

Automatic System for Virtual Human Reconstruction with 3D Mesh Multi-Texturing and Facial Enhancement

Rafael Pagés, Daniel Berjón*, Francisco Morán

Grupo de Tratamiento de Imágenes, E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Abstract

The present paper presents a fully automatic low-cost system for generating animatable and statically multi-textured avatars of real people captured with several standard cameras. Our system features a novel technique for generating view-independent texture atlases computed from the original images, and two proposals for improving the quality of the facial region of the 3D mesh: a purely passive one implying no additional cost, and another based on active techniques such as structured light projection.

Keywords: Human avatar generation, view-independent multi-texturing, 3D face reconstruction

1. Introduction

The proliferation of video games and other applications of computer graphics in everyday life demands a much easier way to create animatable virtual human characters. Traditionally, this has been the job of highly skilled artists and animators that painstakingly model, rig and animate their avatars, and usually have to tune them for each application. The emergence of Virtual Worlds (VWs) also calls for practical and cost-effective ways to produce custom models of actual people.

In this paper we present a patent-pending (ES-P201230768) system for automatically generating an animatable avatar of any person, a three-dimensional (3D) photo booth thanks to which anyone can use, in games or VWs, their own likeness instead of a predefined avatar from a library. While our system cannot match the accuracy of the best professional 3D capture systems, it does offer very reasonable quality for the kind of applications mentioned above, and the big advantage of not requiring either specific hardware or manual tweaking in any phase of the model generation.

We start by describing in Section 2 how we built an initial version of our system by putting together

state of the art methods designed to reconstruct a 3D object (in this case, a real person) photographed by several standard cameras. The approach we chose, by no means revolutionary, consists in: i) approximating with a triangular 3D mesh the subject's shape, inferred by its silhouettes as seen from the different cameras; ii) mapping onto that mesh, to make it look more realistic, a texture taken from the images captured by the cameras; iii) adding the rigging and skinning information to make it possible to animate the avatar. This initial system was disappointing due to two main shortcomings: on one hand, its (naïve) texture mapping technique produced many artifacts; on the other, the shape approximation of the avatar's facial region was too coarse, which was unacceptable because we humans are much less tolerant to errors in an avatar's face than, say, in its leg. We therefore improved those two aspects of our initial system by developing novel multi-texture mapping and facial reconstruction methods that we describe in Sections 3 and 4, each of which starts with a Subsection where we review the corresponding state of the art, and then summarizes our techniques and results. Section 5 concludes the paper.

2. Initial system

The initial basic system, implemented as mentioned above, included the stages shown in Figure

*Corresponding author. Tel./Fax: +34.913367353
Email addresses: rps@gti.ssr.upm.es (Rafael Pagés),
dbd@gti.ssr.upm.es (Daniel Berjón), fmb@gti.ssr.upm.es
(Francisco Morán)

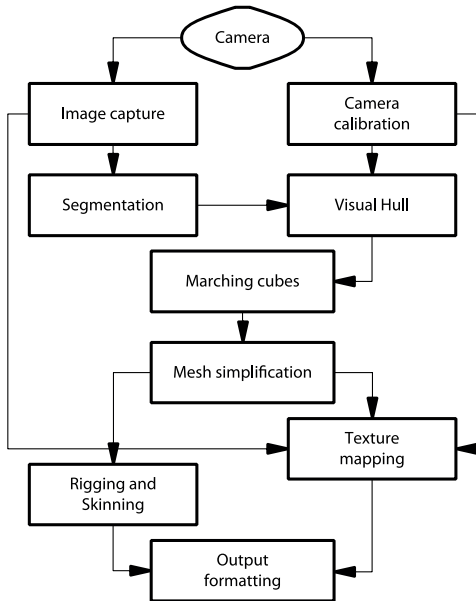


Figure 1: Flow and dependencies between the different stages of the initial system. The new techniques proposed in Section 4 take place after the “Mesh simplification” step, whereas the techniques proposed in Section 3 replace the original naïve texturing technique.

1 and described briefly below.

2.1. Used techniques

Foreground segmentation. Since the proposed system uses a modelling strategy of the type *shape from silhouette*, the input images from each camera must be segmented to find the user’s silhouettes. There are a number of techniques for separating foreground from background, the most traditional ones being based on chroma key, which requires a carefully controlled illumination and a background physically painted in a solid color (usually bright green or blue), but makes foreground segmentation very straightforward. However, modern foreground segmentation techniques [1, 2] achieve comparable results without imposing such strong requirements on the capture room, by means of statistical models of background and foreground.

Camera calibration. When a camera captures a scene, objects in 3D space are projected onto a 2D plane, thus destroying one dimension worth of information. In addition, this projection process is not performed in the same way for every camera: each one introduces a different transform not only depending on its position and orientation, which

determine its *extrinsic parameters*, but also on its optical system and sensor (*intrinsic parameters*); actually, no two different cameras, even of the same make and model, yield exactly the same projective transforms. The process of obtaining these parameters is known as *camera calibration* and, since all camera-based 3D reconstruction techniques rely on them to correctly project data from the images back into 3D space, a method thereof is necessary. There are numerous such methods that usually rely on taking several pictures of a known object (most often a chessboard) and/or verifying mathematical constraints for a point seen by several cameras. Specifically we have used the method described by Ronda et al. [3].

Visual Hull. The *Visual Hull (VH)* is defined as the intersection of the generalized cones defined by the optical centre of each camera and the corresponding silhouette of the previously segmented foreground object. In practical terms, it is the maximum volume that is consistent with the silhouettes of an object. It therefore depends not only on the shape of the object but also on the number and placement of the cameras: the more cameras, the better the approximation to the actual shape of the object (within its intrinsic limitations: a concavity cannot be resolved from silhouettes, by definition). There are two major methods to compute it: polyhedral and volumetric; we have used the latter approach [4], in which the reconstruction space is partitioned into a regular grid of volume elements (voxels), each of which is projected onto each camera and considered to belong to the VH if and only if no projection falls outside the silhouette. This method is simple and works well even for cameras that cannot see the whole subject.

Marching cubes. In order to convert the volume previously obtained into a surface, to be later approximated by a triangular mesh that can in turn be textured and rigged, we have used the *Marching Cubes (MC)* algorithm [5]. This algorithm replaces each of 28 possible $2 \times 2 \times 2$ voxel configurations with a specific set of triangles, and guarantees that each 8-connected volume results in a 2-manifold watertight mesh.

Mesh simplification. The mesh generated by the MC stage is topologically convenient, but has two problems that must be solved: its density of triangles is very high if a meaningful voxel resolution

is employed, which makes the generated model unpractical to use, and it also presents aliasing effects due to the volume discretization in voxels. To avoid these problems, the mesh is first smoothed to remove artifacts using HC Laplacian smoothing [6], and then simplified to reduce the number of triangles using edge decimation based on quadric error metrics [7]. The resulting mesh lacks detail in the facial area; in Section 4 we detail how we improved it.

Texturing. Arguably the most important feature of a model from a subjective quality perception point of view, texturing information can be extracted from the original images. A logical strategy to perform texture mapping consists in ranking each of the input images according to some metric to determine which image will be used to texture each triangle. However, this leads to store/transmit all the input images, which is inefficient. In addition, most systems using this approach employ view-dependent metrics [8], so that the chosen texture for each triangle depends, in run time, on the viewpoint location. While this approach yields good quality, it requires a specific renderer and hinders the reutilization of the model with generic rendering engines; in Section 3 we detail how we generate a single non-view-dependent texture atlas that provides both space-efficiency and generality.

Rigging and skinning. Skeletal animation is the usual technique to animate 3D characters. We use it too in the present system to allow animation of the human models obtained from the reconstruction process. Bone-based animation requires two kinds of information: rigging describes the internal skeleton structure that defines where the joints are located and how they are connected, whereas skinning describes which segments of the skeleton or joints affect the position of which vertices and how much (the position of a vertex may depend on more than one joint). While rigging a model in a completely unknown pose is a hard problem, for this system it is acceptable to ask the user to adopt a standard pose. Given this approximately known pose, rigging can be done using the algorithm proposed by Baran and Popović [9] and skinning can be solved using the method proposed by Hétroy et al. [10].

Integration of results. Finally, all the different kinds of information (mesh, texture, rigging and



Figure 2: Naïve result obtained by the initial system. Although there are areas where the texture should be very uniform, irregular illumination conditions yield very different texture patches. In addition, the facial area obviously lacks detail.

skinning) must be integrated in a way that allows the model to be easily reused from different programs. Most popular open formats for 3D meshes cannot express rigging and skinning or lack proper application support, and most formats that do are proprietary and tightly linked to a specific content creation tool or engine. Fortunately, there is an emergent XML-based open format, COLLADA [11], which has enough expressive power to convey all this information and is becoming a widely adopted industry standard. This makes it more likely that the generated models will be reusable over time; therefore we have chosen this format as our output.

2.2. Shortcomings

Our initial system yielded poor results mainly because of artifacts in the textures and insufficient geometric detail in the facial area of the subjects, as shown by Figure 2.

3. Multi-texture mapping

3.1. Introduction and state of the art

There are different approaches in the multi-texturing field, all of which face two main problems: visible seams in the final model and misalignments among different textured sections of the mesh. Although literature with respect to these problems exists, we found no approach solving our particular issue. For example, Lempitsky and Ivanov [12] proposed a system where misalignments are corrected by post-processing the frontiers of the different patches obtained by Markov random fields, which has problems when the VH reconstruction is very coarse. Other examples are the ones proposed by Rocchini et al. [13], and Gal et al. [14], which iteratively correct the misalignments (both in color and illumination) with image transformation across the patch borders. These techniques consume considerable computer resources and also have trouble with very irregular 3D meshes. Nevertheless, our system is based on the interpolation of the color information provided by several original images for each triangle, which avoids discontinuities and seams in the texture without any iterative process. Our algorithm for selecting which cameras provide the best color information for each particular triangle, and for blending the corresponding images to create the texture atlas, is partially described in a paper published in EG-IT'10 [15].

3.2. Process description

The first step consists in pre-processing the input images. Due to possible errors in the segmentation stage, unwanted remains of background color (which is typically very bright) could derive in very noticeable errors in the texture atlas. The pre-processing step consists in the following: for each image, its input foreground mask is eroded in order to remove any left-over background pixels from the silhouette contour. Then, the in-painting algorithm proposed by Telea [16] is applied in each color image over the area defined by its corresponding eroded mask. Figure 3 shows the result of this pre-processing step, thanks to which we avoid the unwanted background color in the final model.

The second step consists in unwrapping the 3D mesh onto 2D patches. This is done with a zero-distortion approach (as opposed to systems based on parameterization [17]), which geometrically transports each triangle onto a plane, taking into account possible triangle intersection when we

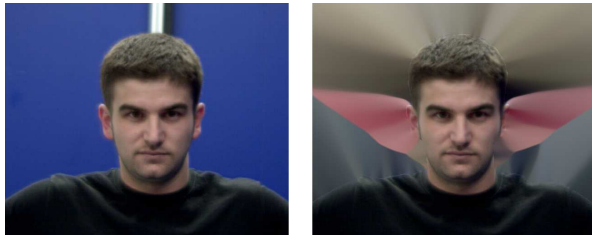


Figure 3: Pre-processing of the input images. Background color (present in left image) is removed so it is not included in the final texture atlas (right image).

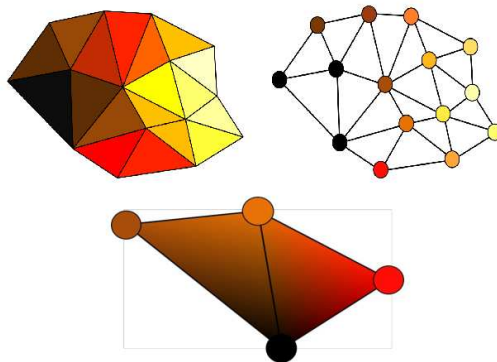


Figure 4: Ratings interpolation. Firstly, a rating is assigned to each triangle. Secondly, these ratings are averaged in every vertex of the mesh. Finally, ratings are interpolated across the triangles to create a smooth texture atlas without discontinuities.

unfold the mesh. This process yields a set of 2D patches which are properly packed in order to save space.

Once the structure of the atlas is calculated, color information needs to be added. The main problem every multi-texturing algorithm faces is the correction of misalignments and visible seams that appear in the model, as they normally use just one image to texture each triangle. On the contrary, the method we propose uses the information provided by a group of cameras (instead of just one) to texture each triangle with an intelligent blending system, so that seams and discontinuities are avoided without any iterative process.

In our system, every triangle is back-projected onto each input image, which is ranked according to the size of this projected area. In this way, it is possible to determine a criterion to choose a camera to be more influential than others and we can discard cameras that do not see the triangle at all, or see it



Figure 5: Top left: texture atlas resulting from the complete multi-texturing process. Top right: corresponding textured model. It is possible to see how discontinuities present in Figure 2 have been completely removed. Bottom: another reconstructed and textured model showing how folds in the pull-over are perfectly reproduced, even though the color information comes from different input images.

“from behind”. In addition, we determine a score for each vertex (and each camera) by averaging the ones of the triangles sharing it. These scores are linearly interpolated across all triangles as shown in Figure 4, yielding a rating $r_{i,j}$ for each pixel p_i and each camera j . To calculate the final color value for each pixel p_i , we first unwarp every triangle to bring the mixing problem to a common scenario using the smart scanline approach proposed by Wolberg [18], where texture misalignments are avoided. Finally, we calculate the final color Clr_i of pixel p_i as a weighted average of the colors provided by each camera for P_i (the 3D point corresponding to the pixel p_i), $Clr_{i,j}$:

$$Clr_i = \left(\sum_j r_{i,j} \cdot Clr_{i,j} \right) / \left(\sum_j r_{i,j} \right). \quad (1)$$

3.3. Results

Results show the good performance of our technique: see in Figure 5 that no discontinuities or seams are visible in the textured models. However,

even with this improved multi-texturing algorithm, some areas lack polygonal resolution. In particular, facial features are not well reconstructed, mainly due to the inability of the VH technique to handle concavities, which are essential in the human face. A selective facial mesh refinement process is therefore necessary as well.

4. Facial reconstruction improvements

4.1. Introduction and state of the art

We have designed and implemented *two different alternatives* to improve the reconstruction of the human model face: one based on an *active* approach and another based on a *passive* one.

The difference between active and passive approaches is that the former need additional hardware in the process, while the latter use exclusively images provided by standard cameras. Normally, active approaches produce more realistic results, as we can see, for instance, in the work of Fechteler et al. [19], which uses a structured light projector to overlay a color pattern on the face of the subject. There are also other advanced active techniques, such as the one proposed by Fanwen et al. [20], which gives very good results thanks to a 3D scanning system based on a sinusoidal pattern.

However, the additional hardware of all active approaches complicates the calibration of the complete 3D capture system, and is normally expensive. Passive approaches, on the other hand, do not use any hardware other than the cameras already needed to take the images, and yet some of them produce really good results, and allow for other interesting possibilities. One of the algorithms with better results is the one proposed by Beeler et al. [21], which uses *mesoscopic augmentation* to achieve an incredibly realistic reconstruction. In this case, the system requires diffuse lighting and uses a set of eight calibrated and synchronized DSLR cameras. Another approach is the one proposed by Choi et al. [22], which uses a set of landmarks detected in the face of the subject in just one image to transform a generic 3D model. The results are not so realistic, but if more than one image is used, they get a lot better. As we will see later, having these landmarks in the face will give us the possibility of animating the face with common facial expressions.

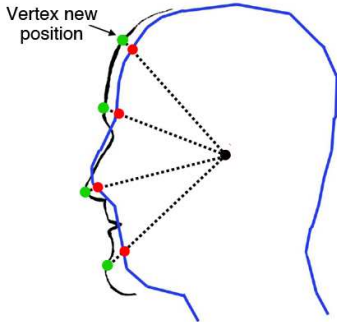


Figure 6: Vertex displacement. Every vertex is moved across the line which joins the vertex with the barycenter.

4.2. Active approach

4.2.1. Process summary

The first approach we propose uses the information provided by an active scanning device (such as a structured light projector, or a 3D laser scanner) to obtain a very detailed facial model, that we will call the *cloud mesh*, and use later to modify the geometry of the original VH, and thus produce a better quality complete human model. There are many different techniques for obtaining high quality models, such as the one proposed by Congote et al. [23], which uses a structured light projector and two cameras to create a high resolution point cloud to be meshed later, or the one proposed by Lee et al. [24], which uses a time-of-flight camera instead. Regardless of the type of acquisition system, it needs to be calibrated with respect to the rest of the cameras. Otherwise, an alignment system, such as ICP, would be necessary to properly align both meshes.

Although the cloud resulting from the scanning system is very accurate in the most important facial areas, there are sections where possible occlusions may yield big errors. As we will modify the original VH mesh with the information provided by the cloud mesh, it is necessary to “clean” it first. To do so, we use head proportions calculated using a face detection system based on the Viola-Jones face detection system [25]. Then, to match the polygonal resolution of the cloud mesh, the facial region of the VH mesh is subdivided using Dyn’s butterfly scheme [26]. Each vertex in the facial section is then displaced along the ray that joins it with the barycenter of the head, making its new position the one where we find an intersection between the ray and the cleaned cloud mesh (see Figure 6). Note that this kind of radial displacement preserves the

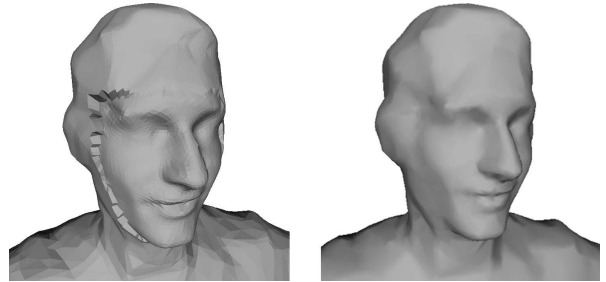


Figure 7: Step smoothing. The same model is presented before (left) and after (right) applying the process.

topological characteristics of the mesh.

As VH-based algorithms tend to slightly increase the volume of the model, a visually annoying *step* may appear between the displaced and fixed sections of the mesh, as illustrated by Figure 7. To avoid this step, we apply a smoothing algorithm to the transition or “bridge” area between the “shores” of both sections, and its width is defined using head dimensions. We call MOVED to the vertices in the side of the step that corresponds to the facial area, and STUCK to the ones located at the other side of the bridge area. This smoothing process is based on a customized interpolation of the position of both MOVED and STUCK vertices. A hierarchical tree structure is created where it is easy to find the shortest path along the vertices to both shores. This way, for each vertex v , we have two values, d_M and d_S , which correspond to the distances to the MOVED and STUCK shores respectively, calculated as:

$$d_k = \sum_{i=0}^n |v_i - v_{i+1}| \quad k \in \{M, S\}, \quad (2)$$

where n is the number of levels from v to the considered shore, and v_i and v_{i+1} are two successive vertices along the mentioned shortest path. The final value is again interpolated along the ray that joins the vertex with the barycenter, preserving again the topology of the mesh. This process can be found in greater detail in Pagés et al. [27].

4.2.2. Results

As shown in Figure 8, the results improve noticeably the quality of the final model without introducing any geometric error in the 3D mesh. Of course, the resulting models after applying the multi-texturing system look even more realistic (see Figure 9). An advantage of this method is that

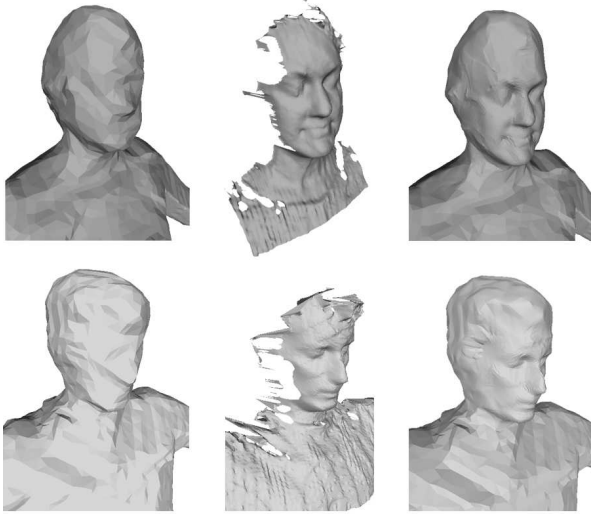


Figure 8: VH mesh (left), cloud mesh (middle) and the result of the process (right) without applying the multi-texturing process.

we can control the quality of the final models by controlling the number of subdivision steps, which helps create models for environments with different quality specifications.

4.3. Passive approach

4.3.1. Process summary

Our system also considers the possibility of creating a realistic human model without including any kind of expensive projector or 3D scanning device. This passive approach uses a neutral head mesh designed by an artist, which is transformed using the facial features of the subject to refine the low resolution facial area in the VH mesh. Figure 10 shows a graphical summary of the process.

The first stage in the process is determining a set of *Control Points (CPs)* both in the face of the subject (in two of the provided input images, which are triangulated using Hartley’s approach [28]) and in the neutral model. These CPs represent important features in the face of subject, such as the tip of the nose, or the corners of the eyes and lips, and can be few or many depending on how accurate the reconstruction must be.

The transformation is performed thanks to geometrical structures we call *Transformation Pyramids (TPs)*, all of which share the same apex, the barycenter of the head, but each of which has a different triangular basis defined by a combination of three CPs. Every vertex of the neutral model inside



Figure 9: Different multi-textured models before (left) and after (right) applying the active facial reconstruction process.

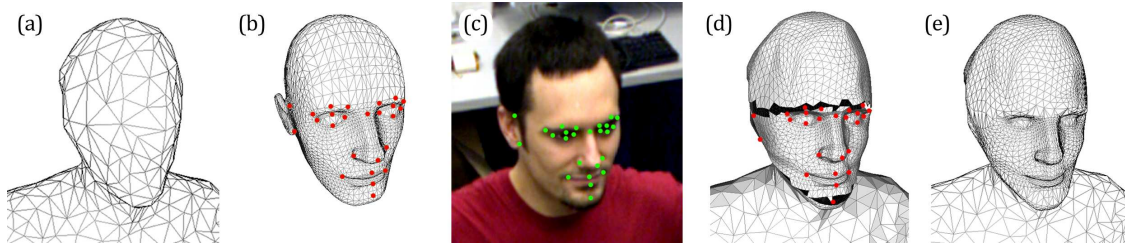


Figure 10: Graphical summary of the passive approach: (a) Original low resolution mesh, (b) Neutral head mesh, with control points highlighted, (c) One of the photographs of the subject, with control points manually selected, (d) Resulting mesh after transformation, (e) Final mesh after sewing the holes.

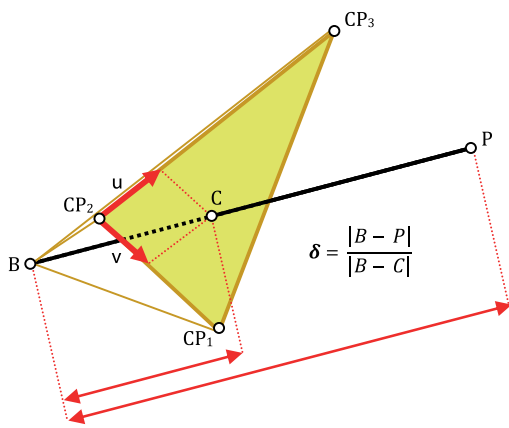


Figure 11: Parameters u and v are the magnitudes of the vectors that represent the projection of point C over two of the sides of the triangle. Point C is the intersection of the line which joins the barycenter (B) with the vertex which will be transformed. δ is the relation between distances to C and P from the barycenter.

the infinite volume defined by these TPs is transformed. The key of this system resides in an auxiliary coordinate system using three parameters, u , v and δ , which are defined by the TPs, as shown in Figure 11. When the values of the CPs in the neutral model are replaced with the ones of the subject, it is possible to apply the inverse transformation to every vertex. The result of this process is the neutral model modified according to the facial features of the subject.

As every vertex outside the transformation area defined by the TPs is removed, a visible hole may appear between both sections. Because of this, the last stage of the process consists in a hole filling process inspired in Liepa’s algorithm [29], but including no additional vertices (however, we do add a normal correction pre-process to align both sections of the mesh). A more detailed explanation of

the whole process can be found in Pagés et al. [30]

4.3.2. Results

As Figure 12 shows, the results of our algorithm are more than satisfactory. In this case, although the same neutral 3D model has been as a starting point, the different reconstructed models look very realistic, and even more so after the multi-texturing process has taken place. This algorithm also gives us another important advantage: as we know the position of a set of important points in the face of the subject, we could use them to animate it with different facial expressions.

As part of another research thread of ours [31], we are currently exploring ways to quantify the influence that each of the proposed steps has in the perceived quality of the final model. This is a challenging task due to the difficulty of obtaining ground truth for our datasets and the complex interactions that exist between geometry and texture, once rendered and perceived by humans.

5. Conclusion

We have presented a simple yet fully automatic system for the automatic reconstruction of a rigged avatar of any real person. After having built an initial version of our system by putting together state-of-the-art methods in order to generate a 3D model of a person photographed by several standard cameras, and having realized their limitations, we have developed novel techniques for multi-texture mapping and facial reconstruction to solve the two main problems detected.

On one hand, since naïve texture mapping techniques produce many artifacts, we propose to create a texture atlas by unwrapping the 3D mesh to form a set of 2D patches with no distortion, and then

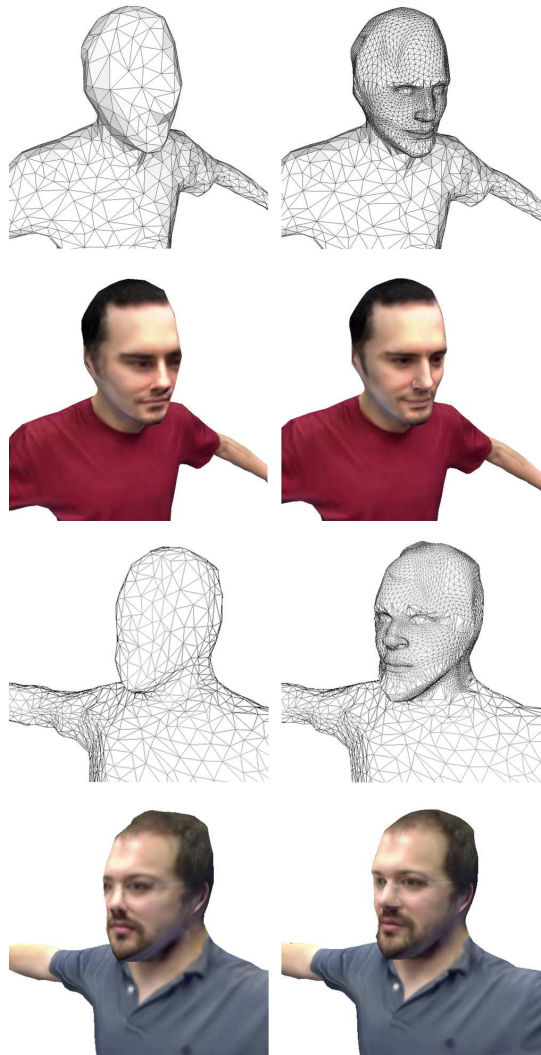


Figure 12: Results before (left column) and after (right column) the passive facial refinement process, both without (rows 1 and 3) and with (rows 2 and 4) texture.

mixing the color information from the images provided by the cameras. On the other hand, since the VH technique cannot handle concavities and therefore yields an excessively coarse shape approximation of the avatar's facial region (which is essential for subjective quality), we propose two techniques to enhance it: an active one based on a structured light projector, and a more affordable, passive one which tailors a generic humanoid head designed by an artist to fit the facial features of the specific person captured by the cameras. While the proposed multi-texturing technique is independent from the geometry-enhancing techniques, textures are crucial for perceived quality. Therefore, geometry enhancements should always be used in conjunction with a good texturing scheme.

Our patent-pending (ES-P201230768) system cannot outperform expensive state-of-the-art 3D scanning systems using very specialized hardware, but is simple and affordable, and yields results that are perfectly adequate for the intended purpose of automatically generating animatable avatars for use in video games.

Acknowledgements

We would like to thank the people at Telefónica I+D Barcelona for their invaluable help in this project. This work was supported in part by the Ministerio de Economía y Competitividad of the Spanish Government under project TEC2010-20412 (Enhanced 3DTV) and by the Spanish Administration Agency CDTI under project 2007-1007 (CENIT-VISION).

References

- [1] J. Gallego, M. Pardo, G. Haro, Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model, in: 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 3205–3208.
- [2] C. Cuevas, D. Berjón, F. Morán, N. García, Moving object detection for real-time augmented reality applications in a GPGPU, *IEEE Transactions on Consumer Electronics* 58 (1) (2012) 117–125.
- [3] J. I. Ronda, A. Valdés, G. Gallego, Line geometry and camera autocalibration, *Journal of Mathematical Imaging and Vision* 32 (2) (2008) 193–214.
- [4] M. Potmesil, Generating octree models of 3D objects from their silhouettes in a sequence of images, *Computer Vision, Graphics, and Image Processing* 40 (1) (1987) 1–29.
- [5] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *ACM SIGGRAPH Computer Graphics* 21 (4) (1987) 163–169.

- [6] J. Vollmer, R. Mencl, H. Müller, Improved Laplacian Smoothing of Noisy Surface Meshes, *Computer Graphics Forum* 18 (3) (1999) 131–138.
- [7] M. Garland, P. S. Heckbert, Surface simplification using quadric error metrics, in: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, ACM Press, New York, New York, USA, 1997, pp. 209–216.
- [8] P. E. Debevec, C. J. Taylor, J. Malik, Modeling and rendering architecture from photographs, in: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, ACM Press, New York, New York, USA, 1996, pp. 11–20.
- [9] I. Baran, J. Popović, Automatic rigging and animation of 3D characters, *ACM Transactions on Graphics (TOG)* 26 (3) (2007) 72.
- [10] F. Hétroy, C. Gérot, L. Lu, B. Thibert, Simple flexible skinning based on manifold modeling, in: *Proceedings of the 4th International Conference on Computer Graphics Theory and Applications (GRAPP)*, INSTICC Press, 2009, pp. 259–265.
- [11] R. Arnaud, M. C. Barnes, *COLLADA: Sailing the Gulf of 3D Digital Content Creation*, AK Peters, Wellesley, MA, USA, 2006.
- [12] V. Lempitsky, D. Ivanov, Seamless Mosaicing of Image-Based Texture Maps, in: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–6.
- [13] C. Rocchini, P. Cignoni, C. Montani, R. Scopigno, Multiple textures stitching and blending on 3D objects, in: *10th Eurographics Workshop on Rendering*, Springer-Verlag, Berlin/Heidelberg, 1999, pp. 119–130.
- [14] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, D. Cohen-Or, Seamless Montage for Texturing Models, *Computer Graphics Forum* 29 (2) (2010) 479–486.
- [15] R. Pagés, S. Arnaldo, F. Morán, D. Berjón, Composition of Texture Atlases for 3D Mesh Multi-texturing, in: *Eurographics Italian Chapter Conference 2010*, 2010, pp. 123–128.
- [16] A. Telea, An Image Inpainting Technique Based on the Fast Marching Method, *Journal of Graphics Tools* 9 (1) (2004) 23–34.
- [17] B. Lévy, S. Petitjean, N. Ray, J. Maillot, Least squares conformal maps for automatic texture atlas generation, *ACM Transactions on Graphics* 21 (3) (2002) 362–371.
- [18] G. Wolberg, *Digital Image Warping*, Wiley-IEEE Computer Society Press, 1990.
- [19] P. Fechteler, P. Eisert, J. Rurainsky, Fast and High Resolution 3D Face Scanning, in: *2007 IEEE International Conference on Image Processing*, IEEE, 2007, pp. III–81—III–84.
- [20] M. Fanwen, W. Lushen, L. Liping, 3D point clouds processing and precise surface reconstruction of the face, in: *2010 International Conference on Image Analysis and Signal Processing*, IEEE, 2010, pp. 104–107.
- [21] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, M. Gross, High-quality single-shot capture of facial geometry, *ACM Transactions on Graphics* 29 (4) (2010) 1.
- [22] J. Choi, G. Medioni, Y. Lin, L. Silva, O. Regina, M. Pamplona, T. C. Faltemier, 3D Face Reconstruction Using a Single or Multiple Views, in: *2010 20th International Conference on Pattern Recognition*, IEEE, 2010, pp. 3959–3962.
- [23] J. Congote, I. n. Barandiarán, J. Barandiarán, M. Nieto, O. E. Ruiz, Face reconstruction with structured light, in: *Proceedings of the 6th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2011, pp. 149–155.
- [24] C. Lee, H. Song, B. Choi, Y.-S. Ho, 3D scene capturing using stereoscopic cameras and a time-of-flight camera, *IEEE Transactions on Consumer Electronics* 57 (3) (2011) 1370–1376.
- [25] P. Viola, M. J. Jones, Robust Real-Time Face Detection, *International Journal of Computer Vision* 57 (2) (2004) 137–154.
- [26] N. Dyn, D. Levine, J. A. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics* 9 (2) (1990) 160–169.
- [27] R. Pagés, F. Morán, 3D facial merging for virtual human reconstruction, in: *2012 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, IEEE, 2012, pp. 1–4.
- [28] R. Hartley, A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, Cambridge, UK, 2003.
- [29] P. Liepa, Filling holes in meshes, in: *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2003, pp. 200–205.
- [30] R. Pagés, S. Arnaldo, F. Morán, Face Lift Surgery for Reconstructed Virtual Humans, in: *2011 International Conference on Cyberworlds*, IEEE, 2011, pp. 249–253.
- [31] D. Berjón, F. Morán, S. Manjunatha, Objective and subjective evaluation of static 3d mesh compression, *Signal Processing: Image Communication* 28 (2) (2013) 181–195.