High-Quality Tactile Paintings†

A. Reichinger[1], S. Maierhofer[1] and W. Purgathofer[1,2]

[1]VRVis Forschungs-GmbH, Austria [2]Vienna University of Technology, Austria

(a) Raffael's Madonna of the Meadow, dated 1505 or 1506. (b) Layered Depth Diagram. (c) Textured Relief. (d) On display at the museum.

Abstract

The aim of this work is to bring the cultural heritage of two-dimensional art closer to being accessible by blind and visually impaired people. We present a computer-assisted workflow for the creation of tactile representations of paintings, suitable to be used as a learning tool in the context of guided tours in museums or galleries. Starting from high-resolution images of original paintings, our process allows an artist to quickly design the desired form, and generate data suitable for rapid prototyping machines to produce the physical touch tools. Laser-cut layered depth diagrams*, convey not only the individual objects in the painting and their spatial layout, but also augment their depth relations. CNC-milled* textured reliefs *additionally render fine details like brush strokes and texture suitable for the sense of touch. Our methods mimic aspects of the visual sense, make sure that the haptic output is quite faithful to the original paintings and do not require special manual abilities, like sculpting skills.*

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Graphics editors I.3.8 [Computer Graphics]: Applications—I.3.m [Computer Graphics]: Miscellaneous—Visual Arts J.6 [Computer-Aided Engineering]: Computer-aided design (CAD)—

1. Introduction

Cultural heritage in the form of two-dimensional art, like drawings and paintings, is omnipresent in our galleries. Unfortunately, blind or visually impaired people are mostly excluded from this world of visual arts. Most of us take for granted, that original works can be perceived as they are. Accordingly, computer graphics in cultural heritage is typically aimed at methods that provide *additional* insight, or tools for digital acquisition and preservation. Interestingly, many visual computing algorithms turn out to be equally well suited

† Co-funded by "KulturKontakt Austria im Auftrag des BMUKK", performed in cooperation with Kunsthistorisches Museum Vienna.

for tactile media. In the present work, we adopt techniques from the field of visual computing to make paintings "visible" to the sense of touch, to enable visually impaired people to gain insight into our world's cultural heritage.

2. Related Work

Some museums offer special guided tours describing selected paintings verbally [dFI91]. However, it is extremely difficult to create a mental image from acoustic impressions alone. Especially positional information is very difficult to describe, and it is impossible to explore the painting on one's own. Moreover, mental images depend a great deal on the interpretation of the narrator, leaving little freedom for own interpretations, and listeners cannot verify whether their mental depictions are "correct," i.e., as intended by the artist.

Tactile representations are a very useful complement to verbal descriptions. For instance, composition, absolute placement of objects and their relations to each other can be more easily perceived through touch than through verbal explanation. The observer is free to explore regions in more detail according to taste, and has the possibility to verify and correct the mental image. While three-dimensional art like sculptures or architectural models are directly suitable for our three-dimensional touch sense [dFI91], paintings are mostly planar, and, first of all, have to be converted into a touch-able three-dimensional surface. Unfortunately, the creation of such tactile media is a complex task, and to date few dedicated conversion tools are available.

2.1. Tactile Diagrams

A widely adopted technique is the raised line drawing, also known as *tactile diagram*. Using swelling paper or Braille embossing printers in graphics mode, two-dimensional drawings can be made tactile. The Art Beyond Sight collaborative provides useful resources [AL03] and released a multi-sensory art history book series [AG00]. While these diagrams have been drawn manually, several attempts have been made to automate the conversion. The TGA software is specialized for technical drawings and performs automatic text to Braille conversion of annotations [LIR*05, JRW*07]. Similarly, the instant tactile-audio map removes text from maps and presents them auditively on demand. In another project faces are automatically converted to tactile line drawings [WXL08]. These applications are targeted on very specific fields and are not applicable to paintings in general. In contrast, the Tactics software processes *general* images into line drawings by using k-means clustering, edge detection, blurring and median filtering [WB97] and later using watershed segmentation [HB00]. However, generating semantically meaningful line drawings from images is not a trivial task (cf. Sec. 4.2).

Tactile diagrams feature a very important concept, that we adopted in our method: simplification of the image into the structures that are most important for comprehension. While they are very useful for conveying the composition of a painting, the amount of detail is strongly limited. Furthermore, since the medium is practically two-dimensional, painted depth cannot directly be transported.

2.2. Bas-Relief

Various depth cues in paintings like shading or perspective foreshortening indicate the arrangement of objects at different depths. While our visual system is able to interpret these cues, our tactile sense is not. Therefore, visual cues should be conveyed as tactile information to improve depth-perception.

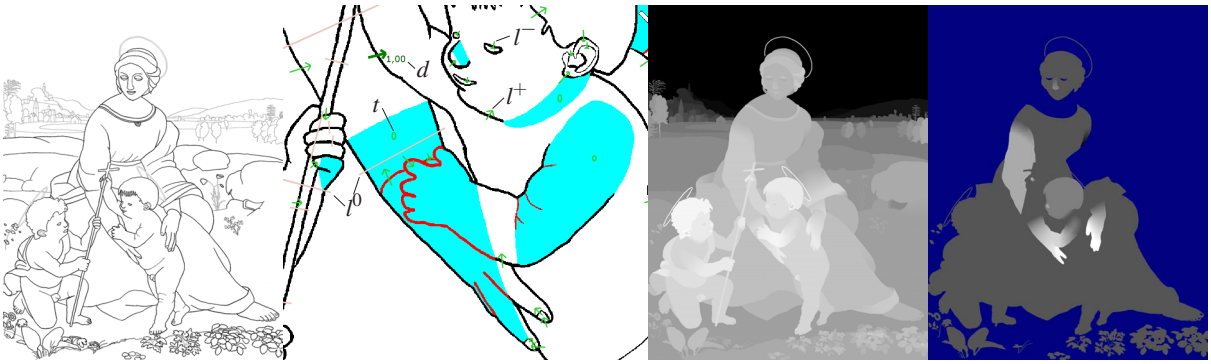
Mikli converts several photos and modern paintings into layers that are realized in cellulose acetate [CBRA09]. The exact process is not disclosed, but it seems to be largely based on color quantization, a process that is only applicable on selected images, with high contrast between depicted objects. The *Museo Anteros* in Bolognia exhibits bas-relief adaptions of several paintings, which have been handmade by professional sculptures. These allow a very detailed haptic perception of the paintings, but require skilled artists and tedious manual work. Recently, they experimented with scanning and reproduction using rapid prototyping machines [OYS10], but the creation process is still manual.

In the visual computing field, techniques are developed that recreate 3D structures from single images. One class of techniques tries to infer depth information through unsupervised learning. Several attempts [SSN09, HEH05, VDH09] first segment the image into super pixels [FH04], and one is based on hand-segmented areas [RT09]. Testing super pixel segmentation on several paintings resulted in suboptimal segments. Furthermore, the quality of the inferred depth depends strongly on the training sets, which are entirely made from photographs of mostly urban scenes and simple geometries. These techniques are not yet suitable for our intended high-quality depth inference in paintings.

Another class are shape from shading techniques. While the basic problem is ill-posed, several semi-automatic techniques have been presented [MBnB07, WTBS07, WSTS08] with promising results. While not yet applied in this work, using such techniques in future work could improve the quality of our results in certain areas (cf. Sec. 5.1).

3. Overview

Our goal is to provide a high-quality tactile experience that, on the one hand, can give a quick overview, while at the same time enables the visitor to explore all the details, according to taste. Since the resulting touch tools are intended to be exhibited in a museum next to the original art works (cf. teaser image d), we additionally want them to be durable and also visually pleasing for sighted visitors. Similar to *tactile diagrams*, the spatial composition of the painting should



(a) Edge-image. (b) Augmenting Depth Relations. (c) Height map. (d) Local height map of layer 17.

Figure 1: Different states in production, as outlined in Section 4.

be conveyed, and the shape of different objects should be clearly perceptible. As outlined in Section 2.2, perception should be improved by conveying depth information.

We conceived a workflow to create two tactile media of increasing complexity. *Layered depth diagrams* (cf. Sec. 4) can be produced using—nowadays widely available—laser cutters. They are inexpensive in production, but have to be hand assembled and are limited to simple structures. *Textured reliefs* (cf. Sec. 5) transport additional texture and brightness variations and allow for arbitrarily complex height fields to be produced with CNC milling machines. The design process for *textured reliefs* directly builds upon the results of the *layered depth diagrams* allowing for an incremental design.

4. Layered Depth Diagrams

A *layered depth diagram* is made of a number of individual shapes cut out of flexible sheets of constant thickness, which are glued on top of each other to resemble a painting. All production steps will be outlined in the following sections.

4.1. Identifying Important Structures

Starting from a high-resolution scan or photograph of the original work, the image needs to be simplified. This is mostly achieved by segmentation into semantically meaningful areas, but sometimes this is not enough. Some features do not have a distinct area although they have a distinct semantic. One example is the infant arm in Fig. 1a. It shares its area with the rest of the body via the shoulder area, where no meaningful separation line can be drawn. At the same time, it is partly separated by the outline forming the shape of the arm from the shoulder down to the fingers. This line arises from the perceived depth discontinuity of the arm held in front of the body, whereas the shoulder part exhibits a smooth depth transition to the body. This is exactly what we want to transport with *layered depth diagrams*. Every depicted part with smooth transitions should end up on the same tactile layer, whereas discontinuities should be made tangible by placing parts on different tactile layers at different heights. In the previous example, this means that the arm has to be on the same surface as the body, but has to gradually bend forward in order to create the desired depth discontinuity with respect to the rest of the body.

Since the tactile sense is easily overstrained by too much detail, it is not sufficient to detect all edges or local discontinuities. In order to convey the desired meaning without causing sensory overload, details have to be omitted and the overall structure has to be simplified. This is a highly creative process, which not only requires semantic understanding, but also artistic interpretation of the original painting. Existing tools and algorithms can not handle such requirements without extensive manual post-processing. Therefore, we decided to manually trace selected boundaries using a standard graphics tablet, which is a very natural way of extracting the desired lines and gives superior quality. This choice was especially underpinned by the desire to create a few high-quality adaptions of paintings as opposed to lower quality mass production.

Concretely, tracing is performed in a separate raster image (the edge-image, cf. Fig. 1a) with the same resolution as the input image: black pixels denote borders, white pixels form the segmented areas a_i. We use a raster image as opposed to a vector format, because it can be more easily exchanged between different programs. Tracing is done using hard edges, without anti-aliasing, since we need a clear delineation of areas in subsequent steps. Still, rounded shapes can be represented sufficiently accurate, since we use a pixel size of approximately 0.2 mm in the final output medium. The separated areas a_i in the edge-image are found with the row-by-row connected component labeling algorithm [SS01] with a 4-connected interior.

4.2. Augmenting Depth Relations

Having segmented the image into areas a_i, we assign appropriate height values h_{a_i} (= the number of layers glued on top of each other). Most height values depend on the height values of other areas, forming a network of height-relations. For example, Madonna's dress in Fig. 1a should be a little higher than her foot, which in turn depends on the height of the floor. Automatic inference of such height-relations is currently a topic of active research [SSN09, HEH05, VDH09] and results are still error prone. We use instead an annotation-based interface, that allows for easy assignment of depth-relations (cf. Fig. 1b): drawing an arrow $l^+_{s,e}$ between two areas establishes an a_e "in-front-of" a_s relation, and drawing a line $l^0_{s,e}$ a "same-height" relation. Starting from the lowest layer, it is straightforward to gradually assign relations in a way to maximize the depth cues for achieving the desired interpretation. This is an interactive process, meaning that after the user manipulates relations, the absolute height values are recalculated, updated and visualized on demand (cf. Fig. 1c). Areas are colored with shades of gray depending on their computed absolute height h_{a_i}, or with a special color if an area is not yet connected to others. Each area may be annotated with multiple arrows, possibly resulting in conflicting height values. In this case, the highest value is chosen to ensure that all annotated depth-discontinuities are satisfied with at least one layer of difference, or at least $d_{s,e}$ layers when the arrow $l^+_{s,e}$ is assigned a higher depth-difference value $d_{s,e}$. The depth-difference value of each arrow is visualized by the size of its arrow head, and additionally by a small number next to it, if the arrow is selected in the user interface.

Every possible depth configuration can be specified using the "push-up" arrows $l^+_{s,e}$. However, it sometimes is desirable to specify the height relative to a higher area (e.g. eyes one layer down from the face layer). In order to efficiently define such relations, we introduce a "pull-up" arrow $l^-_{s,e}$ that brings the area a_e to a height at least $d_{s,e}$ layers below the other area a_s (visualized by an inverted arrow head $\multimap$).

All relations are represented as a directed acyclic graph, where the areas a_i are nodes, and arrows $l^+_{s,e}$ and $l^-_{s,e}$ are directed edges $a_s \to a_e$. Same-height annotations l^0 merge the connected areas into a single node. The solution is similar to the computation of earliest start times in scheduling problems [KW59] and can be efficiently solved by first sorting the nodes in topological order [CSRL01] and then assigning the height $h_{a_i} \geq 0$ of each node a_i in sorted order to be

$$h_{a_i} = \max\left(\max_{j \in I^+_i}\left(h_{a_j} + d_{j,i}\right), \min_{j \in I^-_i}\left(h_{a_j} - d_{j,i}\right)\right),$$

where I^+_i and I^-_i are the sets of incoming push-up and pull-up arrows of area a_i, respectively (more correctly the sets of indices j of areas a_j connected by arrows $l^+_{j,i}$ and $l^-_{j,i}$). In case of conflict, the one yielding the maximum height wins.

The topological sort fails if there are cycles in the graph. In this case, we search for the strongly connected components using Trajan's Algorithm [CSRL01], mark the areas containing cycles in different colors, and ask the user to resolve the cycles.

4.3. Bend-Areas

A cycle in the graph occurs either by user error, or at areas that should be bent (cf. Sec. 4.1). In the second case, the user can resolve the cycle by marking the part where bending should occur as bend-area (filled in cyan, cf. Figs. 1b and 2). This bend-area subdivides the original area into areas that can have different heights, and therefore, effectively breaks the cycle. In order to bridge the new areas, the bend-area smoothly interpolates between their height-values.

We posed two requirements on the interpolator: Firstly, the seam between the bend-area and it's adjacent areas should be at least C0-continuous. And secondly, the interpolation should respect the form of the bend-area. For example, if we have a bend-area in the form of a spiral, there should be a continuous ascent along the spiral. Therefore, the interpolator has to include some notion of distance along the pixels of the bend-area. Concretely, we take the set of adjacent pixels P_{ext} of areas surrounding the bend-area, and group them by their height h_{p_i}, yielding the set of different height values $H = \{h_0, h_1, \ldots\}$, and for each height h_j the subset of pixels $P_{h_j} = \{p_i \in P_{ext} : h_{p_i} = h_j\}$. For each group h_j we perform a distance transform over the pixels of the bend-area p_i to get for each pixel p_i the smallest distance f_{p_i,h_j} to the nearest pixel of the respective subset P_{h_j} (for all pixels $p_i \in P_{h_j}$: $f_{p_i,h_j} = 0$). We use the approximate distance transform as proposed by Borgefors [Bor86] with a 9×9 structure element and standard coefficients. The interpolated height value h_{p_i} at pixel p_i is computed as $h_{p_i} = \sum_{h_j \in H}(h_j \cdot {f_{p_i,h_j}}^{-t}) / \sum_{h_j \in H}({f_{p_i,h_j}}^{-t})$.

With parameter t the form of the interpolation can be adjusted. Concretely, t corresponds to the slope of the interpolation curve halfway between the groups. For two groups, $t = 1$ gives a linear interpolation, and values of $t > 1$ give C1-continuous seams. The user can set the value t for each bend-area separately, by placing an interpolation-type annotation on the bend-area, similar to arrows and lines (Fig. 1b).

If t is set to 0, a planar interpolation mode is used instead. This is for example useful at walls and rooftops of houses, where independent of the form of the bend-area, a planar surface should be used. In this mode, we fit a plane with height $h(x,y) = ax + by + c$ through all pixels $p(x,y) \in P_{ext}$ by linear regression using singular value decomposition. This method should only be used, if a plane can actually be fitted to P_{ext}. Otherwise noticeable seams will occur.

4.4. Discontinuities on Bend-Areas

More complicated cases exist, for example at Madonna's arm in Figure 1b: The arm should bend forward in order

to touch little Jesus' belly, but on top of this bend-area, we want to feel a discontinuity between her arm and sleeve, and between her and Jesus' arm. These bend-areas must not be separated by conventional (black) borders, since this would break a smooth interpolation over the whole bend-area. Instead, we use a different border-type (red) that does not separate the bend-areas. Height-differences between areas separated in this way can then again be annotated with arrows, but in a relative domain specific to the particular bend-area.

To reflect these extensions in the data structures, now every pixel in the edge-image can have one of four states: border/interior (green & blue channel) and flat/bend (red channel). Areas are formed by 4-connected components of each separate combination (flat-interior, bend-interior and bend-border). Bend-interior and bend-border areas are grouped to bend-areas if they are 4-connected over at least one pixel. Lines and arrows are allowed between flat-interior areas and between bend-interior areas of the same bend-group, and their discreet heights h_{a_i} are solved as outlined before.

A *bend-area* can now be composed of multiple areas with different height-values h_{a_i} reflecting their desired relative height differences. The areas with $h_{a_i} = 0$ will belong to the topmost layer that spans the *whole* bend-area, lets call it the *ground-layer*. All areas with $h_{a_i} > 0$ will have h_{a_i} additional layers glued on top of that layer. Accordingly, the interpolation has to be performed on the ground-layer: Instead of working with the heights h_{p_j} of the pixels $p_j \in P_{ext}$ directly, we first subtract the ground-layer offsets of the bend-interior area a_i adjacent to p_j, to get ground-layer heights $g_{p_j} = h_{p_j} - h_{a_i}$ (cf. Fig. 2). After the interpolation of the ground-layer, we add back the offsets h_{a_i} of the respective area to the pixel heights h_{p_j}.

After the interpolation step, the result is a height map of the topmost surface of the *layered depth diagram*, i.e. one height value h_p for each pixel p (cf. Fig. 1c). Since no height was computed for border pixels, we fill them in by interpolating neighboring height values.

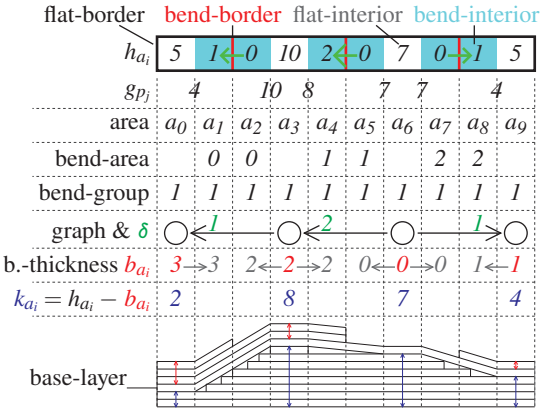


Figure 2: A simple example of the formation of layers.

4.5. Extracting Layers

In order to produce the touchable output, we need to slice the height map into layers of constant thickness that can be cut out of flexible sheets and assembled on top of each other.

For each layer, we basically threshold the height map with the height of the layer h_{l_i}. All pixels p_j where $h_{p_j} \geq h_{l_i}$ delineate the form that has to be cut out for that particular layer. However, bend-areas and discontinuities on bend-areas make the process more complicated (cf. Fig. 2): A bend-area spans multiple layers and can be connected to other areas on different layers, spanning a network of areas on different layers that has to be made from one continuous, bent sheet. We call a connected component of adjacent areas and bend-areas a *bend-group*. In order to be covered with a continuous sheet, the whole surface of a bend-group has to be brought to a single layer, the *base-layer* of the bend-group. Thus, areas of other layers are put onto the base-layer, potentially stealing away the space of areas already present on this layer. These areas actually make up a support structure that shapes the bent surface on top of it. Since the place for these support structures is already occupied, they have to be placed on an additional support-layer. Furthermore, if there are discontinuities on bend-areas, additional layers may have to be glued on the base-layer to form these discontinuities. The following procedure solves all these issues and outputs a set of layers that produce the desired result:

First we compute the *bend-thickness* b_{a_i} for each area a_i as the number of bent layers that have to be glued on top of the base-layer in order to represent all depth-discontinuities inside bend-areas. We form a graph, where flat areas are nodes that are connected by adjacent bend-areas as edges. If a bend-area touches multiple flat areas, all pairs of flat areas are considered for edges. For each pair we compute the value δ as the difference between the relative heights h_{a_i} of the touching bend areas. If $\delta = 0$ the nodes are merged. Otherwise the nodes are connected by a directed edge with a height-difference value $\delta_{i,j} = |\delta|$ pointing from the smaller to the larger h_{a_i} (cf. Fig. 2). With an algorithm, similar to the solution of the push-up arrows in Section 4.2, these height-differences are propagated through the network, until each node has a bend-thickness $b_{a_i} = \max_{j \in N_i} (\delta_{j,i} + b_{a_j})$, where N_i is the set of all incoming nodes to node i. Afterwards, the bend-thickness of areas inside the bend-areas are set to the already computed b_{a_i} of the adjacent areas.

Next, for each bend-group, we need to define which layer to use as the base-layer. Potentially, every layer can be used in which the lowest bent layer has a flat area, i.e. the set of all heights $k_{a_i} = h_{a_i} - b_{a_i}$ of all nodes i in the bend-group. In order to reduce unnecessary cutting, it is best to choose the layer on which the longest connection to other areas exists.

Finally, we have to output the layers. We extract a local depth-image of every main-layer and support-layer at integral heights $h_{l_i} = \{0, \ldots, \max_j(h_{a_j})\}$ according to Algorithm 1. The result is a local height map of the layer (cf.

Fig. 1d) with the following values: $-\infty$ at non-existing parts (blue), 0 at flat parts (gray), and $\neq 0$ where the layer bends up or down. Cuts need to be performed at borders to $-\infty$ (the outlines), and borders between height discontinuities. We perform a modified edge detection to find the cut-lines. These are automatically vectorized using center-line-tracing, resulting in a vector-format suitable for laser cutters. In order to make the best use of the sheets, we tightly pack the individual shapes, and make sure that holes and cut-lines inside shapes are cut before the outlines. The shapes are finally glued on top of each other in the correct order.

Algorithm 1: Extraction of main and support layer.

```
forall the pixels p do
    mainLayer[p] = supportLayer[p] = −∞
    if p ∈ bendGroup then
        if baseLayer ≤ layer ≤ baseLayer + b_p then
            mainLayer[p] = h_p − baseLayer − b_p
            if h_p − b_p − 1 ≥ layer then
                supportLayer[p] = 0
        else if h_p − b_p − 1 ≥ layer then
            mainLayer[p] = 0
    else if h_p ≥ layer then
        mainLayer[p] = 0
```

4.6. Validation

Prototype *layered depth diagrams* were created out of laser-cut polystyrene sheets that were hand assembled and glued together (cf. teaser image b; assembly of the 150 pieces on 25 layers took several hours). Our blind test persons quickly got the overall composition of the painting. Compared to conventional two-dimensional touch-tools, they were able to feel the outlines of the different structures much more easily, were not confused by crossing lines since these occur at different heights, and found it very useful to get an impression of the depth relations in the image. However, large parts of the diagram consist of plain, untextured surfaces. Some test persons were missing information about the look of the painting in these regions. In order to include this information, we developed *textured reliefs* which are described in the following section. *Layered depth diagrams* are currently not on display, mainly because of their rather fragile nature.

5. Textured Reliefs

When observing a painting we do not only perceive its composition, the layout and boundaries of objects, but also fine details like material and surface textures that cannot be transported by our *layered depth diagrams* directly. It would be possible to engrave the sheets in a separate laser-engraving pass with texture information, or to use sheets of different materials. Both options only offer limited capabilities, and are not durable enough to be on permanent display in a museum. Therefore, we decided to use a different production technique for this purpose: carving by a CNC milling machine. This allows us to produce arbitrary height fields that are no longer limited by a fixed-thickness layer structure.

We build on top of the height-fields produced for *layered depth diagrams*, since these already transport the overall structure of the paintings in an easily understandable way. We enrich them with additional information, and finally convert them into control codes for CNC production. Since the layers are no longer restricted to a fixed thickness, arrows with fractional depth-difference values $d_{s,e}$ can be used in *layered depth diagram* creation according to the importance of objects. Furthermore, we optionally mimic perspective by gradually reducing layer thickness in the back by applying a biased gamma function onto the height map.

5.1. Extracting Texture

The human visual system extracts most of the structure information from the luminance channel; the color channels deliver only additional information and have a substantially lower resolution. While color would be interesting, we did not yet attempt to include colors in order to avoid clutter in already detailed images. There are two ways of interpreting brightness variations: First, these can be variations in texture, and second, these can be caused by light interaction on surfaces, giving insight into curvatures, provided the images are physically correct. For the second case, techniques like shape from shading exist, that try to recreate the surface form light interactions [MBnB07, WTBS07, WSTS08]. While this information could be interesting in parts of some images, we rather rely on the more general texture-interpretation that will work on any image. Only where the shading-interpretation is crucial for understanding, we can manually add some shape hints (cf. Sec. 5.2).

The question is, how to convert brightness into meaningful tactile sensations. Using the luminance directly suffers from two problems: High spatial frequency components create high, steep spikes that are unpleasant to the touch, and mask features of lower frequencies. On the other hand, low spatial frequency components, destroy the context of the image by placing objects in a depth according to their average brightness, as opposed to geometrically meaningful depths. We use a logarithmically spaced filter bank with difference of Gaussian (DoG) kernels to roughly separate the image into eight different frequency bands in order to control the frequency spectrum. The DoG kernels effectively removes the steady component, while being rotation invariant. By lowering the high and low frequency components, a pleasant and informative surface can be accomplished. Interestingly, the human visual system has a similar filter characteristic [Bar99], which substantiates our findings. In addition

to filtering, we implemented soft clipping functions on each filtered image to further limit unpleasant peaks.

Filtering, especially with large kernels, has the undesirable side effect of creating noticeable crosstalk between different image regions. Especially border regions and thin objects get severely distorted by brightness variations in surrounding regions, that need to be attenuated. Fortunately, we have a segmentation of the image, stored in the height map: Similar heights will most certainly belong to the same object, even if it is disconnected by other objects. The larger the height difference between two points in the map, the less they should be considered in filtering. We therefore use a modified Gaussian kernel, similar to Gaussian bilateral filtering [TM98], but with the similarity function defined on the height map rather than on the to-be-filtered image. A σ_r of half the average layer thickness for the similarity function effectively reduces the cross talk, while slanted areas are not noticeably affected.

5.2. Interactive Preview

Optimal filter settings depend on the image and the quality of the surface, and can hardly be judged from looking at the filtered image alone. We provide an interactive preview, where all filter settings can be adjusted, and the resulting surface is rendered at interactive frame-rates using DirectX. The height map, luminance image and filtered images are uploaded onto the graphics card. Rendering is performed in two passes: The first pass processes and mixes the individual images into the final height map in a pixel shader, and the second pass creates the geometry in a vertex shader and renders it with Phong shading and different textures: monochrome, original image, depth map, and height iso-lines.

Additionally, the mixing stage allows several *correction layers* to be included in the depth map with adjustable contribution. We used these to import depth maps of important features that were modeled in external programs. For instance, we considered the shapes of faces to be very important: first, because these are important points of reference for blind observers. Second, since the human visual sense is very sensitive to images of faces, it was important to us, to present a pleasant model also for sighted visitors. We modeled the face regions with a similar morphable face database as presented by Blanz and Vetter [BV99] as a semi-transparent overlay on blowup-regions of the painting. A depth map is rendered and stitched into the height field using appropriate seams.

5.3. Production

Finally, the height map is carved out of a solid block using a CNC milling machine. Conventional CAM software was not able to handle the vast amount of data, and specialized modules for height map processing were limited to a height resolution of 8 bits. We implemented our own CAM module that uses floating point height values and directly outputs G-Code [Smi07] for milling machines. Basically, a grayscale dilation [GW06] is performed on the height map, with a height map of the milling tool as kernel, that yields the required tool offset at every pixel. Caused by the tool diameter, fine low-relief details will be rounded, but our filtering method ensures, that the important features are still preserved. For larger quantities, we create a negative cast, in which several copies can be molded. The used material is very robust, pleasant to touch, dirt-repellent and can be disinfected.

5.4. Results

Up to now, four *textured reliefs* have been processed and are currently presented and evaluated in guided tours at the museum. About 50 people (one visually impaired, others blind, 30% from birth) have so far examined the reliefs in special workshops. Almost all people report a clearer perception of the composition, and that our tactile paintings open them "a new dimension" of perceiving images, especially, to get a three-dimensional impression. In addition to composition and depth relations, that are also perceptible in the *layered depth diagrams*, the texture and other fine details could be perceived. Verbal input is in any case mandatory for correct interpretations. People, who are blind for a longer time, tend to understand the tactile paintings faster and easier. Some less experienced users preferred the *layered depth diagrams* because of their higher abstraction and simpler shapes. Both techniques seem to complement each other well. A more comprehensive user study is planned in the near future.

6. Conclusion and Future Work

We developed a complete workflow for creating different kinds of tactile representations of paintings. By adapting algorithms from the field of visual computing and creating a novel annotation-based interface, the conversion process and the production via rapid prototyping machines is completely computer-assisted. The process is not fully automated, but the necessary manual input is intuitive and easily accomplished. Currently the conversion takes several work days, mainly because the current prototype system consists of several unoptimized programs that have to be alternated several times. In the future, we plan to further optimize the process, and to develop more support for the user; for example, semi-automatic line tracing, automatic depth reasoning, or shape from shading based interpolation schemes. It would be interesting to test the procedure on images of other art styles, and to find ways to include color and other information. Finally, we think about high-resolution tactile output devices and fully automatic conversion techniques for home use.

References

[AG00] AXEL E. S., GERSON P. L. (Eds.): *Art History Through Touch and Sound: A Multisensory Guide for the Blind and Visually Impaired.* OpticalTouch Systems Publishers, 2000. 2

(a) Fouquet's Court Jester Gonella, around 1445.

(b) Dürer's Virgin Mary with Child, dated 1512.

Figure 3: More examples of *Textured reliefs* of paintings.

[AL03] AXEL E. S., LEVENT N. S. (Eds.): *Art Beyond Sight: A Resource Guide to Art, Creativity, and Visual Impairment*. AFB Press, 2003. 2

[Bar99] BARTEN P. G. J.: *Contrast Sensitivity of the Human Eye and its Effects on Image Quality*. SPIE Press, 1999. 6

[Bor86] BORGEFORS G.: Distance transformations in digital images. *Comp. Vision Graph. Image Proc. 34* (1986), 344–371. 4

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99* (New York, NY, USA, 1999), ACM, pp. 187–194. 7

[CBRA09] COSTES E., BASSEREAU J.-F., RODI O., AOUSSAT A.: Graphic design for blind users: an industrial case study. In *IASDR* (Coex, Seoul, Korea, 2009). 2

[CSRL01] CORMEN T. H., STEIN C., RIVEST R. L., LEISERSON C. E.: *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001. 4

[dFI91] DE FRANCE F., ICOM (Eds.): *Museums without Barriers: A new deal for disabled people*. Routeledge, London, 1991. 2

[FH04] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision 59* (September 2004), 167–181. 2

[GW06] GONZALEZ R. C., WOODS R. E.: *Digital Image Processing (3rd Edition)*. Prentice Hall, 2006. 7

[HB00] HERNANDEZ S. E., BARNER K. E.: Tactile imaging using watershed-based image segmentation. In *Assets '00* (New York, NY, USA, 2000), ACM, pp. 26–33. 2

[HEH05] HOIEM D., EFROS A. A., HEBERT M.: Automatic photo pop-up. *ACM Trans. Graph. 24* (July 2005), 577–584. 2, 4

[JRW*07] JAYANT C., RENZELMANN M., WEN D., KRISNANDI S., LADNER R., COMDEN D.: Automated tactile graphics translation: in the field. In *Assets '07* (New York, NY, USA, 2007), ACM, pp. 75–82. 2

[KW59] KELLEY JR J. E., WALKER M. R.: Critical-path planning and scheduling. In *IRE-AIEE-ACM '59 (Eastern)* (New York, NY, USA, 1959), ACM, pp. 160–173. 4

[LIR*05] LADNER R. E., IVORY M. Y., RAO R., BURGSTAHLER S., COMDEN D., HAHN S., RENZELMANN M., KRISNANDI S., RAMASAMY M., SLABOSKY B., MARTIN A., LACENSKI A., OLSEN S., GROCE D.: Automating tactile graphics translation. In *Assets '05* (New York, NY, USA, 2005), ACM, pp. 150–157. 2

[MBnB07] MEYER A., BRICEÑO H. M., BOUAKAZ S.: User-guided shape from shading to reconstruct fine details from a single photograph. In *ACCV '07* (Berlin, Heidelberg, 2007), Springer-Verlag, pp. 738–747. 2, 6

[OYS10] OOUCHI S., YAMAZAWA K., SECCHI L.: Reproduction of tactile paintings for visual impairments utilized three-dimensional modeling system and the effect of difference in the painting size on tactile perception. In *Computers Helping People with Special Needs*, Miesenberger K., Klaus J., Zagler W., Karshmer A., (Eds.), vol. 6180 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, pp. 527–533. 2

[RT09] RUSSELL B. C., TORRALBA A.: Building a database of 3d scenes from user annotations. In *CVPR 2009* (2009), IEEE, pp. 2711–2718. 2

[Smi07] SMID P.: *CNC Programming Handbook, Third Edition*, 3rd ed. Industrial Press, Inc., New York, NY, USA, 2007. 7

[SS01] SHAPIRO L. G., STOCKMAN G. C.: *Computer Vision*. Prentice Hall, January 2001. 3

[SSN09] SAXENA A., SUN M., NG A. Y.: Make3d: Learning 3d scene structure from a single still image. *IEEE T. Pattern. Anal. 31* (2009), 824–840. 2, 4

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *ICCV '98* (Washington, DC, USA, 1998), IEEE Computer Society, pp. 839–. 7

[VDH09] VENTURA J., DIVERDI S., HÖLLERER T.: A sketch-based interface for photo pop-up. In *SBIM '09* (New York, NY, USA, 2009), ACM, pp. 21–28. 2, 4

[WB97] WAY T., BARNER K.: Automatic visual to tactile translation. i. human factors, access methods and image manipulation. *IEEE Trans. Rehabil. Eng. 5*, 1 (Mar. 1997), 81 –94. 2

[WSTS08] WU T.-P., SUN J., TANG C.-K., SHUM H.-Y.: Interactive normal reconstruction from a single image. *ACM Trans. Graph. 27* (December 2008), 119:1–119:9. 2, 6

[WTBS07] WU T.-P., TANG C.-K., BROWN M. S., SHUM H.-Y.: Shapepalettes: interactive normal transfer via sketching. *ACM Trans. Graph. 26* (July 2007). 2, 6

[WXL08] WANG Z., XU X., LI B.: Bayesian tactile face. In *CVPR 2008* (June 2008), pp. 1 –8. 2