EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2018)
R. Sablatnig and M. Wimmer (Editors)

Proxy Painting

Vanessa Lange[†1] Philipp Kurth[1] Benjamin Keinert[1] Martin Boss[2] Marc Stamminger[1] Frank Bauer[‡1]

Friedrich-Alexander University Erlangen-Nürnberg (FAU)
[1]Visual Computing [2]Classical Archeology

Figure 1: *Proxy painting enables artists to paint onto a proxy object (center) in a permanent (real paint) or nonpermanent (digital paint) way. The painting is projected onto the original unmodified target object using projection mapping (left, right).*

Abstract

For archaeologists it is often desireable to present statues in their original coloration. With projection mapping real-world surfaces are augmented by digital content to create compelling alterations of the scene's visual appearance without actually altering or even damaging the object. While there are frequent advances in projection quality, content creation is still a challenging and often unintuitive task, especially for non-experts. In our presented system we combine the advantages of digital content creation such as rapid prototyping with the convenience of an analog workflow. Users paint on smaller versions of the projection mapping target, employing real-world brushes and pencils, while the results are presented live on its large counterpart. We further demonstrate the integration of our system into a state-of-art game engine. By leveraging a powerful rendering and material workflow we make creating compelling materials and lighting situations an intuitive experience.

CCS Concepts

•Human-centered computing → Mixed / augmented reality; •Computing methodologies → Mixed / augmented reality;

1. Introduction

The original coloration of ancient statues is an important topic in archaeology. Larger museums therefore often paint 1:1 scaled copies of a statue. This approach however is infeasible for smaller institutions or in an educational context where different coloration variants need to be discussed every term. Nonpermanent alternatives can solve this problem. Spatial Augmented Reality (SAR) using dynamic multi-projection mapping enables users to project digital content onto movable and complex shaped real-world target objects. Hence, instead of actually painting, the archaeologist could simply *project* the color onto the statue. However, two major problems prevent non-experts from leveraging such a technique:

1. Projection mapping is neither a common nor easily accessible output method.
2. Familiar analog and haptic painting workflows, acquired while painting real objects, are not applicable to digital 3D content creation (necessary to drive a projection mapping system).

In this paper we propose solutions to both problems. Our projection mapping system is flexible enough to integrate into existing tools, e.g. game engines, providing a user-friendly editor and up-

† vanessa.lange@fau.de
‡ frank.bauer@fau.de

DOI: 10.2312/gch.20181345

to-date computer graphics techniques. The latter is an important aspect for believable color reproduction as this includes both the paint and the original lighting setting of a statue. Similar to Virtual Reality (VR), our plugin extends a readily available, state-of-the-art game engine (Unreal Engine 4) by just another output device – a fully dynamic multi-projection mapping system based on the method described by Siegl et al. [SCT*15]. This integration seamlessly combines the ease of use and professional feature set of the engine with the advantages of projection mapping to easily augment the appearance of a real-world target object.

Our main contribution is a proxy painting system that specifically targets users without any prior knowledge about computer graphics, computer science or digital content creation and relies on affordable consumer-grade hardware. The users create the digital projection mapping content by physically painting a proxy object – a smaller version of the target object (e.g. 3D print of the statue) – with *real* brushes and *real* paint (see Figure 2).

While pure analog painting is accessible to everyone, it misses many benefits of a digital workflow (e.g. undo, erase). We introduce nonpermanent painting where no real color is applied to the proxy object. Instead, only the projection mapping target is digitally augmented. This approach of painting 3D objects is user-friendly and saves resources as the proxy remains unmodified. Therefore, one can think of the real-world proxy object as a complex shaped graphics tablet (see Figure 1).

2. Previous Work

Projection mapping is applied in various fields such as education, advertisement or product design. 2D or 2.5D projection mapping covers planar/planar-like target objects or surfaces (e.g. façades). Bimber et al. give a comprehensive summary of various challenges to this type of projection mapping [BIWG08]. As a real-world application, Fischer et al. demonstrate up-to-date castle-sized façade mapping with user interaction elements [FvdHH*15] .

3D projection mapping with multiple projectors is applicable to arbitrarily shaped objects and was first introduced by Raskar et al. [RWLB01]. Since then, techniques were developed to also support projection mapping setups that surround the user, e.g. with *IllumiRoom* [JBOW13] and *RoomAlive* [JSM*14]. Complex objects usually require multiple projectors to cover their surface. Siegl et al. introduced the first multi-projector setup for movable and arbitrarily shaped objects while maintaining a geometry-aware consistent luminance level across projector frusta [SCT*15]. For improved tracking of the target object, Asayama et al. developed infrared (IR) markers that are nearly invisible within a projection but trackable by IR sensors [AIS18]. The recent state of the art report by Grundhöfer et al. provides a more detailed summary of the field [GD18].

Content creation for projection mapping is usually left to designers with some expertise in projection mapping. Although software exists to help with the technical and various design aspects (e.g. MadMapper, FaçadeSignage), the tools are limited to more basic target shapes (2D or 2.5D) or are single purpose. Interactive painting systems for arbitrarily shaped objects allow users to digitally draw onto the target object by handling a 3D controller similar to a brush [BRF01] or airbrush pistol [LSC*16]. While such digital

(a) *Projection Mapping Setup* **(b)** *Proxy Painting Setup*

Figure 2: *In the projection mapping setup, multiple projectors face the target object that is tracked via a depth camera. In the proxy painting setup, the proxy object, a scale invariant copy of the target object, is captured by an RGB-D camera.*

tools provide ways for generating projection mapping content, they remain inaccessible compared to analog painting, since they provide either a simple and limited or powerful but complex interface containing many buttons and functions. Moreover, precise painting is hardly possible, since a certain distance to the target object must be kept to prevent occlusion. Although methods exist to remove occluders from 3D projection mapping [ZXT*16] [LSC*17], precise painting is often performed merely millimeters away from the target object. Thereby all projectors that illuminate this region are blocked which renders shadow removal impossible in most practical setups.

In this work we focus on content creation that is as intuitive as possible even for users with no prior experience in the digital realm. We exploit the fact that anyone can use color and a brush to paint an object. Various methods exist for capturing handwriting or drawing on a *planar* surface. Munich et al. restrict the shape of the pen's tip and detect it via edge detection; the ink path provides information about actual contact with the paper [MP02]. In contrast, an inkless pen drawing or imitating mouse clicks is detected by Pfeiffer et al. by tracking the pen's shadow [PMdO14]. However, current methods are not immediately applicable to complex shaped 3-dimensional surfaces due to inhomogeneous lighting and self-shadowing. We demonstrate painting a complex shaped surface using real brushes or brush-like tools.

For more advanced users, we further assist with content creation by integrating our proxy painting into Unreal Engine 4. Its user-friendly and powerful editor offers (among others) animations, dynamic lighting and a material editor.

3. Projection Mapping

One of our projection mapping setups is depicted in Figure 2a. The setup consists of the target object and several projectors illuminating different sections of it. For geometry-aware projection mapping, a high quality mesh of the target object is required. Further, in order to automatically locate the target object, we employ depth-based tracking. As our system includes automatic calibra-

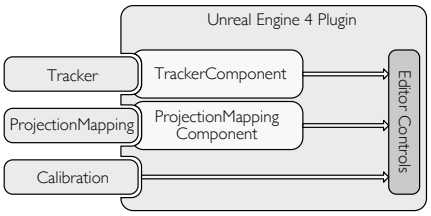

Figure 3: *The Unreal Engine 4 plugin includes libraries for tracking, the projection mapping luminance solver and calibration.*

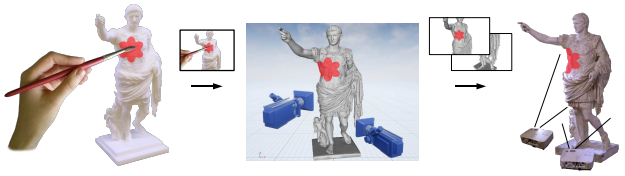

Figure 4: *In every frame we capture an image of the painted proxy object. The recorded color is applied to the projection mapping model that is then rendered for each projector's point of view. The multi-projection mapping system sends the luminance-adjusted images to the projectors that finally project the painting onto the target object.*

tion [KLS*18], we require an additional RGB camera facing the target object.

Based on the projection mapping setup's calibration and by tracking the projection target object, we automatically construct a virtual scene that replicates the real-world setup. The target object is represented by the high quality mesh (model) while the projectors are described by virtual cameras. For projection mapping, the model is rendered by each virtual camera. The resulting G-Buffers (depth, normals) and the rendered color serve as input for the multi-projection mapping's solver that optimizes per-pixel luminance [SCT*15] [LSC*17]. The optimized luminance values ensure that the projection adjusts to the target object's geometry and that overlapping projector frusta maintain the correct brightness level. Finally, the rendered and luminance-adjusted images are sent to the projectors. Thus, the projection is an exact copy of the virtual appearance.

We integrated our projection mapping system into Unreal Engine 4 as a plugin. Its ready availability in a state-of-the-art, off-the-shelf game engine provides many helpful tools for taking projection mapping to the next level. Figure 3 depicts our integration. The functionality for depth-based tracking of an object, the projection mapping luminance solver, and calibrating the projection mapping setup are provided as pre-compiled libraries to the plugin. In order to give the user as much configurational freedom as possible, any parameters are exposed to the editor. The editor is particular helpful for visual assessment and modification of the scene.

Our projection mapping system is designed to be easily embedded into existing output pipelines. The projection mapping luminance solver by Siegl et al. [SCT*15] requires each projector's intrinsics and extrinsics as well as the per-pixel surface normal and world position. Since the world position is easily reconstructed using a projector's camera matrix, it is sufficient to create a single buffer containing the surface normal and corresponding linear depth. In Unreal Engine 4 this information is available through the geometry rendering pass. In order to compose the final output color the solver further requires the target surface color (including all lighting effects in the virtual scene). This buffer is provided by the deferred lighting pass.

4. Permanent Proxy Painting

The projection mapping setup is complemented by the proxy painting environment (see Figure 2b). A second version of the target object of any size is required as the proxy object for painting. Its surface color is captured by an RGB-D camera. The additional depth channel enables tracking the proxy object relative to the camera.

We apply a material to the virtual model that replicates the painting on the real-world proxy object. The workflow for permanent proxy painting is depicted in Figure 4. Every frame, the RGB-D camera captures a color image in which the proxy is (partially) visible. Further, the proxy's position towards the camera is determined by aligning the mesh to the additionally recorded depth data. The model is then colored by assigning each surface point its corresponding pixel color from the color image capturing the real-world proxy. Sections of the model not visible to the camera remain without counterpart in the color image. For example if the RGB-D camera captures only the front of the proxy no color information for the back is available. Such cases are usually detectable by performing a depth check between rendered and recorded depth. When a certain threshold is exceeded the section's color is not updated and maintains the last stored color. In doing so, only valid colors are ever applied to the model and sections remain colored even if no updated information is available.

Color reproduction is an important aspect in this context. In order to achieve correct results, we color-calibrate the RGB-D camera beforehand [CHTD13].

When painting the proxy interactively, while recording the color, users can showcase their painting workflow. Therefore, we automatically remove occluders (e.g. painting tools) from the projection result (see Section 6). However, camera visibility is of concern. Artists have to take special notice that the camera can track and record the color of the proxy. This may impair the artist's movement. In this case, the artist is free to paint the proxy off-camera. When finished, exposing the newly painted sections to the camera updates the stored color (and projection) automatically.

In addition to stored colors, the model may be affected by other elements of the virtual scene. For example, animated light sources (real-world or virtual) may continuously change the illumination of the model.

5. Nonpermanent Proxy Painting

In nonpermanent proxy painting, the user does not apply real color to the proxy. Instead, they *virtually* paint the projection mapping model (and thus the projection). Besides saving resources, this way

Figure 5: *A pointing device with a colorful tip is held close to the proxy object. The color is recorded and applied to the model.*

Figure 6: *Three users paint with laser pointers simultaneously.*

of painting combines the advantages of digital content creation, such as rapid prototyping, with a convenient analog workflow.

5.1. Brush Copy Painting

In the Brush Copy mode we replace real brushes and paint with a colorful pointing device (see Figure 5). Whenever we detect a deviation from the proxy surface color, we store the new information. In order to detect the deviation, the ground truth color information of the proxy object is recorded in a step of preparation. The user therefor rotates the proxy in front of the camera such that every surface point's color is captured at least once. During painting, only color that is different from the ground truth, e.g. the brush's colorful tip, is stored. This restriction keeps the virtual painting intact even after the brush is removed and the camera records the unmodified proxy again. Not yet painted sections keep their ground truth color.

5.2. 3D Graphics Tablet

Nonpermanent proxy painting as described above captures color solely based on the real-world proxy environment, i.e. the pointing device's colorful tip and the proxy's ground truth color as recorded by the RGB-D camera. However, difficult or even changing lighting conditions make recording ground truth data complicated. Furthermore, shadows caused by the brush or the artist themselves may be detected as a painting color (see the accompanying video) and impair the result. Therefore, the 3D Graphics Tablet mode omits the concept of real-world ground truth data. Previously, *any* color different from ground truth could paint the model. In this mode, the user explicitly picks the *single* brush color that is allowed to paint. For example, by registering a brush with a red tip, only this shade of red is applied to the model, a green brush would be ignored.

Although switching the brush is supported at any time and previously recorded color is kept, a new brush for each color is impractical. Therefore, the painting system allows the user to substitute the registered brush color. The substitute is picked from a color wheel, determining the actual paint color. To extend the example from above: while only the red brush is eligible to paint (the green brush is ignored), the actual color applied to the model at the brush's position is determined by the color wheel, e.g. blue.

To register a dummy brush, the user holds the brush in front of the RGB-D camera and selects its tip in the recorded image. Since the brush color is acquired in the RGB-D camera's color space, we can easily recognize it in future frames.

In addition to dummy brushes, it is also possible to paint with laser pointers (see Figure 6). While it is our main objective to support analog painting, laser pointer painting proved to be a fun way to highlight areas and play around from several meters away.

As it is tedious to dye an entire model, the user can pick a base coat from the color wheel or camera image. In some cases, artists may wish to augment an existing coloration. Therefore we allow them to acquire the base paint in a similar fashion to ground truth capturing described in Section 5.1.

6. Implementation Details

The previous sections explained proxy painting on an abstract level. In order to achieve high-quality results there are additional technical details to consider.

Color Storage The setup's RGB-D camera captures the real-world appearance of the tracked proxy object. In order to color the corresponding virtual model we map the recorded image to the model's UV space, thus replicating the proxy's appearance. This is achieved by a simple two-pass rendering scheme.

In the first pass we employ a material that maps the model to the UV space (see Figure 7) and derives a fragment color based on the camera image. In permanent proxy painting (see Section 4) the actual fragment color is determined by performing a color lookup in the camera image via reprojection. This results in a texture containing the camera's projection onto the model. If the fragment's surface point is not visible to the camera, the new color information is discarded, ensuring that previously stored information is not overridden by invalid data. In Brush Copy painting additional restrictions are imposed on a fragment. New color information from the camera image is only adopted if it differs sufficiently from the fragment's previously recorded ground truth color. In our setup (see Section 7.1) we consider a fragment's color similar to ground truth if each color channel differs at most 20 % from its corresponding ground truth value. For more robust results we include ground truth values from neighboring fragments. Finally, in the 3D Graphics

Figure 7: *The model on the left is colored according to its local coordinates within a bounding box. Once the material contains a World Position Offset derived from its UV coordinates, the model on the right is unwrapped in the scene, while maintaining the colors based on the original position.*

Figure 8: *Dilating the texture islands removes the artifact seams.*

Tablet mode a user-defined color (e.g. from color wheel) is written whenever a predefined color is detected in the camera image. While we could extend the shader code of Unreal Engine 4 to accomplish this task, we chose to use the *World Position Offset* (for details see [Bru16]) of a simple Unreal Engine Material.

For the second render pass, this texture is applied to the unmodified model. The scene is rendered from each projector's perspective with all lighting effects enabled. However, sampling artifacts appear when no further measures are taken (see Figure 8). Cracks emerge at seam lines between areas covered by different texture islands. A common technique to overcome this issue is to dilate the islands. Therefore, inspired by [Bru16], the model's sampling material adopts neighboring values if the texture sample is prone to causing seam artifacts. In our setup we dilate by four texels on a 1024×1024 texture.

Depth Culling and Dilation We allow the artist to interactively paint the proxy object. In order to remove occluders (e.g. brush, fingers) from the projection mapping result we cull such objects by performing a depth check between rendered and live depth from the RGB-D camera. However, depth-based tracking using consumer-grade hardware suffers from precision errors. As a result, the model of the tracked proxy is not perfectly aligned with the live depth image but exhibits a slight offset. This tracking offset also impacts depth culling and thus parts of the occluder remain visible (see Figure 9). We solve this issue by dilating the live depth image before culling. Smaller depth values (i.e. closer to the camera) override greater values in their neighborhood. Thus, the region in which occluders are detectable grows and the offset is covered.

Figure 9: *Due to depth-tracking-based imprecision, depth-based culling misses parts of the occluding pen and crops the model's silhouette (top right). These artifacts are removed when dilating the depth image from the depth camera (bottom right).*

Edge Detection and Removal Brush Copy painting transfers the color of a pointing device (brush), that is held closely in front of the proxy object, onto the projection mapping model. Any color that differs from a previously recorded ground truth color texture is applicable. While this technique provides the desired effects for the brush's center, artifacts appear at the edges (see Figure 10). They are for example caused by aliasing in the color image (pixel not 100 % per covered by the brush). This color variation may exceed the ground truth threshold and is thus saved into the painting. As a consequence, when moving the brush, the undesired edge color paints over the desired brush color.

The proximity of the brush to the proxy prevents a reliable depth check to filter the edges (at least with consumer-grade hardware). Therefore, we depend entirely on the recorded color image. We determine the distance d to the closest brightness edge within a certain region for each pixel, employing a Laplace edge detection kernel. For this sampling region, we found a radius value r of 10 – 16 texels (depending on the brush size) to be sufficient, with a camera image resolution of 1920×1080 pixels. This distance d then provides a value α to blend between the current and previously recorded color value for this pixel:

$$\alpha = smoothstep(0, r, d)$$

Figure 10 shows the edge detection result and the effects of edge removal in the Brush Copy mode. With regard to performance we only sample a sparse neighborhood of each pixel.

7. Results

Our proxy painting was tested with a real-world multi-projection mapping setup. In this section we detail our hardware, evaluate the result quality and discuss the overall pros and cons of our system.

(a) *Edge Detection* **(b)** *Edge Removal*

Figure 10: *In (a) the detected edges of the pointing device held in front of the cuirass and many false positives are depicted in black. The bottom image in (b) demonstrates how the artifacts from the top image vanish when lowering the influence of edges.*

Figure 11: *The painting of the colored proxy object on the left is transferred onto the projection mapping target object on the right.*

7.1. Hardware

Figure 2a shows one of our projection mapping setups. An approximate 2 m big plaster cast of the statue of Augustus Prima Porta serves as the target object. Alternatively, we also project onto a 56 cm Augustus bust. Up to three `NEC NP-P451WG` projectors (1280 × 800 pixels) are set up to cover a large part of their respective surface. An `Intel RealSense SR300` camera (1920 × 1080 pixels) automatically tracks the target object's position. The hardware is connected to a standard desktop workstation with an `Intel Core i7-6700K` (4.00 GHz) CPU, 64 GB of RAM and an `NVidia GeForce GTX 1070` graphics card.

The proxy painting setup (see Figure 2b) features a 20 cm 3D print of the statue/bust. The proxy object is tracked by an additional `Intel RealSense SR300` camera. While the tracking relies on the camera's depth channel, the recorded and color-calibrated [CHTD13] images capture the proxy's appearance.

7.2. Quality

For a demonstration of interactive proxy painting we refer the reader to the accompanying video. Figure 11 demonstrates results from permanent proxy painting. The color of the painted proxy is accurately captured and projected onto the target object.

Brush Copy painting is depicted in Figure 12. The artist is equipped with several dummy brushes that feature a colorful tip. Instead of actually drawing onto the proxy, the user only touches the surface with the tip of the brush which transfers the brush's color onto the projection mapping target object. Although the different brush colors are transferred to the model, some artifacts impair the overall painting quality. The shadows caused by the brush and artist's finger exceeds the ground truth threshold and thus overrides the existing painting which leaves undesired stains behind.

Figure 13 highlights results from the 3D Graphics Tablet mode with area and precision drawings. In contrast to Brush Copy painting, no artifacts are caused by the brush or the artist. Instead, this mode generates the best looking and most robust painting results while giving the user total freedom in the choice of brushes and paint color.

7.3. Discussion

A great benefit of our system is the affordable consumer-grade hardware and its usability, as it requires no expertise in computer graphics, computer science or digital content creation.

If the user decides to interactively paint the proxy, we automatically remove occluders (e.g. painting tools). Since our culling approach is depth-based and relies on a threshold value, our system cannot remove the entire tool. Usually, the tool's tip is visible in the projection as it is close to the surface and does not exceed the threshold. However, users found this behavior helpful, since they could use their tools as pointing devices within the projection. If this behavior is not wanted, more sophisticated culling approaches could be implemented, e.g. by tracking the brush itself.

In Section 3 we state that the projection is an *exact* copy of the virtual appearance. Since the RGB-D camera is color-calibrated, real-world colors are recorded and processed as close to the original color as possible. However, adjusting the projection for real-world global illumination is still an open research topic in dynamic multi-projection mapping for complex shaped objects. Siegl et al. compensate for various physical stray-light effects, improving the projection black-level, inter-reflections and adapting for environmental light [SCSB17].

At any point, the user can modify the virtual scene in Unreal Engine's editor in order to affect the projection mapping result (e.g. change the lighting situation). The editor also supports saving and loading their painting (accumulated color texture).

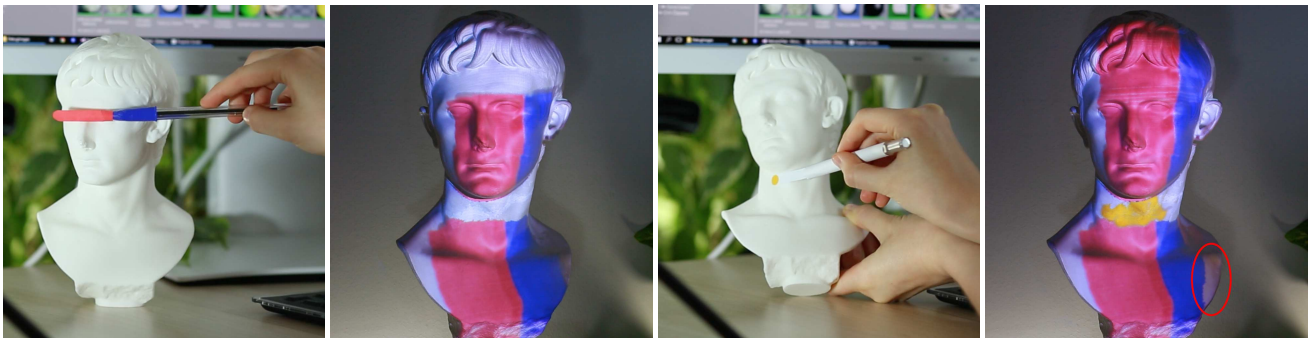

Figure 12: *In this example of Brush Copy painting, the user moves a multi-colored (red, blue) dummy brush across the proxy. The same colors are stored in the painting. The throat is unmodified as it is not visible to the RGB-D camera. In a new pose this gap is then filled in with a yellow brush. Parts of the artist's finger are visible on the right shoulder (circled in red).*

Figure 13: *In the 3D Graphics Tablet mode, the user touches the proxy with a registered yellow dummy brush (different tip sizes). The actual painting color is set digitally (e.g. multicolored in the left example). No artifacts are caused by the artist's fingers or the dummy brush.*

Overall, depth-based tracking the proxy object proves to be our greatest weakness. If the user moves the proxy too quickly or occludes too much of the object, tracking fails and the painting setup has to be realigned. While this does only take a few seconds, it breaks the immersion for the viewers.

Currently, all hardware is connected to a single workstation. However, our system is flexible enough to be divided into a local projection mapping and a remote proxy painting setup. Thus, it would be possible to paint the proxy object at a different location than projecting onto the target object. In addition to collaborative work, we imagine a new and innovative way of student-teacher communication. For example, a classroom may be equipped with one proxy painting setup per student. The teacher may then select which painting they want to demonstrate using projection mapping.

Nonpermanent proxy painting is an interesting new way of creating textures for 3D models. In the future, 3D printers may be an inherent part of every household, such that 'quickly printing a new proxy' might not pose a problem anymore. However, today 3D printing a high quality model is still expensive. Therefore, not wasting a proxy by only coloring digitally is desireable. In our setup, Brush Copy painting works best with bigger brush tips (wider than 0.5 cm). This restriction is imposed by the camera resolution and the necessary edge removal algorithm. One downside of nonpermanent proxy painting in its current basic form is removing the brush from the proxy without continuing to paint. We currently solve this problem by removing the tool towards the camera. Once the depth threshold is reached, the algorithm ignores the tool and no paint is applied anymore. An intelligent brush that detects the small pressure while touching the proxy would solve this issue. Alternatively, a foot pedal would provide a convenient way for the user to en- or disable painting. In the 3D Graphics Tablet mode, the user can employ a double-sided brush with different colors where only one of these is registered. By flipping the brush side, the user switches to the not registered color on the back and can conveniently remove the brush from the proxy without painting.

The 3D Graphics Tablet mode could easily be extended to support painting with materials instead of plain color. In its simplest form this is achieved by painting with gloss weights (the higher the glossier) or shader weights (e.g. 50 % metal on top of 50 % wood). Furthermore, with some adjustments to the existing system painting could become independent from the dummy brush's shape. Given the dummy brush's center, any brush shape can be applied to the virtual model at this position. For example, the user could paint with a circular brush tip but actually draw with a star shape.

We demonstrate proxy painting in the context of projection mapping. However, we think that this technique is also a valid method for general-purpose texture painting as is often required when designing products or digital assets (e.g. virtual museum, game assets). Nonpermanent proxy painting is especially useful if the proxy is reuseable for multiple designs.

7.4. Reality Check

The polychromy of the statue of Augustus Prima Porta [Liv03] is an ongoing research project. Neither the original white marble statue in the Vatican Museums at Rome nor a plaster cast, taken from this statue during the second half of the 19th century, may be endangered by physically applying color on the surfaces. Manufacturing a series of new life-size plaster casts and colorizing them is not affordable. Instead, downscaled (1:10) 3D prints of the statue, as well as 1:4 prints of various portrait busts of Augustus, are inexpensive to buy. These may even be damaged or destroyed during the colorizing process with no significant loss in budget.

For reconstructing and testing the various colorations of the Augustus statue both modes of the presented proxy painting system are put into use: Permanent proxy painting allows for analog painting methods and applying reconstructed recipes of paint. Nonpermanent proxy painting simplifies testing the multitude of various possible shades of one specific color, e.g. the emperor's blond hair. Physically painting the small-scale proxy – lightweight and easy to handle – is a simple task and requires approximately half a day. Being able to see the results not only on the small proxy, but on the original piece of art itself, and conveniently adjusting it further with nonpermanent proxy painting provides a new convenient tool in reconstructing ancient polychromy.

8. Conclusion

We presented an easy-to-use proxy painting system for texture creation in the context of projection mapping. Uninstructed users without prior knowledge in digital content creation are able to paint a proxy version (e.g. 3D print) of the projection mapping target object. The painting is automatically captured by a camera and processed into a texture that is then projected onto the target object. We further introduced nonpermanent proxy painting where the proxy only serves as the drawing pad without actually being modified – similar to a graphics tablet. These simple and – for the user – analog ways of creating projecting mapping content serve as a basis for many interesting applications in cultural heritage, education or product design.

References

[AIS18] ASAYAMA H., IWAI D., SATO K.: Fabricating diminishable visual markers for geometric registration in projection mapping. *IEEE Transactions on Visualization and Computer Graphics 24*, 2 (Feb 2018), 1091–1102. doi:10.1109/TVCG.2017.2657634. 2

[BIWG08] BIMBER O., IWAI D., WETZSTEIN G., GRUNDHÖFER A.: The Visual Computing of Projector-Camera Systems. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library. doi:10.1145/1401132.1401239. 2

[BRF01] BANDYOPADHYAY D., RASKAR R., FUCHS H.: Dynamic shader lamps : painting on movable objects. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality* (2001), pp. 207–216. doi:10.1109/ISAR.2001.970539. 2

[Bru16] BRUCKS R.: Automated and Improved UV Dilation. http://shaderbits.com/blog/uv-dilation, Sep 2016. visited 2018-05-23. URL: http://shaderbits.com/blog/uv-dilation. 5

[CHTD13] CHARRIÈRE R., HÉBERT M., TRÉMEAU A., DESTOUCHES N.: Color calibration of an rgb camera mounted in front of a microscope with strong color distortion. *Appl. Opt. 52*, 21 (July 2013), 5262–5271. doi:10.1364/AO.52.005262. 3, 6

[FvdHH*15] FISCHER P. T., VON DER HEIDE A., HORNECKER E., ZIEROLD S., KÄSTNER A., DONDERA FELIX ANDWIEGMANN M., MILLÁN F., LIDEIKIS J., ČERGELIS A., VERDE R., DREWS C., FASTNACHT T., LÜNSDORF K. G., MERAD D., KHOSRAVANI A., JANNESAR H.: Castle-Sized Interfaces: An Interactive Façade Mapping. In *Proceedings of the 4th International Symposium on Pervasive Displays* (New York, NY, USA, 2015), PerDis '15, ACM, pp. 91–97. doi:10.1145/2757710.2757715. 2

[GD18] GRUNDHÖFER A., DAISUKE I.: Recent Advances in Projection Mapping Algorithms, Hardware and Applications. *Computer Graphics Forum 37*, 2 (2018), 653–675. doi:10.1111/cgf.13387. 2

[JBOW13] JONES B. R., BENKO H., OFEK E., WILSON A. D.: IllumiRoom: Peripheral Projected Illusions for Interactive Experiences. In *ACM SIGGRAPH 2013 Emerging Technologies* (New York, NY, USA, 2013), SIGGRAPH '13, ACM, pp. 7:1–7:1. doi:10.1145/2503368.2503375. 2

[JSM*14] JONES B., SODHI R., MURDOCK M., MEHRA R., BENKO H., WILSON A., OFEK E., MACINTYRE B., RAGHUVANSHI N., SHAPIRA L.: RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (NY, USA, 2014), UIST '14, ACM. doi:10.1145/2642918.2647383. 2

[KLS*18] KURTH P., LANGE V., SIEGL C., STAMMINGER M., BAUER F.: Auto-Calibration for Dynamic Multi-Projection Mapping on Arbitrary Surfaces. In *IEEE Transactions on Visualization and Computer Graphics (To appear)* (2018). 3

[Liv03] LIVERANI P.: Die Polychromie des Augustus von Prima Porta, vorläufiger Bericht. *Neue Forschungen zur hellenistischen Plastik : Kolloquium zum 70. Geburtstag von Georg Daltrop* (2003), 121–140. 8

[LSC*16] LANGE V., SIEGL C., COLAIANNI M., KURTH P., STAMMINGER M., BAUER F.: Interactive Painting and Lighting in Dynamic Multi-Projection Mapping. In *Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2016* (2016), Springer, Cham, pp. 113–125. doi:10.1007/978-3-319-40651-0_10. 2

[LSC*17] LANGE V., SIEGL C., COLAIANNI M., STAMMINGER M., BAUER F.: Robust Blending and Occlusion Compensation in Dynamic Multi-Projection Mapping. In *Eurographics 2017 - Short Papers* (2017). doi:10.2312/egsh.20171000. 2, 3

[MP02] MUNICH M. E., PERONA P.: Visual input for pen-based computers. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 3 (March 2002), 313–328. doi:10.1109/34.990134. 2

[PMdO14] PFEIFFER G. T., MARROQUIM R. G., D. OLIVEIRA A. A. F.: Webcampaperpen: A low-cost graphics tablet. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images* (Aug 2014), pp. 87–94. doi:10.1109/SIBGRAPI.2014.54. 2

[RWLB01] RASKAR R., WELCH G., LOW K.-L., BANDYOPADHYAY D.: Shader Lamps: Animating Real Objects With Image-Based Illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques* (London, 2001), Springer-Verlag. doi:10.1007/978-3-7091-6242-2_9. 2

[SCSB17] SIEGL C., COLAIANNI M., STAMMINGER M., BAUER F.: Adaptive stray-light compensation in dynamic multi-projection mapping. *Computational Visual Media 3* (2017), 263–271. doi:10.1007/s41095-017-0090-8. 6

[SCT*15] SIEGL C., COLAIANNI M., THIES L., THIES J., ZOLLHÖFER M., IZADI S., STAMMINGER M., BAUER F.: Real-time Pixel Luminance Optimization for Dynamic Multi-projection Mapping. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 237:1–237:11. doi:10.1145/2816795.2818111. 2, 3

[ZXT*16] ZHOU Y., XIAO S., TANG N., WEI Z., CHEN X.: Pmomo: Projection Mapping on Movable 3D Object. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, ACM, pp. 781–790. doi:10.1145/2858036.2858329. 2