The 8th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2007)
D. Arnold, F. Niccolucci, A. Chalmers (Editors)

A Pipeline for the Digitization and the Realistic Rendering of Paintings

Frédéric Larue† Lucas Ammann† Jean-Michel Dischler†

LSIIT UMR 7005 CNRS-ULP - Université Louis Pasteur Strasbourg I, France

Abstract

Digitization and visualization are both of great importance for Cultural Heritage, for instance for the design of virtual galleries. Despite a lot of research, enabling a real-time walkthrough around complex digital copies still remains difficult and challenging in the general case due to the complexity of the measurement and to the amount of data that has to be dealt with. In this paper, we introduce a new dedicated pipeline for both digitization and realistic rendering of art paintings. We exploit the fact that geometrical variations over the canvas are generally small, yet not negligible from a visual point of view. Unlike most existing painting digitization systems, we thus propose to acquire both geometry and texture. Then, we render both as a whole by using, for the texture, an analytical model which is fitted from real measurements, and by using for the geometry a hybrid approach combining two relief rendering techniques according to the scale. This allows us to derive an efficient adaptive scheme guaranteeing fast rendering rates for all viewpoints. With our pipeline the painting's relief is well preserved, thus the rendering is of high quality, and in addition the final data representing the digital copies remain compact.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Digitizing and scanning, Display algorithms, Three-Dimensional Graphics and Realism

1. Introduction

In the research field of computer graphics, many works attempt, since many years, to improve the quality of synthesized images with more and more complex objects, like those provided by digitization devices. The rendering quality is not the only goal of this research field: the speed of the rendering algorithms is also an important topic in computer graphics. Many researchers try to find a way to combine high quality of computed images with high speed of the rendering process.

In the field of cultural heritage, digital acquisition is now a common way to protect but also to spread art pieces to a general public. For example, the Michelangelo project [LPC*00] was created to digitize some major Michelangelo's sculptures. The result of this project is a database of very detailed meshes, compound of hundreds of millions of triangles. Indeed, data produced by 3D scanners become, nowadays, more accurate but also more complex to manipulate. Due to the broad variety of objects to be digitized, an interesting question is related to the use of dedicated techniques for the acquisition and the rendering of specific kinds of art pieces. In this paper, we have focused our efforts on the digitization and the real-time realistic rendering of art paintings. We want to offer a method which is technically simple and which allows a free real-time walkthrough into virtual art painting galleries or museums, yet preserving a high degree of visual accuracy.

The state of the art concerning the digitization of art paintings is particularly sparse. In general, the geometry is approximated by a simple plane, and the chromatic information is simply captured by a set of digital pictures. But, in reality, there exists a lot of relief on the painting surface that comes from the canvas texture or the paint layers. These reliefs may be visually significant and have to be considered in order to achieve a realistic rendering of the art piece. Indeed, relief plays a key role in the shading of the painting surface and adds some important geometric features, some-

† e-mail: {larue, ammann, dischler}@lsiit.u-strasbg.fr

Figure 1: Left*: pictures of the Harbour painting model.* Middle*: a rendering with diffuse texture mapping on a simple quad.* Right*: a rendering with our hybrid approach and a reconstructed bidirectionnal texture.*

times wished by the painter himself. Paintings made by using a palette knife are good examples. Therefore, the geometric information of the painting must be considered not only by the acquisition process but also during the rendering step.

We propose a new method to accurately acquire the geometric and photometric information of art paintings, and an algorithm for the visualization of the acquired digital copies. We use an analytical model to represent the color texture in a realistic manner and a hybrid approach to display the geometric features of the painting, speeding up the rendering with an adaptive technique. We will show that real-time rendering is achieved on modern graphics hardware and that data representing the final copy are particularly compact. The major contributions of this paper can then be summarized as follows:

- A complete pipeline, from acquisition of both texture and relief of a painting up to its final rendering;
- Some pre-processings, specifically tailored for data arising from art paintings;
- An efficient algorithm to achieve, with a small memory usage, a realistic rendering at real time thanks to a new dedicated adaptive rendering scheme.

After a brief discussion about some related works, we present an overview of our pipeline (in section 3), including the digitization process and the rendering algorithm, respectively described in sections 4 and 5. Next, we present some results in section 6 and conclude with some possible perspectives for this work in section 7.

2. Related Works

In the field of art painting acquisition, we can quote Tominaga *et al.* [TTK04], who developed a technique to acquire the color of a painting using a multi-band camera. This work considers the relief of the surface for the acquisition part only : roughness of the surface is used to determine the spectral reflectance of the painting but not during the rendering part. Another work, Grattoni *et al.* [GS03], presents a new method to digitize painted surfaces. This method uses an active vision system to acquire both geometry and color of painting. However, this paper only considers the geometry of the painting support, and not of the painting surface itself. This method is dedicated to digitization of painted surfaces like frescoes for conservation and restoration purposes. In the field of restoration, authors of [GAL03] explain how they use a 3D range camera to scan a Leonardo Da Vinci's wood painting. This project is focused on geometry acquisition to provide diagnostics on the state of the painting's wood support. These methods consider the relief for acquisition but not for rendering, unlike our approach.

To obtain a realistic rendering of complex surfaces with a small amplitude relief, there exist a lot of different ways. A first one is, obviously, the use of a complex and detailed mesh, but this solution is generally untractable for real time rendering. A second one is the visual simulation of these reliefs, which provides very good rendering performances compared to the use of detail meshes. Bump mapping [Bli78] was the first of these methods. A particular texture mapping is used to perturb the normal vectors at the surface, and the induced changes in lighting computations simulate the relief. This algorithm provides good visual results, high rendering performances, and some works have already proposed to capture bump maps from real objects [RTG97]. Unfortunately, several problems occur at grazing angles, like silhouette or parallax errors, especially when the relief to be simulated becomes too important. Techniques like displacement mapping [Coo84] then propose to avoid these prob-

lems by directly modifying the underlying geometric object. More recently, relief mapping [OBM00, POC05, BD06] and parallax mapping [Tat06] were introduced to visually add true geometric details over the surface by a complex texture mapping technique, thus avoiding needs of explicit meshes. These kinds of algorithms use modern graphics hardware capabilities to perform advanced processings and provide good visual results as well as real-time rendering.

In addition to geometry, the major feature of an art painting is obviously its color, which has to be accurately acquired too, in order to be further rendered in a realistic manner. Color changes over the surface are closely tied to spatial variations of the surface reflectance resulting from the different paints used. A lot of works are focused on measurement and rendering of object surface reflectance. The Bidirectional Texture Function (BTF) [DNGK97] was introduced for this purpose. The BTF represents the spatial variations of a material, including BRDF changes, over a surface. Since the BTF is a 6-dimensional function, not particularly tractable for real time rendering due to its high memory requirement, some methods have further attempted to acquire and to compress this complex information [LFTG97, MWL*99, NDM05]. For more information on BTF, a survey was published by Müller *et al.* [MMS*05]. Like McAllister *et al.* [MLH02], our method uses a Lafortune model [LFTG97] to represent the reflectance at each point of the painting surface. As pointed out in [MMK03], the measurement of BTF for highly depth-varying surfaces introduces a lot of errors. In fact, BTF is not able to handle correctly high varying reliefs. It is the reason why our method is focused on both acquisition and rendering of the painting relief instead of using straightforwardly a BTF only.

Most of the previous rendering methods are, under particular conditions, not adapted because of their computational cost or their unsuitable rendering quality. Hybrid techniques can then provide an alternative solution. Such a mechanism consists in using a different rendering algorithm according to various parameters, such as the relative camera position or the projected object size on screen. For example, Becker *et al.* [BM93] alternatively use three different techniques, depending on the viewer position and on its distance to the scene. The main problem of hybrid techniques resides in the switch between the different rendering algorithms, which is prone to visual artifacts. As pointed by Heidrich *et al.* [HDKS00], the most prominent visual inconsistencies arise from illumination differences between the different algorithms used. Usually, smooth transitions to prevent these 'popping' effects are managed by a simple alpha blending, sometimes improved, as in [GW07]. But, all these problems aside, hybrid methods remain powerful tools for the rendering speedup. However, there is no general solution: each technique is adapted to a specific situation. This is why we have chosen to introduce our own technique, designed for the specific and efficient rendering of painting surfaces.

3. Overview

The method proposed in this paper is based on the observation that many art paintings have some geometric features that must be preserved. In fact, some reliefs over the canvas are visually significant and must be rendered by a method that preserves it. But an important part of the relief remains visually negligible and can be approximated by a simple visualization technique. We then propose to decouple the visually significant part of the relief from the remaining.

Firstly, the whole painting surface is approximated by a heightfield, that is to say a map defining at each point an elevation with respect to a reference plane. This representation is particularly suited to art paintings due to the highly planar shape of such objects. Relief is then classified by thresholding the heightfield. Points below the threshold are rendered by using a simple bump mapping technique. The remaining points, associated to the significant relief part, are represented as a set of 3D boxes located above the canvas plane and rendered by using a more complex relief mapping technique accounting for true underlying geometry. The mix of both rendering techniques is managed by a dedicated hybrid and adaptive scheme, drastically increasing rendering speed.

Color rendering of the art painting is finally achieved by using a spatially varying description of its material, represented by a texture of Lafortune lobes [LFTG97], fitted from real measurements and evaluated on the fly by the graphics hardware, using shader programming.

4. Acquisition and Preprocessing

In this section, we explicit our choices concerning acquisition of both geometry and texture. We also present the preprocessings required by our hybrid algorithm for the rendering of digitized paintings, including relief classification.

4.1. 3D Acquisition

The geometric information is acquired by using a structured light range scanner. As chromatic details on paintings may present high contrast, we capture multiple range images, with different exposures, to ensure that both lightened and darkened regions are correctly acquired. Due to the particular shape of the considered kind of objects, only one view is acquired for the 3D information, nearly perpendicular to the painting canvas. The 2D parameterization of the range image and the quasi orthogonal viewing angle provide us a geometric representation which is not really far from the desired heightfield representation. Let us see in the following subsection how to properly extract it from the data.

4.2. Automatic extraction of painting canvas

Depending on the application, one may want to discard information not directly related to the painted surface itself. It

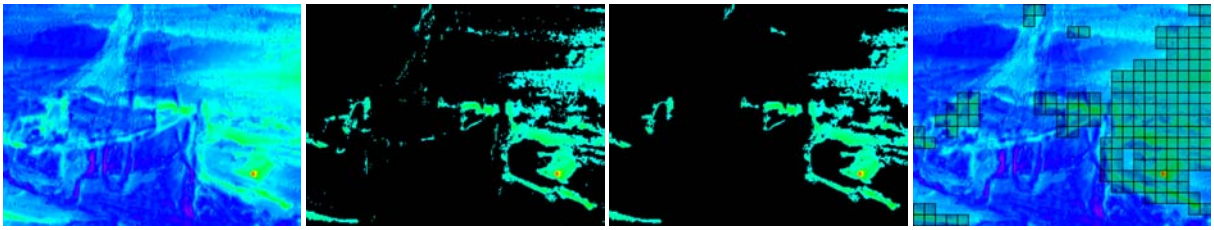

Figure 2: *Relief classification. From left to right:* ***1)*** *heightmap resulting from PCA alignment of the acquired range image, highlighted by a color gradient,* ***2)*** *hard thresholding of height coordinates,* ***3)*** *cleaning process to avoid noise influence,* ***4)*** *boxes created around the remaining connected components and further used for the rendering of the relief significant part.*

is then interesting to propose a processing in order to extract the canvas by discarding the painting frame and all other irrelevant geometrical features.

As said before, an important characteristic of a digitized painting is the highly planar shape of the resulting 3D data. Indeed, surface relief can be seen as a height variation with respect to a representative plane. To recover the heightmap corresponding to the digitized painting, we perform a principle component analysis (PCA) on the range data to extract a new local frame $(\vec{X},\vec{Y},\vec{Z})$, where $\vec{X}$ and $\vec{Y}$ are the axis of principle dispersion, tangent to the canvas plane, and $\vec{Z}$ is the orthogonal axis. The location of each range point is then recomputed with respect to this new frame, and the coordinate along $\vec{Z}$ is referred to as *height coordinate*.

By considering this new local frame, transition between the canvas and the frame necessarily presents a rough height discontinuity compared to the small geometric features of the painting itself. Thus, we compute a gradient image by applying the Sobel operator on the height coordinate to highlight pixels of high discontinuity. As we want to keep only the inner part of the painting (that is to say, the canvas), we use a seed fill algorithm, which stops to propagate when pixels of too high gradient are encountered. However, seed fill starting point must also lay on the canvas. We simply choose the center point of the range image, as it is obvious that correct measurements of painting models must be in shot and well centered with respect to the acquisition device. The region resulting from the filling algorithm is then used as a mask for the range image, discarding all points that lay outside, and only keeping those corresponding to the canvas. A final erosion of a few pixels is performed to avoid possible residual artifacts.

Now that only the relevant information remains, we perform a new PCA to recompute a local frame $(\vec{X},\vec{Y},\vec{Z})$ which is best aligned with the only canvas plane, thus getting more accurate height coordinates.

4.3. Relief classification

Our goal is now to extract the painting relief which is significant enough, in a visual sense. We define a threshold δ_0 below which height is considered as negligible. Actually, this threshold is provided manually and is, in our tests, between 20% and 40% of the maximum height of the canvas. All pixels whose height is upper than δ_0 are marked as *valid*, and the others are set to *invalid*, so as to create a binary mask image $\mathcal{M}_v$. $\mathcal{M}_v$ is then processed to discard connected components of valid pixels whose size are too small to avoid taking into account the possible influence of digitization noise and to keep only the really significant relief.

Next, the pixel grid of the range image is subdivided into a set of square cells, of $n \times n$ pixels each. We further show, in section 5, that the parameter n is important for our hybrid rendering as it enables to control the granularity of our adaptive scheme. Cells that do not contain any valid pixel are discarded. For the remaining ones, a box is created whose size is defined by the pixel size of the cell and by the maximum height of the canvas. The segmented relief is then completely enclosed into the final set of boxes, as illustrated in figure 2, which is further used as base for a separate rendering of the significant relief part. It is obvious but important to note that the smaller the resolution n, the more boxes are created. The choice of this value is discussed in the results section. Moreover, we also compute the height h_{max} and the texture coordinates (u_{max}, v_{max}), in the 2D space of the range image, of the highest point of each box. These values are used in our rendering technique, further described in section 5.3.

4.4. Bidirectional texture recovery

Concerning the photometric information, the appearance of the painting is captured from several pictures, taken under different viewing and lighting conditions. Camera localization is performed by using the structured light parametrization described in [LD06], and light source localization is achieved by using a fixed mechanical structure, enabling to cover the upper hemisphere over the object.

Once pictures have been registered with respect to geometry, radiance samples are extracted for each valid point of the range image by back-projection onto the image space of each picture. These samples are compound of the local viewing and lighting directions, and of the color of the hit pixel. Among all samples, some correspond to cases of self occlusions, when light transport is stopped by the underlying

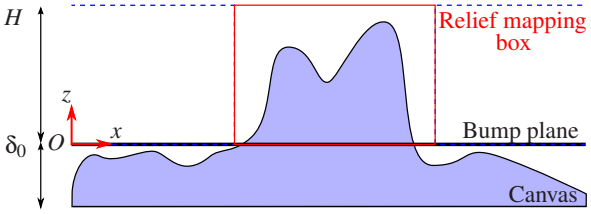

Figure 3: *Relative positions of bump plane and relief boxes. The red box encloses a relief part considered as significant.*

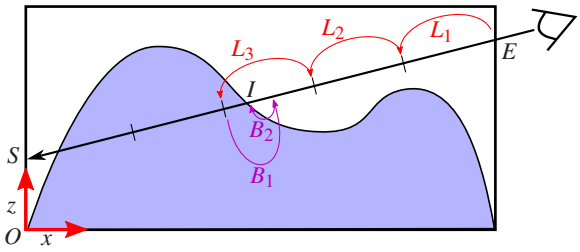

Figure 4: *The GPU-based ray / heightfield intersection search performed by the relief mapping rendering algorithm.*

geometry. Analytical models for texture representation are generally sensitive, in terms of accuracy, to the high discontinuities represented by such shadowing effects. Moreover, since relief information is available in our case, shadows due to self occlusions may eventually be simulated during the rendering. We thus choose to discard samples that do not correspond to direct illumination. Self occlusions are simply detected by ray casting on the acquired geometry.

A Lafortune BRDF model is finally fitted at each 3D point of the range image. We use the approximation proposed in [MLH02], with one specular lobe, as in equation 1:

$$f_r(\vec{v},\vec{l}) = \rho_d + \rho_s \left(C_x v_x l_x + C_y v_y l_y + C_z v_z l_z\right)^k \qquad (1)$$

where ρ_d and ρ_s are RGB vectors respectively describing diffuse and specular contributions to the outgoing lighting energy, and $\vec{v}$ and $\vec{l}$ are the local viewing and incident light directions. We have chosen this model for its simplicity, making its evaluation easier on graphics hardware, and for its compactness, as only three 2D textures are required: two for ρ_d and ρ_s, and a third one for lobe shape parameters (C_x, C_y, C_z, k). Obviously, other texture representation models can be used, but we recall that one of our goal is the compactness of the final digitial copies.

5. Rendering

After the binary relief classification, we obtain two different kinds of reliefs, the *negligible* and the *significant*, each one being rendered by using a different technique: bump mapping and relief mapping, respectively. These two rendering methods are mixed together using a mechanism which automatically chooses the better rendering method depending on viewing conditions, thus enhancing the rendering speed. The significant relief part is associated to the set of boxes computed in the aforementioned pre-processing step. In addition to these boxes, we also use a heightfield texture to represent the height and the normal vector at each point of the canvas. Height coordinates correspond to the values previously computed by PCA alignment, and normals are computed by considering the triangles that can be created with the 8-neighbourhood on the range image. Only the two components n_x and n_y are stored, the third one being recomputed on the fly. This heightfield texture is used for rendering of both negligible and significant reliefs.

5.1. Negligible relief rendering

For the negligible relief part, painting surface is approximated by a plane, represented as a simple quad. This plane is rendered using a classical bump mapping algorithm. Since the significant relief part begins only above the fixed threshold δ_0, a gap exists between the minimum height of the canvas and the starting height of boxes used to represent the significant relief. To solve this problem, we shift the bump mapping plane to this threshold. With this correction, the significant relief rendering starts exactly at the same level as the bump plane. In fact, as shown on figure 3, real canvas is under the bump plane but the highest point of the significant relief's part is the same as the real painting highest point. Therefore, with this bump plane position, this rendering method respects the real relief of the painting.

5.2. Significant relief rendering

The rendering algorithm used for significant relief areas is close to the relief mapping algorithm described by Policarpo *et al.* [POC05]. The parts rendered with this algorithm are superposed over the bump mapping plane. This allows us to correct parallax errors introduced by bump mapping for the most significant part of the painting.

Relief mapping reminder The relief mapping algorithm is an extension of classical texture mapping. It uses GPU to modify the geometry of the rendered object. For all rendered pixels of a simple polygon, a ray is cast and the intersection with the real object surface is determined. This is achieved by first using a linear search along the ray direction, followed by a binary refinement, respectively represented by the steps L_i and B_j on figure 4. The real object surface is represented by a heightfield texture, whose pixel intensities are interpreted as height variations instead of colors. When the intersection point between ray and real surface is obtained, the shading of this pixel can be computed.

Relief mapping over boxes Traditionally, relief mapping algorithm uses a single polygon as support of its rendering. Our method uses boxes defined in 3D space. Each of these boxes is defined by its position (u_b, v_b) and its size (w_b, h_b) in the image space of heightfield texture. Each box is then

Figure 5: *Behaviour of our view dependent adaptive scheme. For close-ups (left) or at grazing angles (right), more boxes are automatically selected to be rendered with relief mapping so as to prevent parallax error where relief becomes significant.*

associated to a portion of the heightfield texture, which is consequently shared by all boxes. Minimum and maximum height values are the same for all boxes, and are respectively corresponding to the bump plane level and to the maximum height coordinate H of the whole canvas.

To efficiently render relief mapping while using boxes as basis, we associate 3D texture coordinates to each vertex to define a local frame at each box in which the ray casting algorithm is performed, as shown by axis O_x and O_z on figure 4. During rendering, these texture coordinates immediately provide the entry point coordinates of the ray into the box, whatever the considered box side is. Relief mapping algorithm then proceeds as follows: for all pixels hit by the currently rendered box, the entry point E and exit point S of the viewing ray are found by using 3D texture coordinates. These two points give the traversal vector along which the intersection lookup with the heightfield has to be done. Linear and binary searches are performed similarly to the standard relief mapping algorithm.

5.3. Adaptive rendering mechanism

By looking at a painting along a direction close to its surface normal, we can observe that there are no visual differences between relief and bump mapping. The same observation can be made when the camera is far from the painting surface, that is to say when the height variation becomes negligible with respect to its projected size on screen. Indeed, the distance between the real surface point and the corresponding point on the bump plane is too small to be noticeable.

For this reason, we have developed an algorithm to automatically determine whether boxes must be displayed or not. Moreover, alpha blending is performed to prevent eventual 'popping' artifacts. To determine the state of a box, a coefficient α_{box} is computed by using its highest point I, whose world coordinates are easily determined from values h_{max} and (u_{max}, v_{max}) previously computed (see section 4.3). By considering the projection B of I along the viewing direction $\vec{v}$, we observe that the ratio $r = \|\vec{IB}\| \ / \ \|\vec{CI}\|$, where C is the camera position, directly depends on the viewing angle, but also on the distance between the painting surface and the viewpoint, as illustrated on figure 6. Ratio r tends

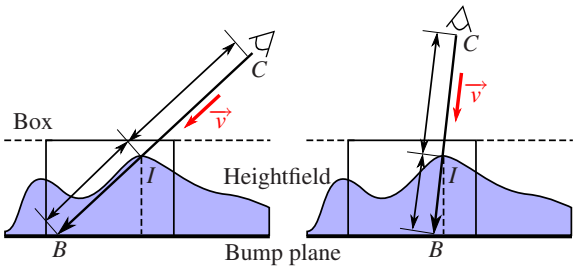

Figure 6: *Switching α coefficient calculus based on the highest point I of each box. Ratio $\|\vec{IB}\|/\|\vec{CI}\|$ depends on the view angle and on the distance to the surface.*

to be null for front or distant viewpoints, and increases at grazing angles or for near viewpoints. It is thus relevant for our adaptive rendering purpose. Computation of the per box coefficient α_{box} is given by equation 2:

$$\alpha_{box} = \begin{cases} 0 & \text{if } Kr < \varepsilon_{min} \\ 1 & \text{if } Kr > \varepsilon_{max} \\ \frac{Kr - \varepsilon_{min}}{\varepsilon_{max} - \varepsilon_{min}} & \text{otherwise} \end{cases} \quad (2)$$

Where ε_{min} and ε_{max} are two clamping thresholds, and K is a user defined scale factor to control the sensitivity of the switching algorithm. Depending on the value of α_{box}, there are three cases to consider. If α_{box} is null, the box is not drawn. If $\alpha_{box} = 1$, the box is drawn without blending. If $\alpha_{box} \in]0,1[$, the box is displayed with alpha blending.

Generally, illumination inconsistencies may appear when different rendering methods are mixed together. But here, the same normal and color information are used for both bump and relief mappings, and the illumination model is the same for the two rendering methods. Hence, no illumination difference is visible at the transition between bump and relief mapping areas, as shown on figure 7.

5.4. Shading

As mentioned in section 4.4, we use a spatially varying Lafortune model to compute illumination. Equation 1 is directly evaluated on GPU. Color information is available via

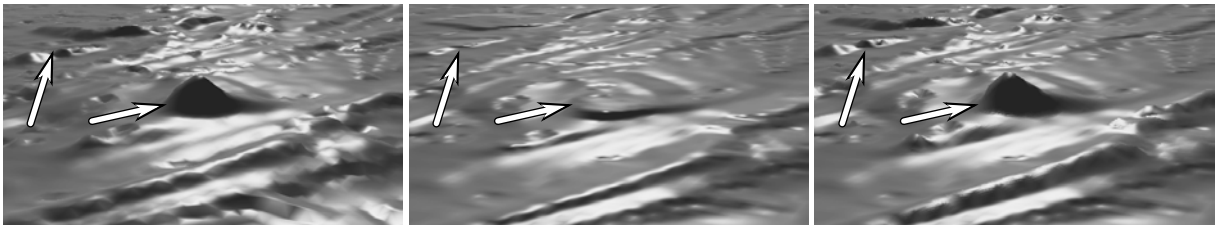

Figure 7: *Comparison between mesh rendering (left), bump mapping (middle) and our hybrid rendering (right). The parallax error introduced by bump mapping at grazing angles is corrected by using relief mapping where the relief becomes significant.*

two RGB and one RGBA textures. Lighting and viewing directions are both corrected with the height coordinate, even for bumpmapping, so as to account for the real surface position, and the normal vector is recomputed on-the-fly by the graphics hardware from the n_x and n_y coordinates previously estimated and stored into the reliefmap (see section 5).

6. Results and Discussion

We tested our method on the Harbour painting, illustrated on figure 1 by an original picture and a rendering of our digital copy. The main advantage of our method is that we do not use any mesh to represent the geometry, therefore providing a low cost memory consumption. In fact, the range image captured by the 3D scanner is directly processed and used as heightfield texture, thus working on the initial geometry sampling. Only one RGBA and three RGB 16bits floating point textures and a list of boxes are used to represent the digital copy, including bidirectionnal texture information. The total size after processing is about 11Mo. The corresponding mesh is composed of 815K triangles.

Our hybrid method built upon bump and relief mapping algorithms provides real-time rendering performances, as illustrated on table 1. Frame rates are measured on an AMD Athlon X2 4200+ and a NVIDIA GeForce 7900GTX. One can observe that the speedup provided by our adaptive algorithm is significant, outperforming the standard relief mapping in all cases. However, our approach is less efficient than a mesh based rendering for viewpoints really near to the surface. Indeed, the speed rendering for the adaptive mechanism highly depends on the viewpoint. At grazing angles or for close-ups, the relief is more perceptible and our adaptive rendering uses more relief boxes for such configurations, thus decreasing the rendering performances. It also depends on the number of boxes, which is directly linked to their size. If a small size is chosen, boxes are well fitting the relief, but the box visibility computation becomes more costly. On the contrary, a bigger box size leads to render more pixels with the heightfield mapping algorithm, which is more time consuming than bump mapping rendering. Performances for various box sizes are shown on table 2.

Relief mapping is particularly adapted to the rendering of art paintings, because of their planar nature. Indeed, traditional paintings are well represented by a simple heightfield. Another advantage of using relief mapping is that parallax errors introduced by bump mapping are fixed, as shown in figure 7. However, it is important to note that rendering of colored heightfields has some limitations. Indeed, if an important height jump occurs between two adjacent pixels of the height map, color information along the 'cliff' is simply interpolated between its top and bottom points. Thus, the more important the height discontinuities, the more stretched the color texture is. This limitation proves that this method is not easy to extend to the rendering of bas-relieves. Another problem with relief mapping is, as discussed in [BD06], the presence of some artifacts due to the discrete sampling of the ray during the linear and binary intersection searches. In our case, the relief is small compared to what is traditionally rendered using this algorithm, so artifacts are very limited. The number of steps required for our intersection lookup is small, thus increasing rendering performances.

Dist.	View Angle	Mesh	RM	Hybrid	Hybrid + Adapt.
near	grazing	71	36	33-48	36-57
	front	85	32	39-48	56-72
in shot	grazing	60	100	61	330
	front	52	74	60	350
far	grazing	96	280-450	82	650
	front	96	155-230	80	1250

Table 1: *Rendering speed (in Hz) for the Harbour painting using different techniques and for different view angles.*

Box Size	♯ Boxes	Front	Intermediate	Grazing
5×5	2073	305	142	18
10×10	714	300	185	38
20×20	253	330	250	75
50×50	81	240	175	145
100×100	35	135	130	125

Table 2: *Comparison of rendering speeds (in Hz) for different box sizes and different view angles.*

7. Conclusions and Perspectives

In this paper, we introduced a new pipeline to digitize and to render art paintings. This method allows one to acquire both

geometry and color of the painting surface. We use a hybrid method involving two different rendering techniques to account for the acquired relief, depending on its importance. An adaptive mechanism is used to combine both rendering algorithms. Actually, we provide real-time rendering performances and a low memory cost, since the painting is only represented by a heightfield and three textures for the bidirectionnal material representation. Moreover, the real relief added by the rendering process improves the realism.

For the moment, we only worked on small scale paintings. An extention of this pipeline to allow the acquisition of bigger paintings is an interesting task. The heightfield structure used imposes some restrictions to the representable object class. Using another data structure would allow the processing of some modern art pieces, like collage, not efficiently handled by our method due to their high varying relief. Finally, as we have notice that mesh rendering is more efficient for close-ups, we are interesting in adding a third rendering level, using the acquired mesh itself at the finest scale.

References

[BD06] BABOUD L., DÉCORET X.: Rendering geometry with relief textures. In *Proc. of the 2006 conference on Graphics interface* (2006), pp. 195–201.

[Bli78] BLINN J. F.: Simulation of wrinkled surfaces. In *Proc. of SIGGRAPH '78* (1978), pp. 286–292.

[BM93] BECKER B. G., MAX N. L.: Smooth transitions between bump rendering algorithms. In *Proc. of SIGGRAPH '93* (1993), pp. 183–190.

[Coo84] COOK R. L.: Shade trees. In *Proc. of SIGGRAPH '84* (1984), pp. 223–231.

[DNGK97] DANA K. J., NAYAR S. K., GINNEKEN B. V., KOENDERINK J. J.: Reflectance and texture of real-world surfaces authors. In *Proc. of the Conference on Computer Vision and Pattern Recognition* (1997), p. 151.

[GAL03] GUIDI G., ATZENI C., LAZZARI S.: 3d optical scanning diagnostics for Leonardo Da Vinci's "Adorazione dei Magi" conservation. In *3DIM* (2003), pp. 110–115.

[GS03] GRATTONI P., SPERTINO M.: A mosaicing approach for the acquisition and representation of 3d painted surfaces for conservation and restoration purposes. *Mach. Vision Appl. 15*, 1 (2003), 1–10.

[GW07] GIEGL M., WIMMER M.: Unpopping: Solving the image-space blend problem for smooth discrete lod transitions. *Computer Graphics Forum 26* (2007), 46–49.

[HDKS00] HEIDRICH W., DAUBERT K., KAUTZ J., SEIDEL H.-P.: Illuminating micro geometry based on precomputed visibility. In *Proc. of SIGGRAPH '00* (2000), pp. 455–464.

[LD06] LARUE F., DISCHLER J.-M.: Automatic registration and calibration for efficient surface light field acquisition. In *Proc. of VAST06* (2006), pp. 171–178.

[LFTG97] LAFORTUNE E. P. F., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. In *Proc. of SIGGRAPH '97* (1997), pp. 117–126.

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital Michelangelo project: 3d scanning of large statues. In *Proc. of SIGGRAPH '00* (2000), pp. 131–144.

[MLH02] MCALLISTER D. K., LASTRA A., HEIDRICH W.: Efficient rendering of spatial bi-directional reflectance distribution functions. In *Proc. of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2002), pp. 79–88.

[MMK03] MESETH J., MÜLLER G., KLEIN R.: Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium* (2003), pp. 89–96.

[MMS*05] MULLER G., MESETH J., SATTLER M., SARLETTE R., KLEIN R.: Acquisition, synthesis, and rendering of bidirectional texture functions. *Computer Graphics Forum 24*, 1 (2005), 83–109.

[MWL*99] MARSCHNER S. R., WESTIN S. H., LAFORTUNE E. P., TORRANCE K. E., GREENBERG D. P.: Image-based BRDF measurement including human skin. In *Rendering Techniques '99, Proc. of the Eurographics Workshop* (1999), pp. 131–144.

[NDM05] NGAN A., DURAND F., MATUSIK W.: Experimental analysis of BRDF models. In *Eurographics Symposium on Rendering* (2005), pp. 117–126.

[OBM00] OLIVEIRA M. M., BISHOP G., MCALLISTER D.: Relief texture mapping. In *Proc. of SIGGRAPH '00* (2000), pp. 359–368.

[POC05] POLICARPO F., OLIVEIRA M. M., COMBA J. L. D.: Real-time relief mapping on arbitrary polygonal surfaces. In *Proc. of the 2005 symposium on Interactive 3D graphics and games* (2005), pp. 155–162.

[RTG97] RUSHMEIER H. E., TAUBIN G., GUÉZIEC A.: Applying shape from lighting variation to bump map capture. In *Proc. of the Eurographics Workshop on Rendering Techniques '97* (1997), pp. 35–44.

[Tat06] TATARCHUK N.: Dynamic parallax occlusion mapping with approximate soft shadows. In *Proc. of the 2006 symposium on Interactive 3D graphics and games* (2006), pp. 63–69.

[TTK04] TOMINAGA S., TANAKA N., KOMADA T.: Imaging and rendering of oil paintings using a multi-band camera. In *Southwest Symposium on Image Analysis and Interpretation* (2004), pp. 6–10.