3D FACIAL MERGING FOR VIRTUAL HUMAN RECONSTRUCTION

Rafael Pagés and Francisco Morán

ABSTRACT

There is an increasing need of easy and affordable technologies to automatically generate virtual 3D models from their real counterparts. In particular, 3D human reconstruction has driven the creation of many clever techniques, most of them based on the visual hull (VH) concept. Such techniques do not require expensive hardware; however, they tend to yield 3D humanoids with realistic bodies but mediocre faces, since VH cannot handle concavities. On the other hand, structured light projectors allow to capture very accurate depth data, and thus to reconstruct realistic faces, but they are too expensive to use several of them. We have developed a technique to merge a VH-based 3D mesh of a reconstructed humanoid and the depth data of its face, captured by a single structured light projector. By combining the advantages of both systems in a simple setting, we are able to reconstruct realistic 3D human models with believable faces.

Index Terms — 3D reconstruction, human 3D models, visual hull, structured light, 3D mesh merging.

1. INTRODUCTION

The 3D graphics world is still in a very exciting moment: every day, more and more applications use 3D models in different ways. New tools are being developed not only in the traditional fields related to 3D graphics (video-games and movies), but also in other industries such as security, topography or medicine, which need 3D models as well. Although these industries frequently hire professional 3D artists to create their models, they sometimes need more realistic 3D meshes. As a consequence, there has been a great advance in 3D acquisition techniques, both in software applications and in hardware devices with different accuracies. These devices are more and more affordable, and the general user is now nearer to the latest systems to reconstruct 3D scenes.

Due to their use in many of the mentioned applications, human bodies and faces are among the most demanded models, and are very often captured from real people's bodies and faces thanks to techniques based on the extraction of the visual hull (VH) of the model from its silhouettes in a set of images taken simultaneously [1]. The VH can be understood as the 3D intersection of a group of human-shaped pseudo cones, and may be meshed, for example, after applying the marching cubes algorithm to the voxelized VH volume. These techniques are popular because their input is a set of ordinary images taken with standard cameras, but they all share the same big drawback: by construction, VH cannot handle concavities. Although in most of the body this is not very noticeable, the results are specially disturbing in the face, which we pay a lot of attention to, and where we have important concavities such as the eye sockets.

This paper presents an automatic algorithm able to merge the depth information obtained with a single structured light projector focused on a person's face into a (typically, VH-based) 3D mesh representing her/his body. The topological structure (e.g., the manifoldness) of the initial mesh is preserved, but in the facial area its polygonal resolution is increased and its vertices are moved to match as faithfully as possible that specific person's facial features, captured by the very detailed depth map output by the structured light projector.

2. PREVIOUS WORK

There are many different techniques to acquire a human body, and more specifically to obtain a mesh representing a face with very good accuracy. We mainly divide these techniques into two big categories: active vs. passive. The difference between them is that, contrary to passive techniques, which only use photographs of the scene to create a 3D model, active ones require specific hardware which somehow interacts with the scene, e.g., a laser scanning system, or a projector casting a light pattern onto the scene, paired with a special camera system registering the light pattern deformations, and guessing the 3D geometry from them.

Among the active techniques with best results, we find the one proposed by Fechteler et al. [2] who combine the result of a structured light scanning with an image taken with a DSLR camera. The process includes not only algorithms to extract the 3D information from the color pattern projected onto the face of the subject, but also to triangulate the resulting 3D cloud to obtain a regular triangular structure. On the other hand, among the passive techniques we find the one proposed by Beeler et al. [3], which obtains incredible results using a set-up of professional cameras, thanks to the mesoscopic augmentation. Choi et al. [4] have developed a system which can provide different results depending on the input: it works with one or several images. Of course, the results in the second case are much better, since the facial landmarks it needs are obtained in 3D, so their position is very reliable. Another technique, in this case, a semi-automatic one, is the one by Brunton et al. [5], who obtain very accurate facial meshes by using wavelets in a Bayesian environment. There are also systems which need a database to be able to reconstruct the 3D face. This is the case of Blanz and Vetter's [6] technique, which obtains good results (and is even able to reconstruct famous people with just one image) by allowing the user to determine a set of parameters which are later computed and included in a cost function. The main problem of this technique is the need of a complete database with different sexes and races, and of high computational resources to calculate the cost function.

Although the results are quite good in all these techniques, none of them consider the possibility of merging this facial mesh with another representing the whole body to produce a complete human model. To the best of our knowledge, this has been only

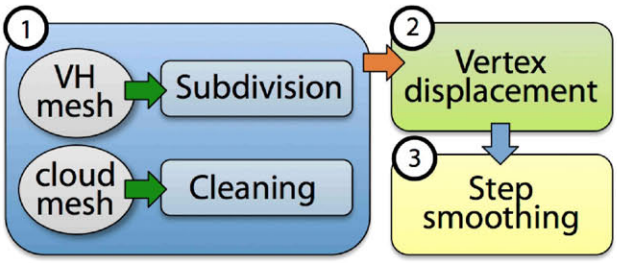

Figure 1. General process overview

considered in another technique of ours [7], where facial landmarks representing important feature points in the subject's face are used to create a basic geometric transformation of a generic facial model by using pyramids formed by these landmarks and the barycentre of the head. However, even tough the results are good, transforming a generic 3D model will never be as accurate as obtaining the concrete subject's facial geometry.

3. PROPOSED TECHNIQUE

The only data our algorithm needs are: a 3D mesh representing the subject's body, which has typically been obtained using a VH-based technique, and is thus called "**VH mesh**" below; a **frontal image** of the subject; and a second 3D mesh for the facial area, resulting from scanning the face with a structured light projector, that we will call "**cloud mesh**" because the output of that scan is usually just a point cloud, from which we obtain a mesh with the system proposed by Congote et al. [8].

The process consists of the following stages (see Figure 1): 1. preparation of the two input meshes, which are subdivided and cleaned as needed; 2. vertex displacement (specific vertices of the subdivided VH mesh are moved to match the information provided by the cleaned cloud mesh); 3. step smoothing (the transition between the displaced and fixed regions must usually be smoothed by using a customized interpolation). The subsections below elaborate on each of these stages.

3.1. Preparation of the two input meshes

By feeding the frontal image of the subject to an OpenCV-based face detection system, we obtain an ellipse containing the face. Then, thanks to the calibration parameters of that camera, we are able to determine the height and width of the head, and of course place the camera in the 3D scene. The line defined by the centres of the camera and the (back-projected into 3D) ellipse intersects the VH mesh in two points, that allow us to determine the depth of the head and complete our set of head dimensions. From them, we obtain an approximate position of the head barycentre, and also which vertices and triangles of the VH mesh are in its facial section. The resolution of this facial section is refined with Dyn's butterfly subdivision scheme [9] to adapt it to that of the cloud mesh, which is much denser than the VH mesh.

The last phase of this initial stage consists in adapting the cloud mesh for the subsequent vertex displacement. As a result from the scanning process, which sometimes yields residual vertex information, the cloud mesh presents irregular triangles in areas which are too far from the projector, or close to holes in occluded areas. Removing these triangles is necessary for the next stage of the process, because they could interfere with the vertex displacement and propagate errors to the final model. Using the head dimensions, we define a set of cutting planes and then a mask containing all the useful information of the cloud mesh.

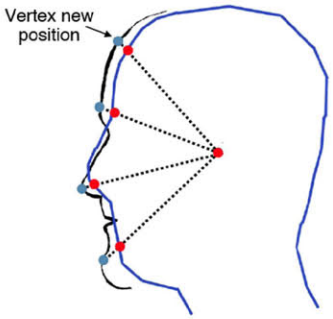

Figure 2. Vertex displacement: the vertices from the VH mesh (shown in blue) are moved until they hit the cloud mesh (in black).

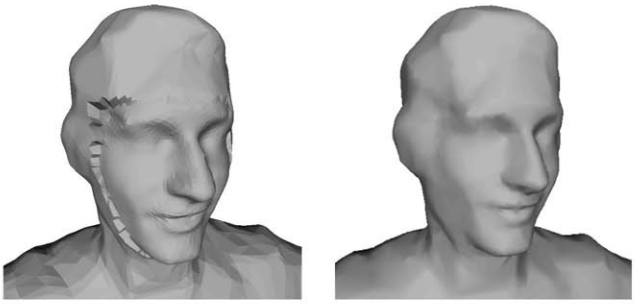

Figure 3. Step smoothing: vertex displacement typically causes a visually uncomfortable step to appear between the fixed vs. moved vertex regions of the mesh. The example shown here is the worst case we have found, yet the results after step smoothing are acceptable.

3.2. Vertex displacement

Once the cloud mesh has been cleaned and the facial section of the VH mesh subdivided to match the polygonal resolution of the cloud mesh, we are ready for the vertex displacement. In this stage, no additional information (vertices or triangles) is extracted from or created in any of the two meshes: we only move existing vertices of the subdivided VH mesh towards the cloud mesh.

From the previously calculated head barycentre, we cast a ray for each vertex in the facial section of the subdivided VH mesh: see Figure 2. If this ray intersects the cloud mesh in any point, either further away or closer than the original vertex was, we move the vertex to that intersection point and mark it as *MOVED*.

It is important to remark that, as in this kind of transformation all rays converge in one point (the barycentre), we can guarantee that the topological characteristics of the mesh are preserved. In other words, after applying the vertex displacement, the mesh will be as manifold as it was before, and there there will not be any intersecting triangles or topological errors.

3.3. Step smoothing

As illustrated by Figure 3 (left), a very visually annoying "step" may appear between the displaced and the fixed sections of the mesh. This phenomenon is normal when working with different 3D meshes, but is even more common when we need to use meshes obtained from a VH process, because it tends to increase the volume of the models. Although this artifact is not always no-

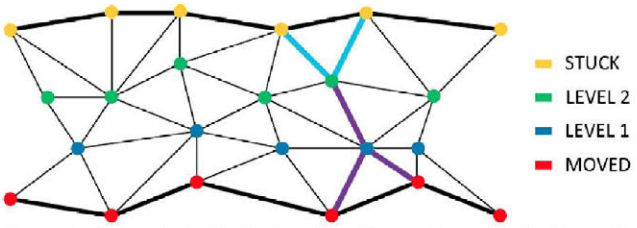

Figure 4. Vertices in the "bridge" region of the mesh are classified in levels according to their position with respect to the *MOVED* ones.

ticeable on the VH mesh considered individually, if we want to merge into it information coming from a very accurate source, as the structured light projector is, the 3D model has to undergo a smoothing process.

This step smoothing process will only affect a specific region in the 3D model: a "bridge" area in the fixed region adjacent to the displaced one. The width of this bridge is calculated using the head dimensions and it can vary depending on the model. We mark the already displaced vertices as *MOVED* and the ones not included in the bridge area as *STUCK*. The smoothing process is based on a customized interpolation of the position of both *MOVED* and *STUCK* vertices.

To be able to perform this interpolation we first need to create a hierarchical tree structure thanks to which, for each vertex in the bridge, we can easily find the shortest path from that vertex to both "shores" by moving along edges (see Figure 4, where a level is assigned to each vertex in the bridge region). This tree makes it easier to calculate distances, as we have already registered triangle dimensions. For each vertex $\mathbf{v}$, we have two values, d_M and d_S, which correspond to the distances to the *MOVED* and *STUCK* shores respectively, calculated as:

$$d_k = \sum_{i=0}^{n} |\mathbf{v_i} - \mathbf{v_{i+1}}| \qquad k = \{M, S\},$$

where n is the number of levels from $\mathbf{v}$ to the considered shore, and $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$ are two successive vertices along the mentioned shortest path.

The specific way in which we calculate the new position of each vertex in the bridge is by performing a linear interpolation between the nearest *STUCK* and *MOVED* vertices over the line defined by the barycentre and the vertex. The model thus preserves its topological structure, and no artifacts or errors are introduced.

4. RESULTS

As illustrated by Figure 5, the results obtained after the process are quite good. In the left column, we can see the initial VH meshes, which do not have enough polygonal resolution, especially in the face region. In the middle, we have the result of applying the structured light scanning and meshing the resulting depth map (i.e., point cloud), with all the residual polygons and surfaces which are removed before merging both models. In the right, we can see the final models, whose polygonal resolution has been increased considerably, and whose shape is very faithful to each subject. The system even works when facing specially problematic VH models, like the one shown in Figure 3, where the step between the displaced and fixed regions was considerable, but is smoothed correctly.

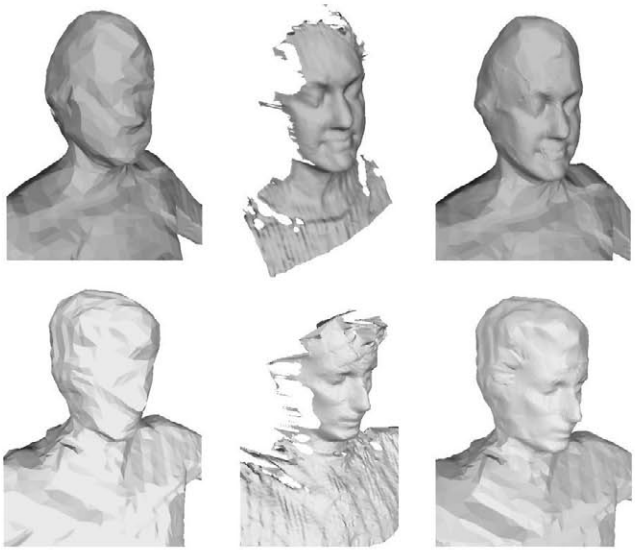

Figure 5. Plain mesh results: for each of two subjects (top and bottom rows), three "naked" 3D models have been rendered in flat shading mode corresponding to the original VH mesh (left column), cloud mesh (middle), and final mesh after the merging process (right).

Figure 6. Textured mesh results: for each of two subjects (top and bottom rows), both the original VH mesh (left column) and the final mesh (right) have been rendered with our multi-texturing technique [10], which cannot help severe inaccuracies in the face geometry due to its VH-based 3D reconstruction, as illustrated by the top-left model; however, these problems are solved by our mesh merging process.

Our mesh merging process is not only able to obtain realistic models, but also very efficient: it needs less than 20 seconds (average for all models) on a laptop computer with an Intel Core i5 CPU @ 2.4 GHz, 4 GB of RAM, and no GPU acceleration.

Figure 6 shows that the results are even better if we apply our multi-texturing technique [10] which uses all the images taken to elaborate the VH mesh. It is also interesting to compare the results of multi-texturing the original VH mesh, since we can see how some facial features are wrongly repeated across the face of the top-left model due to the too coarse simplification of the facial geometry. This conveys a very visually displeasing feeling. On the other hand, the final model obtained by merging the VH mesh and the cloud mesh has the proper shape, which represents the subject's face with high accuracy, so the texture is mapped correctly on it and makes it look very realistic. Models obtained with the whole process, including multi-texturing, may be used for many different applications requiring realistic human 3D models.

It is very important to note that all cameras must be correctly calibrated, both the ones used for extracting the VH mesh and that/those associated to the structured light projector. Indeed, a slight displacement of the cloud mesh with respect to the VH mesh may be unnoticeable, but when the final 3D mesh resulting from the merging process is textured, the results can easily be visually catastrophic. Our calibration process follows the approach proposed by Ronda et al. [11], which avoids possible misalignments or rotation distortions.

5. CONCLUSIONS

The fast development that computer hardware is experiencing lately is boosting in particular 3D graphics applications. This sector has always been related to video-games and desktop PCs, but due to the huge increase of the graphics capabilities of laptops and even mobile devices, we find more and more realistic 3D graphics applications able to run smoothly on them. This is why not only the 3D industry, but also researchers, focus on innovative ways to create believable 3D models, and in particular human 3D models as similar as possible to specific real people.

In this paper we have presented an innovative and efficient system to merge 3D mesh information obtained from different sources. The results of our mesh fusion process are visually correct and, after applying our multi-texturing technique [10], we can confirm that they are also correct geometrically, and even more realistic. Moreover, our system is completely automatic (it needs no human interaction whatsoever) and its algorithm mathematically simple, so it was adopted in a more complex system for virtual human reconstruction for which a patent is pending.

The main disadvantage of our process is perhaps its dependency on an accurate calibration of the cameras to avoid visual artifacts in the final models. Also, to calculate the approximate barycentre of the head section, we rely on the OpenCV-based head detection algorithm which, in case of failure, would lead us to a very erroneous 3D model. However, this latter dependency is less worrying since we could also calculate the barycentre by just averaging the coordinates of every vertex in the head section.

As future work, we consider optimizing the implementation of our mathematically simple algorithm for an even more efficient performance (e.g., by applying parallel programming techniques to it) and, more substantially, adding semantic information to the model, to be able to animate it with different facial expressions and use it in applications such as video-games. This could be done by using a feature detection system in the frontal image of the subject, and casting rays from the centre of the camera to the (back-projected) detected 2D feature points, which would intersect the final 3D mesh at or close to its sought feature vertices.

6. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Government under project TEC2010-20412 (Enhanced 3DTV). We also want to thank Sergio Arnaldo and our colleagues from Telefónica I+D in Barcelona for their invaluable cooperation.

7. REFERENCES

[1] J. Gallego, M. Pardas, and G. Haro, "Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model," in *Intern. Conf. on Image Processing (ICIP)*. IEEE, 2009, pp. 3205–3208.

[2] P. Fechteler, P. Eisert, and J. Rurainsky, "Fast and high resolution 3D face scanning," in *Intern. Conf. on Image Processing, 2007 (ICIP)*. IEEE, 2007, vol. 3, pp. 81–84.

[3] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 40, 2010.

[4] J. Choi, G. Medioni, Y. Lin, L. Silva, O. Regina, M. Pamplona, and T.C. Faltemier, "3D face reconstruction using a single or multiple views," in *Intern. Conf. on Pattern Recognition (ICPR)*. IEEE, 2010, pp. 3959–3962.

[5] A. Brunton, J. Lang, E. Dubois, C. Shu, et al., "Wavelet model-based stereo for fast, robust face reconstruction," in *8th Canadian Conf. on Computer and Robot Vision (CRV 2011)*, 2011.

[6] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conf. on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[7] R. Pagés, S. Arnaldo, and F. Morán, "Face lift surgery for reconstructed virtual humans," in *2011 Intern. Conf. on Cyberworlds*. IEEE, 2011, pp. 249–253.

[8] J. Congote, I. Barandiaran, J. Barandiaran, M. Nieto, and O. Ruiz, "Face reconstruction with structured light," in *8th Intern. Conf. on Computer Vision and Applications. VISAPP'10*, 2010, pp. 149–155.

[9] N. Dyn, D. Levine, and J.A. Gregory, "A butterfly subdivision scheme for surface interpolation with tension control," *ACM transactions on Graphics (TOG)*, vol. 9, no. 2, pp. 160–169, 1990.

[10] R. Pagés, S. Arnaldo, F. Morán, and D. Berjón, "Composition of texture atlases for 3D mesh multi-texturing," *Proc. Eurographics-IT 2010 Conf.*, pp. 123–128, 2010.

[11] J.I. Ronda, A. Valdés, and G. Gallego, "Line geometry and camera autocalibration," *Journal of Mathematical Imaging and Vision*, vol. 32, no. 2, pp. 193–214, 2008.