. The model consists of a *Vision Transformer (ViT)* encoder and a convolutional decoder, which together predict an affine-invariant point map and a mask that excludes regions with undefined geometry, such as the sky. (Wang et al., 2024)

The method directly predicts point maps from images, which can be used to derive a depth map and camera parameters like focal length or *field of view (FOV)* when needed. The point-maps are affine-invariant, where the 3D points are influenced by an unknown global scale and a 3D shift. This distinction is crucial, as it resolves the focal-distance ambiguity, which can negatively impact network training. (Wang et al., 2024)

MoGe employs a set of novel global and local geometry supervisions to enhance the accuracy of its predictions. These include a robust point cloud alignment solver for accurate global shape learning and a multi-scale local geometry loss for precise local geometry supervision. The model is trained on a large, mixed dataset, demonstrating strong generalizability and high accuracy across diverse unseen datasets. (Wang et al., 2024)

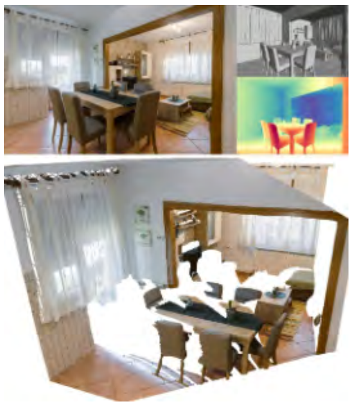

Figure 1. Given an image, MoGe reconstructs an affine-invariant 3D point map of the scene and can also produce a depth map. Figure from Wang et al., 2024

1.3 Robust Dense Feature Matching

Feature matching is a technique in computer vision used to find corresponding points or features between two or more images. (Example in Figure 2.) Dense methods estimate all such correspondences. It plays a role in tasks like image alignment and 3D reconstruction.

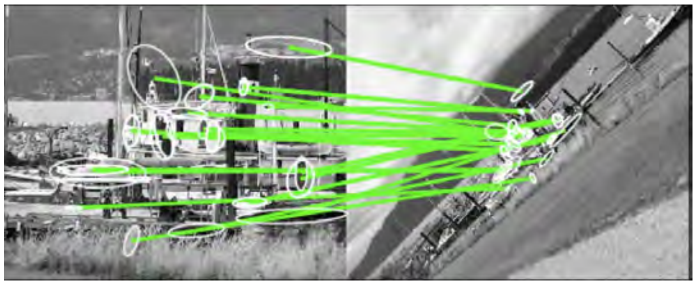

Figure 2. Matching between two images. Figure from lecture by Per-Erik Forsén.

Having very few old images for this project, a model for robust dense feature matching, RoMa (Edstedt et al., 2023), was tested and used.

Previous methods often learn coarse features through 3D supervision, but this approach has drawbacks. Collecting real world 3D datasets is expensive, leading to limited data and a higher risk of overfitting, which reduces robustness to unseen scenes. Freezing pretrained backbones can mitigate overfitting, but their out-of-the-box performance for feature matching is insufficient. (Edstedt et al., 2023). Recently,

Self-supervised pretraining with *Masked Image Modeling (MIM)* has shown promise Xie et al., 2022. MIM works by masking random patches of an image and training a model to reconstruct the missing parts. This forces the model to focus on understanding spatial relationships and local details in the image, which helps it retain local information better than classification-based pretraining. These properties allow MIM models to generalize effectively to dense vision tasks.

DINOv2 is a MIM model Oquab et al., 2024. However, applying DINOv2 to feature matching remains challenging due to the lack of fine features needed for precise refinement.

RoMa solves this by by using a frozen DINOv2 encoder for coarse features and a specialized ConvNet encoder for the fine features. To effectively leverage the coarse and fine features extracted by the DINOv2 and specialized ConvNet encoders, RoMa incorporates a Transformer-based Match Decoder.

1.4 Photoshop

Photoshop is software that enables users to modify images. It allows for manual image colorization using a variety of available brushes.

Moreover, Photoshop includes a feature that uses artificial intelligence to automatically colorize images. This feature, known as Neural Filters, employs machine learning to enhance, modify, or transform images. Among the various Neural Filters, the one designed for image colorization is called *Colorize* 2023.

This tool generates colors automatically but also lets users adjust and enforce specific colors on selected parts of the image. If a color is not specified, the tool will fill in the rest with the color that seems to correspond best. Multiple colors can be enforced at different points of the image, or none at all. The tool will automatically adjust based on the information provided.

1.5 DeOldify

DeOldify is an open source project started in 2018 by Jason Antic. The goal of the DeOldify project is to restore old images and videos using machine learning (Antic, 2024). It is available under an MIT license and can be freely used and modified. DeOldify not only has the source code, but also several different trained models available. The models are primarily trained using *Imagenet*, which is explained in Section 1.7. The three available models are; "stable", "artistic", and "video". The artistic model is trained to have more vibrant colors, however it is typically worse than stable for landscapes and portraits Antic, 2024.

1.6 GAN and NoGAN Machine Learning

GAN stands for *Generative Adversarial Network*. It is a form of machine learning system based on having two neural networks working against each other to enhance performance. The two networks in a GAN are a generator and a critic. The generator is the network that will be used once training is complete. In this project, the generator is trained to take an input of a black and white image and output a colored version. The critic is then trained to determine whether a colored image is an original image or generated by the generator.

NoGAN is a much more recent, and not as well established, form of machine learning compared to standard GAN. It is the basis of the DeOldify project. A NoGAN system is a combination of standard convolutional neural networks and GAN. The generator and critic are first trained as standard convolutional neural networks before they are combined into a GAN system (Antic, 2024).

1.7 ImageNet

Imagenet is a Database of labeled images for training machine learning models, Deng et al., 2009. At the time of writing this report, the database contains over 14 million images. Imagenet was considered, but not directly used in this project. However, DeOldify used Imagenet for training the models that were used as a base in this project.

1.8 Unreal Engine

Unreal Engine is a powerful real-time 3D visualization application used to create immersive environments and realistic simulations. It supports physically-based rendering (PBR), dynamic lighting, and material customization, making it ideal for high-quality visual presentations (Epic Games, 2024b).

For this project, we utilized Unreal Engine's tool set to realistically display the reconstructed building, along with the mural, old paintings, and uniforms. Additionally, we made use of the FAB library, which offers a wide range of free high-quality 3D assets that integrate seamlessly with Unreal Engine (Epic Games, 2024a). Together, these features allowed us to create an engaging and historically inspired virtual environment.

2 Problem Formulation

This project aims to re-imagine historic art and buildings, using modern tools and limited data. The primary questions explored are:

1) How can we recreate and present historic art in a way that is both interactive and engaging?
2) How can modern computer vision and machine learning techniques be used to accurately restore and colorize old buildings and images?
3) Is it possible to create a historically accurate visualization of the "Röda Magasinet" building, together with its mural?

To address these questions, the project defines the following concrete goals:

- **Create an accurate 3D reconstruction of Röda Magasinet** using Structure from Motion to recreate the building's geometry.
- **Colorize grayscale images** using machine learning (DeOldify) and manual editing to achieve historically accurate representations of the uniforms and other elements.
- **Develop an interactive virtual environment** in Unreal Engine to visualize the reconstructed models, images, and the mural in an immersive way.
- **Bring old uniforms and paintings to life** by using monocular depth estimation to create 3D-like models to be viewed in the virtual environment.
- **Explore dense feature matching** to interpolate views between historical images, simulating a 3D-like experience with limited data.

3 System Design

The system design for this project involved four main components: the reconstruction of a historic building, image recoloring, monocular depth estimation, and trying a robust dense feature matching method. Each component contributed to creating a virtual representation of the building and other relevant photos.

3.1 Building Reconstruction

The reconstruction of the historic building was carried out in several steps, starting with data acquisition and proceeding through 3D modeling and texture integration:

3.1.1 Data Acquisition

Images of the building were captured in the real world. The images were taken in a large circular pattern around the building, ensuring sufficient overlap and diverse viewpoints, as recommended by the COLMAP documentation.

3.1.2 3D Reconstruction

The captured images were processed in COLMAP to generate a sparse point cloud and a Poisson mesh. These outputs served as the foundation for creating the building's 3D geometry.

We chose COLMAP for this project due to its streamlined approach for SfM. As described in Section 1.1, COLMAP simplifies the 3D reconstruction process and can generate high-quality point clouds, which is exactly what is needed for this project.

3.1.3 Model Cleanup and Simplification

The Poisson mesh was imported into Meshlab, where it was cleaned to remove unnecessary geometry. The cleaned model was then imported into Blender, where its complexity was reduced even further. This simplification ensured efficient rendering while preserving key details.

3.1.4 Texture Mapping and Mural Integration

Before simplifying the model in Blender, the original texture was extracted to preserve resolution and detail. A texture map was created and used as a base for adding the mural, seamlessly integrating it with the building's geometry. The final model, now including the mural, was exported in FBX format for use in Unreal.

Blender was a natural choice for this process due to its ease of use and built-in tools, such as the decimation function that allowed us to efficiently reduce the model's complexity. Additionally, the *Smart UV Project* feature provided a straightforward way to create a UV map, which was essential for accurate texture mapping. Blender was also one of the few programs capable of directly exporting to FBX, a format fully supported by Unreal, making it an ideal choice for this project.

3.1.5 Integration and Visualization

The FBX file was imported into Unreal Engine 5, where the reconstructed building and its mural were displayed in a virtual environment, completing the building reconstruction process.

We chose Unreal Engine over other alternatives like OpenGL due to its advanced rendering capabilities and ease of use for creating realistic visualizations. Unlike OpenGL, Unreal comes with many features already implemented, which were mentioned previously in 1.8. This allowed us to focus more on the artistic and creative aspects of the project, rather than spending time on less relevant technical details on the rendering side.

3.2 Coloring

Recoloring was attempted with two different methods. One was using Photoshop, and one was a machine learning model based on DeOldify. A dataset of images was supplied by the municipality for this task. The dataset contained 25 grayscale images and 24 colored images.

3.2.1 Photoshop

To ensure accurate color matching, we referred to the available reference images, focusing on details such as costumes to determine their corresponding colors. However, not all costumes were documented in the references, so some of the soldiers couldn't be colorized.

We also identified two key reference points: the building, which was known to be red, and the grass, which was known to be green, both of which served as the basis for our adjustments.

3.2.2 Machine learning

The machine learning model was based on the artistic model available from DeOldify. Starting from a trained model was done due to limited time and data. DeOldify was chosen because of its MIT license, availability, and proven quality. The license ensures that there are no legal issues with modifying the code. The code is freely available to download from Github. It is also relatively simple to set up on a linux-based machine with access to python. DeOldify has been used as a base for other successful projects and commercial ideas and is therefore known to have a certain level of quality.

The model was finetuned to increase the historical accuracy of the recoloring. The available colored images were included as training data for additional rounds of GAN training for the model. Grayscale copies were created for each image, those images were then recolored by the generator. The critic inspected a batch of images every round and classified images as either original or modified by the generator. Once finetuning was complete the finished generator model was used to recolor all grayscale images.

3.3 Monocular Depth Estimation

As presented in Section 1.2 MoGe was used to create 3D models based on single images. With some changes to their published code, (Wang et al., 2024) we were able to generate 3D-models from different images. (One example in Figure 3 of a mannequin.) Once the 3D model was created we processed the models in Meshlab and Blender. Since the generated models from MoGe often had a very high complexity Meshlab could be used to do a rough trimming of the models to make them smaller and making it possible to import them to Blender. In Blender some final cleanup was done as well as downsampling the models. Downsampling was done for the virtual museum to run smoother and did not affect the visual quality. These models were later imported to Unreal Engine.

MoGe was chosen for this project because it is a recently published method, delivers promising results, and makes generating a good 3D model easier compared to the other methods we considered.

3.4 Robust Dense Feature Matching

After testing with different ideas of how to use and display RoMa in this project a GIF was made by using the matching method on 3 different images each representing a side of Röda magasinet to estimate warps between them. The aim was to create the impression of moving around the building, making the experience more immersive than viewing a still image. This was done with minor changes to the demo code provided at github.com/Parskatt/RoMa (Johan Edstedt, 2023). The animation was done by parameterizing $W(t) = (1-t)x^A + t\hat{x}^B$ where x^A is the pixel coordinates corresponding to image A and $\hat{x}^B$ are the pixel coordinates warped to image B using the warp field calculated by the RoMa model.

4 Results

This Section presents the outcome of the project, focusing on the reconstruction of the "Röda Magasinet" building,

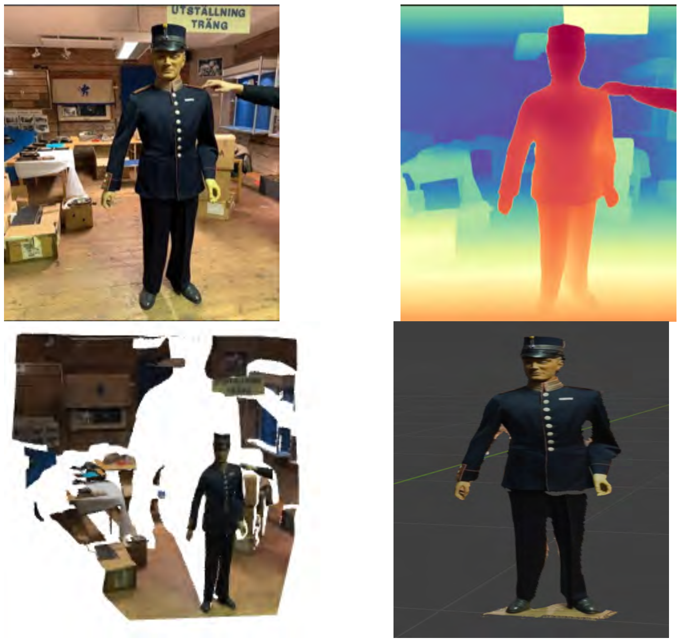

Figure 3. From a 2D picture (top left) to a trimmed 3D model (bottom right). The result of MoGe in bottom left and resulting depth map top right.

recoloring old grayscale images, and their integration into the virtual museum. It details the methods used for 3D modeling, image processing, and visualization, as well as the challenges faced and solutions implemented during the process.

4.1 Building reconstruction

The reconstruction of "Röda Magasinet" involved several stages, from collecting data to producing a detailed 3D model. The process included capturing images of the building, processing them to generate a point cloud, cleaning and simplifying the model for visualization, and finally creating a texture map to preserve visual details.

4.1.1 Data Acquisition

A total of 200 images were taken, evenly spread out around the building to try and maximize the number of correspondences in each image. Figure 4 provides examples of the captured images. These images were used for the reconstruction of the building.

However, as illustrated in the third example in Figure 4, the backside of the building was partially obstructed by trees. This obstruction resulted in an incomplete reconstruction of that side of the building.

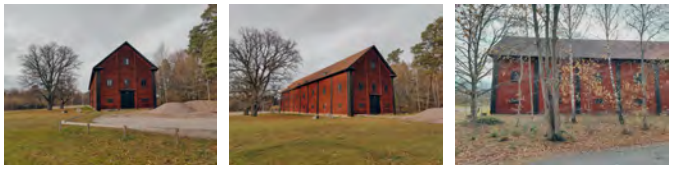

Figure 4. Examples of three out of the 200 images taken of the building. The third image shows the backside of the building obstructed by the trees.

4.1.2 3D Reconstruction

The images captured were processed in COLMAP to perform the 3D reconstruction of the building. The reconstruction followed the automatic pipeline described in Section 1.1, which includes steps such as feature detection, feature extraction, and dense reconstruction.

Figure 5 showcases the resulting sparse 3D point cloud of the building, including the estimated camera positions as visualized in COLMAP. For the meshing process, the Poisson surface reconstruction method was used and later imported into MeshLab for cleanup. More details can be found in the next Section 4.1.3).

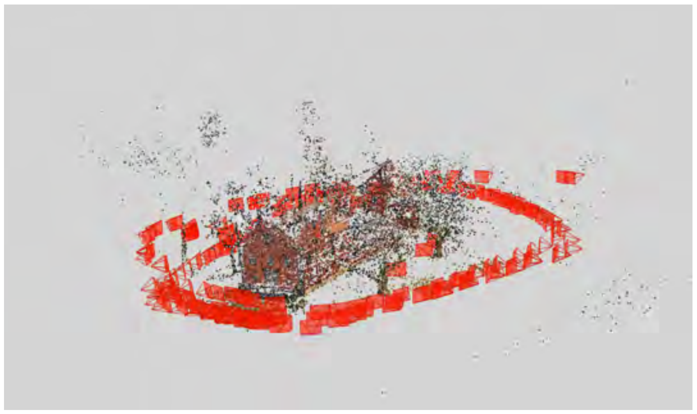

Figure 5. Showcase of the pointcloud of the building and camera coordinates, inside COLMAP

4.1.3 Model Cleanup and Simplification

The Poisson mesh and pointcloud were imported into MeshLab for cleanup. Objects such as trees, grass, and other elements not part of the building were identified and removed to focus solely on the building itself.

This process required careful inspection to ensure that no important details of the building were accidentally deleted, as MeshLab does not support an undo function. Figure 6 shows the raw, untrimmed version compared to the cleaned and simplified version.

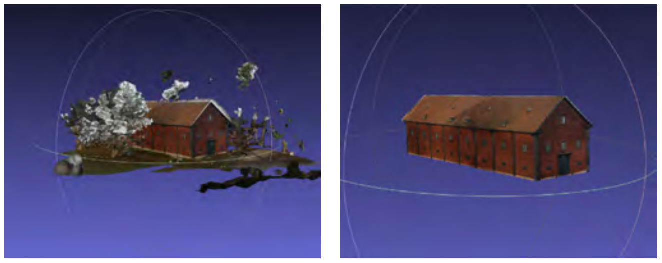

Figure 6. Progression of the cleanup process, inside MeshLab.

After cleanup, the trimmed model contained approximately 1.2 million faces and $600,000$ vertices, with each vertex holding data for position, normal, and color. The color data portion is particularly important and will be discussed later. However, the level of complexity is too high for efficient rendering in Unreal Engine, meaning further simplification is needed.

Using Blender, a decimation technique was applied to reduce the number of faces. Initially, the collapse method was used, which merges vertices progressively while maintaining the mesh's overall shape. A decimation ratio of 0.1 was applied, reducing the face count by 90 percent. Next, a planar modifier was added to simplify flat surfaces by merging faces with similar angles, using a threshold of 5 degrees. After these modifications, the model was reduced to around $6,000$ faces, achieving an overall reduction of 99.5 percent.

Figure 7 illustrates the simplified model, shown with and without color. The texture appears blurry and low quality in the vertex-colored version. This is because reducing the number of vertices also gets rid of necessary color information needed for coloring. To avoid this loss of detail, a UV map and texture extraction are necessary before reducing the model's complexity. This process ensures the preservation of finer details and is discussed in the next Section 4.1.4.

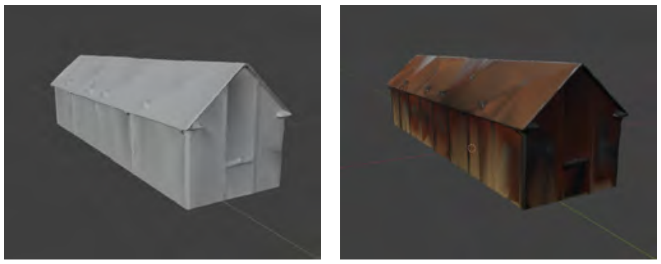
Figure 7. The simplified model in Blender: (Left) With material only; (Right) With vertex color.

4.1.4 Texture Mapping

To preserve the details of the building's appearance after reducing the model's complexity, a texture map was created in Blender. The process involves unwrapping the model's surface and baking its color information into a texture file.

First, the *Smart UV Project* method was used to automatically generate a UV map. Next, the texture was baked onto the UV map by creating an image in the *UV Editor* and using Blender's *bake* function. The baked texture stores color and surface details, allowing the simplified model to still have good quality. Figure 8 shows the resulting texture map. The texture map was then also edited to include the colored mural.

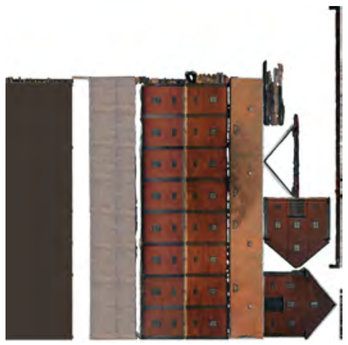
Figure 8. The generated texture map of the building.

4.2 Robust Dense Feature Matching

Example of the RoMa method on photos relevant to this project can be seen in Figure 9. The generated GIF is best visualized by running the digital museum on a computer.

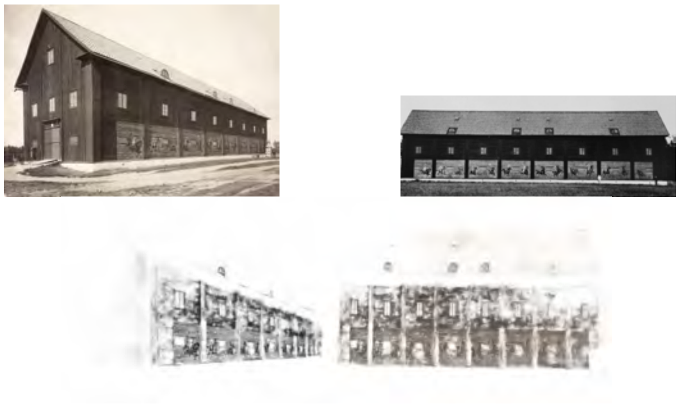
Figure 9. Correspondences estimated by RoMa on two old photographs of Röda magasinet. The estimated correspondences are visualized by grid sampling coordinates bilinearly from the other image, using the estimated warp, and multiplying with the estimated confidence.

4.3 Monocular Depth Estimation

Depth map and 3D model from MoGe on both an indoor- and outdoor scene can be seen in Figure 10.

4.4 Recoloring

This Section displays a sample of the results from recoloring the available images. The same image is displayed for multiple steps for easier comparison.

4.4.1 Photoshop

Photoshop was used in two different ways. First, we used the Colorize tool to do a global colorization of the image and see what the initial results were. We specified that the building was red and the grass was green, and the rest of the colors were calculated by Photoshop's machine learning tool. The process took less than a minute to calculate. The results are shown in Figure 11. The colors weren't really discernible, so we decided to do some manual coloring in addition to the global coloring.

For the manual coloring we used the previously obtained colored image as a base and hand-colored each costume using the provided references. Some of the costumes were left uncolored due to a lack of information about them. The results were slightly better. The whole process took about 1 hour. The results are shown in Figure 12.

4.4.2 Deoldify baseline model

DeOldify was set up on a remote machine with trained weights for the premade "artistic" model. All grayscale images were recolored with the model. Special attention was paid to pictures with the mural visible. An example of such a picture can be seen in Figure 13. The pretrained DeOldify "stable" model was also tested and compared to the "artistic" models results.

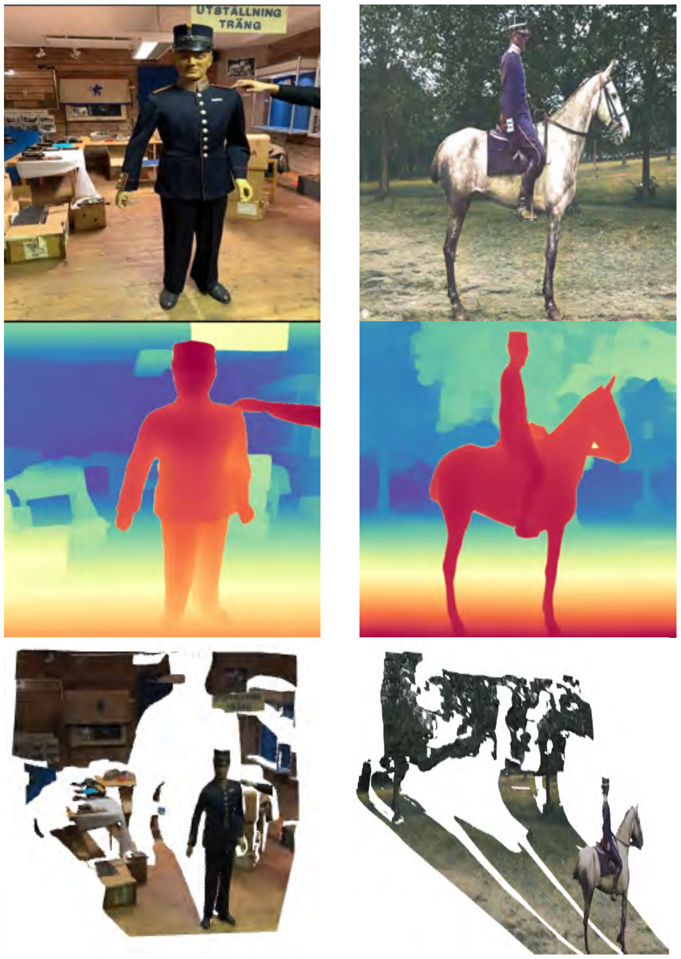

Figure 10. Top left: Photo of an indoor scene. Top right: Image colored with DeOldify of an outdoor scene. Middle row: Depth map generated by MoGe on theese images. Bottom row: 3D model generated by MoGe on theese images

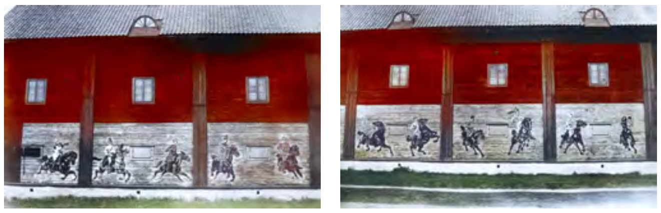

Figure 11. Example of recoloring, with *Colorize*, using two key references (building, grass)

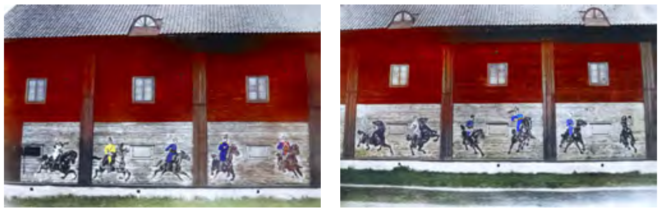

Figure 12. Example of recoloring, with *Colorize*, using two key references and manual image colorization on the costumes

4.4.3 *Finetune generator*

The generator was finetuned by attempting to recreate the colored images from grayscaled copies. These training images were also analysed to better understand the limitations and possible historical inaccuracies present in the model. An example of one of the training images can be seen in Figure 14.

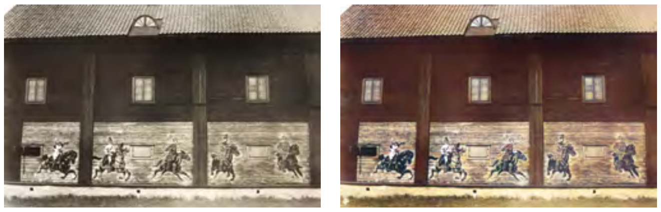

Figure 13. Example of recoloring from default DeOldify model. Original is left and recolored is right.

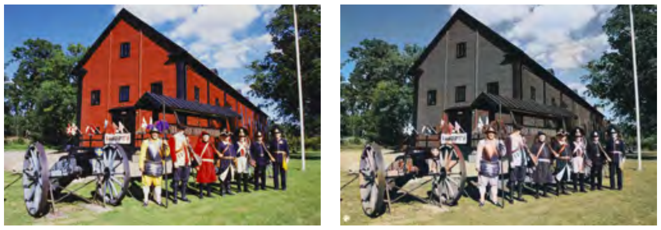

Figure 14. Example of generator attempting to recreate a training image. Original is left and generated is right.

To compare the difference between the fine-tuned image and the original colored image, we calculated the mean squared error (MSE). The MSE provides the mean of the squared differences between the pixels of the two images.

Type of picture	N°	MSE
Group picture with uniform	1	719,76
	4	211,64
	9	541
Postcard	2	140
	3	110,56
	8	165,42
	10	821,35

Table 1
MSE Results

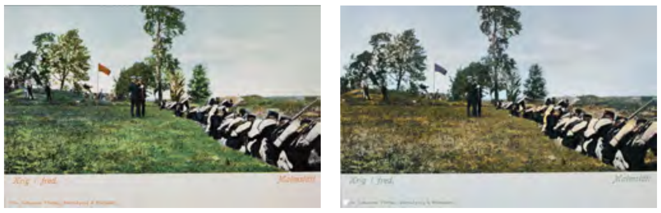

Figure 15. Example of generator attempting to recreate a training image. Original is left and generated is right.

Across 10 different images, the MSE ranged from 110.56 to 821.35, with an average of 382.31. The images with the highest MSE were those featuring buildings where the software failed to correctly convert the red color of the structures, as shown in Figure 14. On the other hand, the images with the lowest MSE were postcards, where most of the colors are not as vibrant as in the other pictures as shown in Figure 15.

4.4.4 Finetune critic

The critic was finetuned by labeling images as either original or recolored. The original were labeled "images" and the recolored were labeled "finetuned image gen". Some transformations were also applied to the images, to reduce risk of overfitting, before being sent to the critic. Some examples of transformations include only using a portion of the image, or mirroring the images. An example of the critics output can be seen in Figure 16.

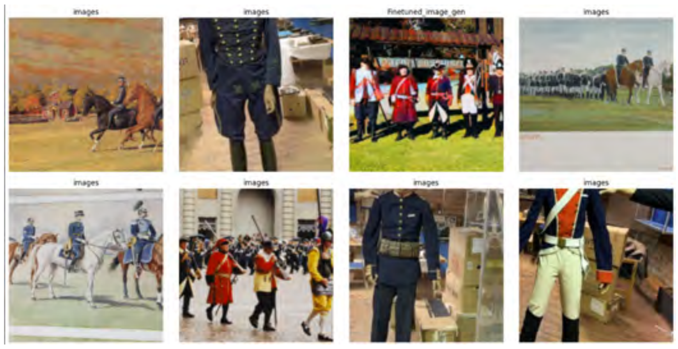

Figure 16. Example of critic labeling training images. The critic is 100% accurate in this prediction.

4.4.5 Finetune model

The finetuned model was used to recolor all grayscale images. The results were compared to the results from the baseline DeOldify model. An example of such a comparisson can be seen in Figure 17. The finetuning process was attempted a total of six times with slightly different parameters.

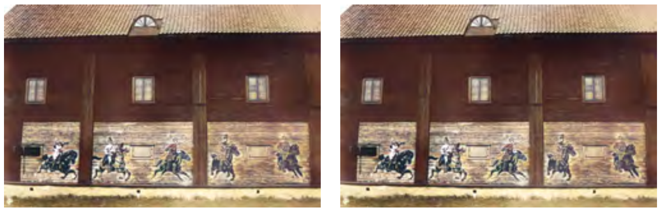

Figure 17. Comparison between original model (left) and finetuned model (right).

4.5 Visualization

The final step of the project involved importing all the processed assets into Unreal Engine to create an immersive virtual environment. The reconstructed building, complete with the mural, was integrated into the virtual museum, along with the recolored paintings and uniforms. The final result can be found in the Figure 18.

5 Evaluation

This Section covers the performance and limitations of the various methods and tools used in the project. The subsections highlight challenges encountered, outcomes achieved, and potential improvements for future work.

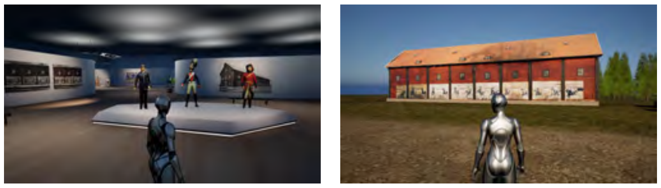

Figure 18. Images from the finished virtual museum.

5.1 Building reconstruction

The obstruction of the trees on the backside of the building led to challenges in achieving a complete reconstruction. Additional images were required to supplement the initial dataset to improve the quality and to fill any holes that were evident. This shows the importance of capturing multiple different angles of the object when doing a reconstruction.

5.2 Robust Dense Feature Matching

The RoMa method performed well in matching features between images, especially for the mural and the surrounding walls. These elements were accurately aligned even when the images varied in quality, which highlights the model's robustness. However, the roof posed some challenges, with fewer reliable matches identified in that area. Overall, the method showed good consistency and was effective for this part of the project. For future work it would be interesting to use this method (or similar) to also generate 3D models.

5.3 Monocular Depth Estimation

The depth estimation and 3D models MoGe produced were surprisingly accurate, capturing fine details of both indoor and outdoor scenes effectively. Example of this in Figure 10. There is still room for improvement, for example with rounded surfaces. The tool was straightforward to use, making it a practical choice for this project. However, some of the generated models were overly detailed, leading to memory issues that made them difficult to process. To address this, additional steps in MeshLab and Blender were often necessary to trim and simplify the models, to make them manageable for use in the final result.

5.4 Coloring

The differences between the finetuned models and the baseline model, as seen in Figure 14, were imperceptibly small. The model weights were different, however no substantial differences were visible in the images. This problem could have likely been fixed with more time, data, and better chosen hyper-parameters. The only training data used for finetuning was the dataset provided by the municipality. Additional images could most likely have been collected from other sources. The lack of data was very apparent since the baseline model was trained on thousands of images via Imagenet, while the finetuning used less than 25 images. There are also techniques for getting more usefull data from small datasets that could have been employed. One example is mirroring or flipping images and saving them as an additional data-sample. Progress on the recoloring was unfortunatly delayed, leading to less time than desired devoted to finetuning the model.

The model showed clear difficulties with certain colors. Firstly, the color yellow was very difficult to replicate and was instead usually colored white. Secondly, the colors blue and red were often mixed up. The differences can be seen in Figure 14. Both of these issues are easily explained by how similar the colors are in grayscale. This issue is very difficult to completely remove, but could most likely be improved with more finetuning.

A big problem was that the model had a difficult time recoloring the mural. It might have helped to have more training images of artworks. A connected issue was that the group members were unsure of what colors some uniforms in the mural were supposed to be. Some uniforms didn't match any uniforms in the available colored images. This could have also benefited from more data.

6 Conclusion

This project aimed to create a virtual museum by combining new technology with historical content. It focused on rebuilding a historical building, recoloring images in grey scale, and making everything accessible in an interactive 3D space.

This project demonstrates that combining AI and 3D modeling is an engaging way to present cultural heritage. The interactive virtual museum format offers museums a modern tool to attract audiences, maybe younger visitors and maybe people that can not get to a real museum. However, several challenges became apparent. Some building details were incomplete due to obstructive elements like trees. Additional images from varied angles would help address this. Provided that it is still possible to get new images of the object. The AI struggled with certain colors, such as yellow (often misclassified as white) and red/blue (frequently mixed). A larger training dataset could improve accuracy. Simplifying 3D models for smoother performance in Unreal Engine caused some texture blurring. Refining this balance is necessary. The tools used cannot guarantee historical precision, particularly in colorization. Issues like color misclassification and a lack of training data for murals highlighted these limitations. Despite this, the approach is a valuable way to present history interactively, provided the potential inaccuracies are disclosed to viewers.

Integrating the reconstructed building and recolored assets into Unreal Engine successfully created an immersive environment. The 3D models and animations make the experience more engaging adding to the potential for use in digital museums.

Future improvements could be to gather more diverse images to enhance both reconstruction and recoloring accuracy. Expanding the AI training dataset would help improve color classifaication, particularly for challenging elements like the mural in this project. Finally, incorporating additional features, such as guided tours or augmented reality components, could further enhance the interactivity and engagement of the virtual museum experience.

Acknowledgement

We would like to thank Johan Edstedt for supervising our research and providing valuable guidance throughout the project.

We would also like to thank Fredrik Viksten for his support and supervision, which was instrumental in our progress.

Finally, we would like to thank Ted Henriksson for his assistance in setting up the VPN and resolving IT issues to ensure a smooth workflow.

References

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M. and Szeliski, R. (2011). 'Building Rome in a day'. In: *Communications of the ACM* 54, pp. 105–112.

Antic, J. (2024). *DeOldify/README*. URL: https://github.com/jantic/DeOldify/blob/master/README.md.

Colorize (2023). URL: https://helpx.adobe.com/photoshop/using/colorize.html.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009). 'ImageNet: A Large-Scale Hierarchical Image Database'. In: *CVPR09*.

Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M. and Felsberg, M. (2023). *RoMa: Robust Dense Feature Matching*. arXiv: 2305.15404 [cs.CV]. URL: https://arxiv.org/abs/2305.15404.

Epic Games (2024a). *FAB Library*. Available: https://www.unrealengine.com/fab.

— (2024b). *Unreal Engine*. Available: https://www.unrealengine.com/.

Johan Edstedt, p. (2023). *RoMa*. https://github.com/Parskatt/RoMa.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A. and Bojanowski, P. (2024). *DINOv2: Learning Robust Visual Features without Supervision*. arXiv: 2304.07193 [cs.CV]. URL: https://arxiv.org/abs/2304.07193.

Schönberger, J. L. (2024). *COLMAP Documentation*. Accessed: 2024-11-28. URL: https://colmap.github.io/.

Schönberger, J. L. and Frahm, J.-M. (2016). 'Structure-from-Motion Revisited'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113. DOI: 10.1109/CVPR.2016.445.

Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X. and Yang, J. (2024). *MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision*. arXiv: 2410.19115 [cs.CV]. URL: https://arxiv.org/abs/2410.19115.

Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H. and Cao, Y. (2022). *Revealing the Dark Secrets of Masked Image Modeling*. arXiv: 2205.13543 [cs.CV]. URL: https://arxiv.org/abs/2205.13543.

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J. and Zhao, H. (2024). *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data*. arXiv: 2401.10891 [cs.CV]. URL: https://arxiv.org/abs/2401.10891.

Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S. and Shen, C. (2020). *Learning to Recover 3D Scene Shape from a Single Image*. arXiv: 2012.09365 [cs.CV]. URL: https://arxiv.org/abs/2012.09365.