Automatic Texturing without Illumination Artifacts from In-Hand Scanning Data Flow

Frédéric Larue[†], Matteo Dellepiane[†],
Henning Hamer[§], and Roberto Scopigno[†]

[†] ISTI-CNR, Pisa, Italy [§] ETH, Zürich, Switzerland

Abstract. This paper shows how to improve the results of a 3D scanning system to allow to better fit the requirements of the Multi-Media and Cultural Heritage domains. A real-time in-hand scanning system is enhanced by further processing its intermediate data, with the goal of producing a digital 3D model with a high quality color texture and an improved representation of the high-frequency shape detail. The proposed solution starts from the usual output of the scanner, a 3D model and a video sequence gathered by the scanner sensor, for which the rigid motion is known at each frame. The produced color texture is deprived of the typical artifacts that generally appear while creating textures from several pictures: ghosting, shadows and specular highlights. In the case of objects made of diffuse materials, the system is also able to compute a normal map, thus improving the geometry acquired by the scanner. Results demonstrate that our texturing procedure is quite fast (a few minutes to process more than a thousand images). Moreover, the method is highly automatic, since only a few intuitive parameters must be tuned by the user, and all required computations are particularly suited to GPU programming, making the method convenient and scalable to graphics hardware.

Keywords: 3D Scanning, Texturing, Bump Map fitting, Video analysis

1 Introduction

Thanks to the fact that scanning technologies are nowadays quite affordable and reliable, the acquisition of digital 3D models from real objects is becoming more and more common for many application fields of Computer Graphics, such as video games, cinema, edutaining and/or entertaining software. Despite this fact, the whole pipeline enabling to pass from real to virtual still remains a complex process. Moreover, providing a good representation of the geometry is not sufficient for many applications, such as for example the Multimedia or the Cultural Heritage domains. In these applications, we need to pair the geometry model with an accurate representation of the surface appearance (color, small shape details, reflection characteristics); interactive visualization requires to be able to provide synthetic images as near as possible to the real appearance of the depicted object. Commercial scanning systems have mostly focused

on shape measurement, putting aside until recently the problem of recovering quality textures from real objects. This has led to a lack of efficient and automatic processing tools for color acquisition and reconstruction: in the Computer Graphics industry, even while working with meshes coming from 3D scanning, the texturing phase is still mainly performed by hand and consists in one of the most tedious tasks to be done in order to produce multimedia content.
In fact, creating quality textures for 3D models by using real pictures is a process that is prone to many problems. Among them, there are the calibration of the pictures with respect to the 3D model, and the significant artifacts that may appear in the reconstructed color, due to the presence of shadows, specular highlights, lighting inconsistencies, or small reprojection misalignment because of calibration inaccuracies.
In this paper, we present a complete system performing in an intuitive and highly automatic manner the simultaneous acquisition of shape and color, as well as the texturing of the recovered mesh. During a first step, the acquisition is made by an in-hand digitization device performing 3D scanning in real-time, which captures at the same time a color video flow of the measured object. This flow is used during the texturing step to produce over the object surface a diffuse color texture deprived of the aforementioned artifacts. Moreover, our system is also able to estimate a normal map starting from the video flow, in order to extract the finest geometric features that cannot be properly acquired by the scanning device.
The integration of the proposed method with the in-hand scanning system permits to obtain accurate 3D models of small objects within minutes. In the field of Cultural Heritage, this permits to digitize a big number of artifacts in a short time, for the purposes of archival, knowledge sharing, monitoring, restoration.
The remainder of this paper is organized as follows. The Section 2 reviews related work on software approaches or complete systems for color acquisition and texture reconstruction. The Section 3 presents our texturing method, as well as the real-time in-hand scanning system it is based on, and our approaches for hole filling and normal map extraction. Finally, Section 4, shows some results and Section 5 draws the conclusions.

2 Related work

2.1 Color acquisition and visualization on 3D models

Adding color information to an acquired 3D model is a complex task. The most flexible approach starts from a set of images acquired either in a second stage with respect to the geometry acquisition, or simultaneously but using different devices. Image-to-geometry registration, which can be solved by automatic [15, 21, 7] or semi-automatic [12] approaches, is then necessary. In the method proposed here, this registration step is not required, because the in-hand scanning system provides images which are already aligned to the 3D model.
Once alignment is performed, it's necessary to extract information about the surface material appearance and transfer it on the geometry. The most correct

way to represent the material properties of an object is to describe them through a reflection function (e.g. BRDF), which attempts to model the observed scattering behavior of a class of real surfaces. A detailed presentation of its theory and applications can be found in Dorsey [10]. Unfortunately, state-of-the-art BRDF acquisition approaches rely on complex and controlled illumination setups, making them difficult to apply in more general cases, or when fast or unconstrained acquisition is needed.
A less accurate but more robust solution is the direct use of images to transfer the color to the 3D model. In this case, the apparent color value is mapped onto the digital object's surface by applying an inverse projection. In addition to other important issues, there are numerous difficulties in selecting the correct color when multiple candidates come from different images.
To solve these problems, a first group of methods selects, for each surface part, a portion of a representative image following a specific criterion - in most cases, the orthogonality between the surface and the view direction [5, 1]. However, due to the lack of consistency from on image to the other, artifacts are visible at the junctions between surface areas receiving color from different images. These can be partially removed by working on these junctions [5, 1, 6].
Another group "blends" all image contributions by assigning a weight to each one or to each input pixel, and selecting the final surface color as the weighted average of the input data, as in Pulli et al. [17]. The weight is usually a combination of various quality metrics [3, 2, 18]. In particular, Callieri et al. [4] presented a flexible weighting system that can be extended in order to accommodate additional criteria. These methods provide better visual results and their implementation permits very complex datasets to be used, i.e. hundreds of images and very dense 3D models. Nevertheless, undesirable ghosting effects may be produced when the starting set of calibrated images is not perfectly aligned. This problem can be solved, for example, by applying a local warping using optical flow [11, 9].
Another issue, which is common to all the cited methods, is the projection of lighting artifacts on the model, i.e. shadows, highlights, and peculiar BRDF effects, since the lighting environment is usually not known in advance. In order to correct (or to avoid to project) lighting artifacts, two possible approaches include the estimation of the lighting environment [22] and the use of easily controllable lighting setups [8].

2.2 Real-time 3D Scanning

An overview of the 3D Scanning and Stereo reconstruction goes well beyond the scope of this paper. We will mainly focus on systems for real-time, in-hand acquisition of geometry and/or color. Their main issues are the availability of technology and the problem of aligning data in a very fast way.
3D acquisition can be based on stereo techniques or on active optical scanning solutions. Among the latter, the most robust approach is based on the use of fast structured-light scanners [14], where a high speed camera and a projector are used to recover the range maps in real-time. The alignment problem is usually solved with smart implementations of the ICP algorithm [20, 24], where the most

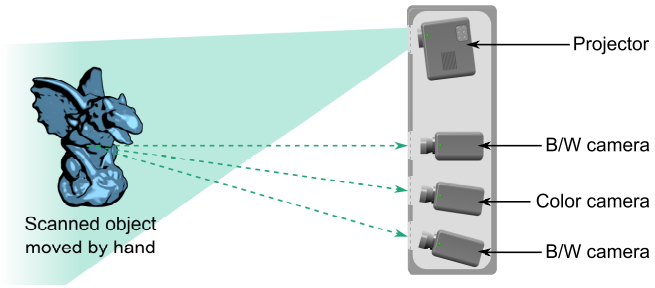

Fig. 1. The in-hand scanning device producing the data flow used as input for the methods described in this paper.

difficult aspect to solve is related to the loop closure during registration.
In the last few years, some in-hand scanning solutions have been proposed [20, 25, 23]: they essentially differ on the way projection patterns are handled, and in the implementation of ICP. None of the proposed systems take into account the color, although the one proposed by Weise et al [23] contains also a color camera (see next section for a detailed description). This is essentially due to the low resolution of the cameras, and to the difficulty of handling the peculiar illumination provided by the projector. Other systems have been proposed which take into account also the color, but they are not able to achieve real-time performances [16] or to reconstruct the geometry in an accurate way [13].

3 Texturing from in-hand scanning data flow

The scanning device which produces the data flow used by our texturing approach [23] is shown in Figure 1. Shape measurement is performed by phase-shifting, using three different sinusoidal patterns to establish correspondences (and then to perform optical triangulation) between the projector and the two black and white video cameras. The phase unwrapping, namely how the different signal periods are demodulated, is achieved by a GPU stereo matching between both cameras. The whole process produces one range map every $14ms$. Simultaneously, a color video flow is captured by the third camera. During an acquisition, the only light source in the scene is the scanner projector itself, for which the position is always perfectly known.
The scanning can be performed in two different ways. If the object color is neither red nor brown, it can be done by holding the object directly by hand. In this case, occlusions are detected by a hue analysis which produces, for each video frame, a map of skin presence probability. Otherwise, a black glove must be used. Although much less convenient for the scanning itself, it makes the occlusion detection trivial by simply ignoring dark regions in the input pictures. Each scanning session then produces a 3D mesh and a color video flow. For each video frame, the viewpoint and the position of the light are given, as well as the skin probability map in the case of a digitization performed by hand.

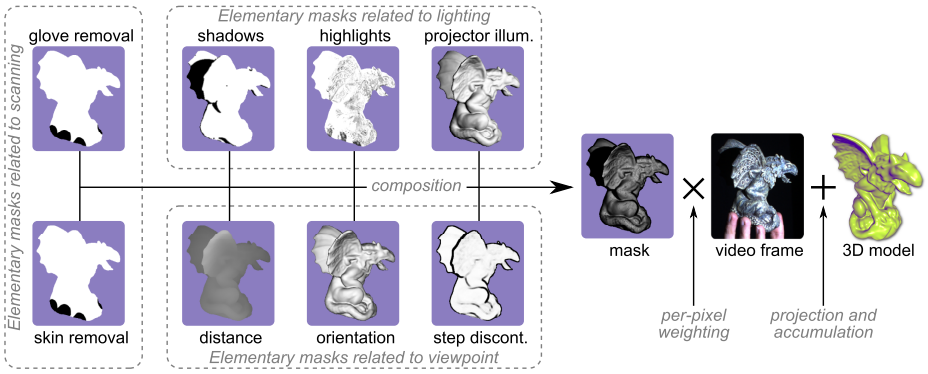

Fig. 2. The texture is computed as the weighted average of all video frames. Weights are obtained by the composition of multiple elementary masks, each one corresponding to a particular criterion related to viewing, lighting or scanning conditions.

3.1 Artifacts-free color texture recovery

Our texturing method extends the work proposed in [4] so as to adapt it to the data flow produced by the scanning system presented above. The idea, summarized in Figure 2, is to weight each input picture by a mask (typically a gray scale image) which represents a per-pixel confidence value. The final color of a given surface point is then computed as the weighted average of all color contributions coming from the pictures into which this point is visible. Masks are built by the composition of multiple elementary masks, which are themselves computed by image processing applied either on the input image or on a rendering of the mesh performed from the same viewpoint. In the original paper, the approach has been designed for the general case, meaning that absolutely no information is available about the lighting environment or about how the geometry has been acquired. For this reason, only the following criteria related to viewing conditions have been initially considered, defined to deal with data redundancy in a way that ensures seamless color transitions:

- *Distance to the camera.* When a part of the object surface is visible in two different pictures, the one for which the viewpoint is the closest obviously contains a more accurate sampling of the color information. This elementary mask assigns to a pixel a confidence value which decreases when the distance to the camera of the corresponding surface point increases.
- *Orientation wrt. the camera.* Similarly, the more grazing the angle between the line of sight and the surface, the lower the quality of the sampling. This mask is computed as the dot product of the normal vector by the viewing direction, giving more importance to surface parts facing the camera.
- *Step discontinuities.* Due to inaccuracies during calibration, slight misalignment may be observed while collecting color contributions by reprojecting the mesh onto the picture. The most noticeable reprojection errors obviously

occur near the object silhouette and the step discontinuities. A lower weight is then assigned to these regions so as to avoid misalignment artifacts.

Although these masks are sufficient to avoid texture cracks, they obviously cannot handle self projected shadows or specular highlights, since knowledge about lighting environment is necessary. In our case, as explained before, both the positions of the viewpoint and the light (the projector lamp) are always exactly known. Moreover, the light moves with the scanner, which means that highlights and shadows are different for each frame, but also that the incident illumination angle changes. So, by setting appropriate masks, we can make prevailing image parts deprived of illumination effects. We then define the three following masks to account for illumination:

- *Shadows.* Since the complete geometric configuration of the scene is known, we can use a simple shadow mapping algorithm to estimate shadowed areas, to which a null weight is assigned.
- *Specular highlights.* Conversely to shadows, highlights partially depend on the object material, which is unknown. For this reason, we use a multi-pass algorithm to detect them. The first pass computes the object texture without accounting for highlights. Due to the data redundancy, the averaging tends to reduce their visual impact. During the subsequent passes, highlights are identified by computing the luminosity difference between the texture obtained at the previous pass and the input picture. This difference corresponds to our highlight removal mask. In practice, only two passes are sufficient.
- *Projector illumination.* This mask aims at avoiding luminosity loss during the averaging by giving more influence to surface parts facing the light source. The mask values correspond to the dot product between the surface normal and the line of sight.

We also introduce two other elementary masks to cope with the occlusions that are inherent to in-hand scanning. Indeed, if they are ignored, picture regions that do not correspond to the real object color might be introduced in the computation, leading to visible artifacts in the final texture. Thus, when digitization is performed with the dark glove, an occlusion mask is simply computed by thresholding pixel intensities, so as to discard the darkest picture regions. In the case of a digitization made by hand, we use the skin probability map produced by the scanner to generate the mask values.
Each elementary mask contains values in the range $]0, 1]$. They are multiplied all together to produce the final mask that selectively weights the pixels of the corresponding picture. During this operation, each elementary mask can obviously be applied more than once. The influence of each criterion can then be tuned independently, although we empirically determined default values that work quite well for most cases.
All operations consist in local computations, making the process suitable for a GPU support. Moreover, pictures are processed sequentially, which makes the memory consumption independent to the length of the video flow.

3.2 Assisted selection for texture hole filling

If each independent weighting mask represents the confidence of a given video frame, the total weight sum at a given surface point can be seen as a quality measurement of the color sampling at this point. Indeed, a low sum value may represent either a sparse picture coverage or the fact that the point always projects onto picture areas of low confidence. In both cases, the reconstructed texture can be considered as less reliable at this point than on the rest of the surface. In practice, regions for which the sum is relatively small mostly correspond to concavities or surface parts acquired only at grazing angles, where color defects arise due to the lack of quality in the input sampling.
This information can then be exploited advisedly to rectify holes and poorly sampled texture regions. While computing the texturing, we keep track of the total weight sum at each surface point, as well as of its minimum and maximum bounds for the whole surface. The user is then asked to choose a threshold in this range. All surface points for which the total weight sum is lower than this threshold are considered as belonging to a hole. In the current implementation, filling is finally performed by diffusing inside holes the colors of the valid points located on their boundaries. More sophisticated approaches based on texture inpainting can also be designed.

3.3 Normal map recovery

Since the light position is known for each video frame, shape-from-shading can be used to extract a bump map, similarly to [19]. Let p be a point on the object surface and F_i a video frame into which this point is visible. We then denote by the column vector l_i the unit incident light direction at p, by d_i the distance of p to the light, and by c_i the color intensity observed at p. If we assume that the surface is purely Lambertian, the following equation can be written:

$$\rho_d \ (n^T \ l_i) = d_i^2 \ c_i \tag{1}$$

where the column vector n is the normal at p and ρ_d is a constant depending on the light intensity, the surface diffuse albedo and the camera transfer function. If p is visible in several frames $\{F_i\}_{1 \leq i \leq N}$, the normal n that fits the best the N measurements can be found by solving the following equation:

$$\rho_d \ n = \arg\min \xi(X) \tag{2}$$

where ξ is a quadratic form defined as:

$$\xi(X) = (LX - C)^2 \quad \text{with } L = \begin{bmatrix} w_1 \ l_1^T \\ \vdots \\ w_N \ l_N^T \end{bmatrix} \text{ and } C = \begin{bmatrix} w_1 \ c_1 \ d_1^2 \\ \vdots \\ w_N \ c_N \ d_N^2 \end{bmatrix} \tag{3}$$

where w_i is a confidence factor for the i^{th} measurement F_i, corresponding to the weighting masks in our case. Solving equation 2 is equivalent to finding the value of X for which the derivative of ξ is null, which leads to the following solution:

$$\xi'(X) = 0 \quad \Longleftrightarrow \quad X = (L^T L)^{-1} \ (L^T C) \tag{4}$$

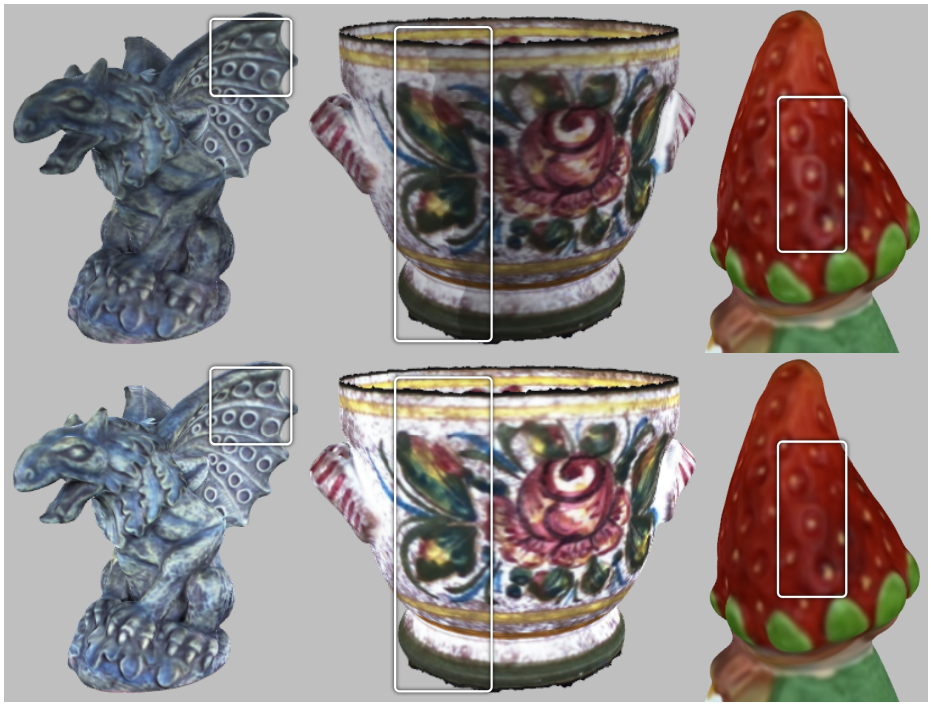

Fig. 3. Comparison of texturing obtained by a naive averaging of all input frames (*top row*) and the proposed approach using weighting masks (*bottom row*).

If we constrain the normal vector to be unitized, n is finally obtained by normalizing X. Since both matrices $(L^T L)$ and $(L^T C)$ can be constructed incrementally, normal extraction can be performed similarly to color reconstruction, by processing input pictures one by one on GPU. Both processings are then performed simultaneously, without introducing noticeable additional cost.

4 Results

Our texturing results are shown in the bottom row of Figure 3. The top row proposes a comparison to the results obtained by a naive approach that performs a direct averaging of color contributions, ignoring weighting masks. The most obvious difference that can be noticed is clearly the drastic loss of luminosity that occurs in the case of the naive approach. As expected, the *projector illumination* mask tends to increase the influence of image regions that corresponds to the most illuminated surface parts, which leads to a conservation of luminosity. Other improvements can be observed. For the Gargoyle model (*left*), we can see that fine details on the wings are much less blurry when the weights are introduced in the computations, thanks to the fact that the surface orientation with respect to the viewpoint is considered. For the Pot model (*middle*), the big vertical crack

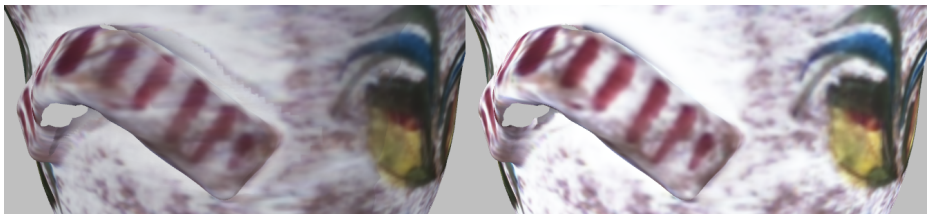

Fig. 4. Example of artifacts appearing due to misalignment errors.

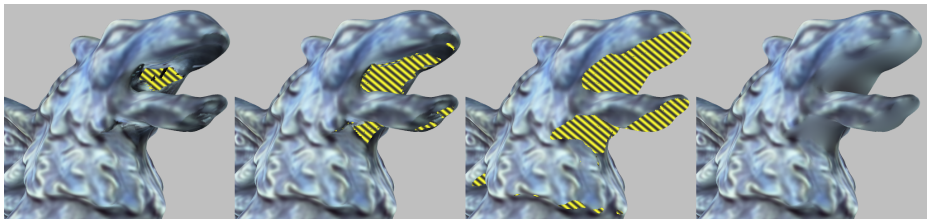

Fig. 5. Hole filling. Regions to process (indicated by yellow/black stripes) are selected by manually setting a threshold on the weight sum that results from the texturing.

in the white rectangle results from the fact that one portion of the surface was depicted by a much greater number of frames with respect to the adjacent one: this produces an imbalance among the number of summed color contributions, and the consequent abrupt change of color. Thanks to masks related to visibility and illumination criteria, with our approach the big crack completely disappears. For the Strawberry model (*right*), some regions are slightly brighter in the case of the naive texturing. They arise from strong specular highlights in the input pictures and are highly reduced in the case of our method. Figure 4 shows the typical artifact appearing at step discontinuities when misalignment errors are neglected. Once again, the use of the adequate mask permits to remove it.
Figure 5 illustrates the hole filling. The left image shows a region left empty by the texturing process. Around this hole, artifacts can be seen due to the fact that the surface was captured at grazing angle during the whole video, then leading to a color sampling of bad quality. The second and third images show how the selection based on the total weight sum adapts its shape when the threshold is changed. The right image finally shows the model once the selected region has been filled.
Figures 6 and 7 shows different renderings of the Gargoyle and the Strawberry models, with and without the normal map extracted by our system. As it can be seen, high-frequency shape details are clearly missing in the rough geometry acquired by the in-hand scanner. Thanks to our shape-from-shading approach, these details can be accurately captured and rendered afterward by bump mapping.
Computation times are reported on Table 1 and have been measured on an Intel i7 2.8GHz and a GeForce GTX 480. Recorded times correspond to two texturing

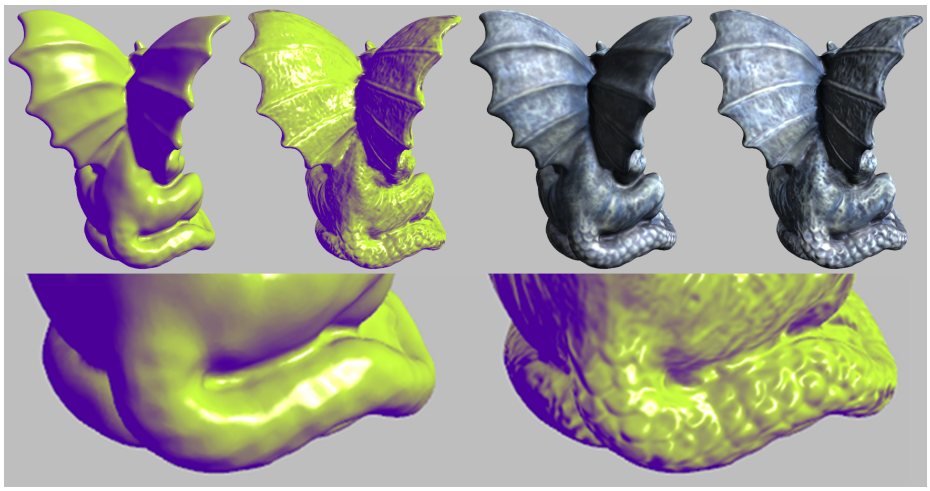

Fig. 6. The Gargoyle model rendered with and without the extracted normal map.

passes, since specular highlight removal has been activated. It can be noted that the additional computation cost introduced by the normal map extraction is negligible. It can be explained by the fact that color and normal reconstructions are performed simultaneously: the data loading on the graphics card (which is the main bottleneck of the process) has to be performed only once.
The parameters that must be set by the user are the followings: the hole filling selection threshold, for which a visual feedback similar to what is shown in Figure 5 is provided in real time, the iteration number for the diffusion process, which can be set progressively until having reached the desire result, and the number of applications for each elementary mask. As said before, default values working well for most cases have been determined. Nonetheless, considering the aforementioned computation times, the user can easily try several combinations by relaunching the texturing so as to see how different values impact the final result. The set of parameters is then really small and can be tuned in an easy and intuitive manner.

5 Conclusion

In this paper, we presented an automatic approach that produces an enhanced textured mesh for a model acquired using an in-hand scanner. The approach proposed is quite general, since it can be applied to any 3D scanning system that is based on structured light, and uses a video-camera as sensing unit. The advantages of the enhanced digital model are extremely important for all those applications where knowledge of the pure shape is not enough; pairing the overall shape model with data on color, surface reflection and high-frequency scale geometry detail is of paramount importance to produce high quality digital replicas of real objects. An advantage of our proposal is that it does not require to

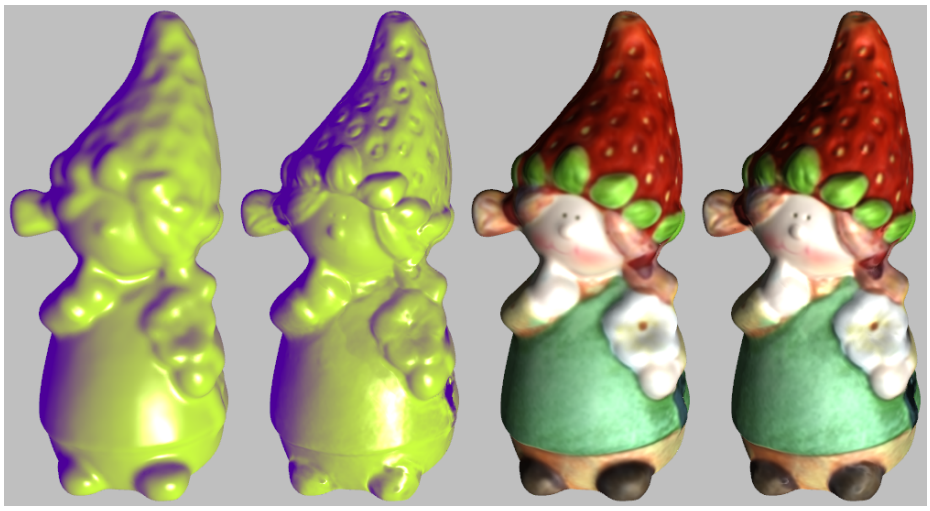

Fig. 7. The Strawberry model rendered with and without the extracted normal map.

Model	*No. of pictures*	*No. of highlight detection passes*	*Total texturing time (in seconds)*	
			without normal map	*with normal map*
Gargoyle	1070	2	194	198
Strawberry	2110	2	266	277
Pot	330	2	27.5	29

Table 1. Texturing times for different data sets.

modify the scanning hardware design, since it is based only on an automatic, improved processing of the input data.
Some possible future work is as follows. In the current implementation, texturing is performed as a post-processing task; but we could also study a solution to perform the texturing online, while the measurement is performed, to present immediate feedback to the user during the scanning session. The current bump map estimation tends to produce inconsistent normals when the sampling is too sparse; one possible idea to improve robustness is to perform a constrained computation by analyzing the dispersion of the cone defined by the sampling directions. Finally, displacement mapping can be added on the base of the information contained in the normal map, to enrich the rough geometry given by the scanner with a better visualization.

Acknowledgment

This work has been performed in the frame of the ERCIM fellowship program. The research leading to these results has received funding from the EU 7^{th} Framework Programme (FP7/2007-2013), through the 3D-COFORM project, under grant agreement n. 231809.

References

1. N. Bannai, A. Agathos, and R.B. Fisher. Fusing multiple color images for texturing models. In *3DPVT04*, pages 558–565, 2004.
2. A. Baumberg. Blending images for texturing 3d models. In *BMVC 2002*. Canon Research Center Europe, 2002.
3. F. Bernardini, I.M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Tr. on Visual. and Comp. Graph.*, 7(4):318–332, 2001.
4. M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno. Masked photo blending: mapping dense photographic dataset on high-resolution 3d models. *Computer & Graphics*, 32(4):464–473, Aug 2008.
5. M. Callieri, P. Cignoni, and R. Scopigno. Reconstructing textured meshes from multiple range rgb maps. In *7th Int.l Fall Workshop on Vision, Modeling, and Visualization 2002*, pages 419–426, Erlangen (D), Nov. 20 - 22 2002.
6. M. Chuang, L. Luo, B.J. Brown, S. Rusinkiewicz, and M. Kazhdan. Estimating the laplace-beltrami operator by restricting 3d functions. In *Proceedings of the Symposium on Geometry Processing*, pages 1475–1484, July 2009.
7. M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. *Computer Graphics Forum*, 28(7):1755–1764, 2009.
8. M. Dellepiane, M. Callieri, M. Corsini, P. Cignoni, and R. Scopigno. Improved color acquisition and mapping on 3d models via flash-based photography. *ACM Journ. on Computers and Cultural heritage*, 2(4):1–20, Feb. 2010.
9. M. Dellepiane, M. Callieri, R. Marroquim, P. Cignoni, and R. Scopigno. Flow-based local optimization for image-to-geometry projection. *IEEE Transaction on Visualization and Computer Graphics*, (in press), 2011.
10. J. Dorsey, H. Rushmeier, and F. Sillion. *Digital modeling of material appearance*. Morgan Kauf./Elsevier, 2007.
11. M. Eisemann, B. De Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating textures. *Computer Graphics Forum (Proc. Eurographics EG'08)*, 27(2):409–418, 4 2008.
12. T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. Minimizing user intervention in registering 2D images to 3D models. *The Visual Computer*, 21(8-10):619–628, sep 2005.
13. R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless montage for texturing models. *Comput. Graph. Forum*, 29(2):479–486, 2010.
14. O. Hall-Holt and S. Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *ICCV 2001*, pages 359–366 vol.2, 2001.
15. K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, and Y. Okamoto. The great buddha project: digitally archiving, restoring, and analyzing cultural heritage objects. *Int. J. Comput. Vision*, 75(1):189–208, 2007.
16. F. Larue and J-M. Dischler. Automatic registration and calibration for efficient surface light field acquisition. In *7th VAST International Symposium on Virtual Reality, Archeology and Cultural Heritage*, pages 171–178. Eurographics, 2006.
17. K. Pulli, H. Abi-Rached, T. Duchamp, L. Shapiro, and W. Stuetzle. Acquisition and visualization of colored 3d objects. In *Proc. of ICPR 98*, pages 11,15, 1998.
18. V. Rankov, R. Locke, R. Edens, P. Barber, and B. Vojnovic. An algorithm for image stitching and blending. In *SPIE*, volume 5701, pages 190–199, 2005.

19. Holly E. Rushmeier, Gabriel Taubin, and André Guéziec. Appying shape from lighting variation to bump map capture. In *Proc. of the Eurographics Workshop on Rendering Techniques '97*, pages 35–44, London, UK, 1997.
20. S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 21:438–446, July 2002.
21. I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai. Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. *Int. J. Comput. Vision*, 78:237–260, July 2008.
22. J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. In *EGRW '03*, pages 141–149, 2003.
23. T. Weise, B. Leibe, and L. Van Gool. Fast 3d scanning with automatic motion compensation. In *IEEE CVPR'07*, June 2007.
24. T. Weise, T. Wismer, B. Leibe, , and L. Van Gool. In-hand scanning with online loop closure. In *3DIM09*, October 2009.
25. L. Zhang, B. Curless, and S.M. Seitz. Spacetime stereo: shape recovery for dynamic scenes. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II – 367–74 vol.2, june 2003.