REAL-TIME AVATAR ANIMATION WITH DYNAMIC FACE TEXTURING

Philipp Fechteler, Wolfgang Paier, Anna Hilsmann, Peter Eisert

Fraunhofer HHI & Humboldt Universität zu Berlin, Germany

ABSTRACT

In this paper, we present a system to capture and animate a highly realistic avatar model of a user in real-time. The animated human model consists of a rigged 3D mesh and a texture map. The system is based on KinectV2 input which captures the skeleton of the current pose of the subject in order to animate the human shape model. An additional high-resolution RGB camera is used to capture the face for updating the texture map on each frame. With this combination of image based rendering with computer graphics we achieve photo-realistic animations in real-time. Additionally, this approach is well suited for networked scenarios, because of the low per frame amount of data to animate the model, which consists of motion capture parameters and a video frame. With experimental results, we demonstrate the high degree of realism of the presented approach.

Index Terms— real-time, avatar animation, capturing, texture

1. INTRODUCTION

Because of todays omnipresence of virtual and augmented reality applications, efficient techniques to capture and model human characters are required. Until now, the availability of the Kinect made a broad range of real-time applications possible.

In this work, we target a capturing approach to provide photo-realistic animations of captured humans in real-time with the additional possibility to adjust the pose, e.g. gaze correction. Our approach is based on an animateble human shape model which results always in plausible poses and reduces the jittering contained in the input parameters. For capturing the kinematic pose, we use the KinectV2 sensor so that the captured motion capture parameters simply need to be converted to our model representation. Additionally, to achieve a photo-realistic visualization of the captured user, we track the user's face and update frame-wise the texture map of the corresponding region with the RGB camera input.

1.1. Related Work

The topic of visualization captured humans as realistic as possible is highly researched.

The authors of [1, 2] describe a model-based approach to recover the users geometry, which is useful for tele-presence applications since a low dimensional parameter vector is sufficient to describe the captured content. In [3, 4, 5], the authors combine image-based rendering techniques with an articulated geometric model in order to achieve photo-realistic animations of human motions.

In [6], a similar technique is presented to dynamically adapt the texture map of an animated template model from a RGBD stream. They extract motion capture parameters directly from the RGBD stream by optimizing the pose of a kinematic template model with the Coherent Point Drift algorithm. Although promising results are presented, the runtime complexity is far beyond real-time capability.

Fig. 1. The human model animated with KinectV2 motion capture parameters and dynamically textured face region.

Capturing facial geometry is a well studied research topic: recent methods like [7, 8, 9, 10] provide highly realistic results but are typically not usable in realtime systems. On the other hand there exist methods like [11, 12] that can capture and transfer the facial expression of a user based on linear models which can be driven in realtime. However, these systems focus only on facial capture and usually require a powerful computer with a recent GPU. We, in contrast, focus on full body capture with a single RGBD setup. This means that our cameras cover a wide field of view to allow capturing a full-body performance of an adult human, e.g. moving around, dancing, etc. In this case, the captured facial area is rather small and does often not provide enough details to fit complex deformable models to the input. Therefore, we rely on a low complexity method as for example described in [13] to provide a robust and realtime capable avatar animation pipeline with texture based facial expression transfer.

2. PROPOSED SYSTEM

The main idea of our system is to combine geometry and texture based techniques to animate a personalized avatar. The user's performance is captured by an RGBD camera and transferred to the avatar in realtime. We rely on a skeleton based animation to transfer large scale deformations of the body, e.g. walking, jumping or moving the arms. However, facial expressions typically involve fine scale and complex deformations which are hard to model in geometry. Therefore we rely on an image based approach for the facial animation part.

Our camera setup consists of a commodity depth and RGB camera (fig. 2). The depth sensor provides convenient access to depth maps, skeleton tracking data and a set of facial features for auto-

matic initialization and tracking recovery, while the additional color camera allows for retrieving face-textures with a higher spatial resolution. The workflow of the proposed system (fig. 3) consists of three stages. In the initial idle state the system relies on the skeleton tracking data provided by the depth camera. Potential users are detected as they walk in front of the depth camera and do not move for a short time span (e.g. 1 second). Based on five facial features (eyes, nose, mouth corners) provided by the depth camera we detect if the user looks straight into the depth camera. In the second stage, the user is asked to turn his/her head left and right in order to create a textured 3D model of the face. Using this 3D model, we run a model-based face tracking to accurately estimate the user's head pose in the captured video stream in real time. Finally, a hybrid animation approach is employed to drive the avatar [6]: first, a skeleton based animation strategy is used to apply large scale deformations and movements to the avatar. Secondly, we use a texture based approach in order to transfer the user's facial expression. For each tracked RGB frame, a face texture is extracted to update the facial region of the original texture. We use Poisson blending [14] to seamlessly integrate the new facial expression in the avatar's texture. Finally, the animated avatar is rendered using the updated texture.

2.1. Preprocessing and Initialization

For tracking/texture extraction our system uses the pinhole camera model and assumes that all data streams are undistorted as well as synchronized. The avatar's geometric 3D shape model is generated based on the SCAPE dataset [15] and using the LBS/DLB skinning method described in [16]. In order to initially register the avatar to the personalized face tracking model, we use 5 facial features which are provided by the depth camera. The corresponding 3D locations on the avatar's mesh are labeled manually which has to be done only once for each avatar.

The initialization starts as soon as a user stands still in front of the system and looks straight into the camera. Position and speed are computed from the skeleton tracking data while the viewing direction is approximated as the normal of the plane fitted to the tracked facial features (eyes, and mouth corners). In order to create an appropriate tracking model, a cylindrical sampling topology (r=0.2m,h=0.34m) is used to merge several tracked facial depth and texture maps while the user turns his/her head left and right for approximately +/- 20 degree. The sampling cylinder is centered at the 3D position of the head joint, taken from the skeleton tracking data, with its axis pointing straight upwards, i.e. $[0, 1, 0]^T$. Finally the mean cylindrical depth and texture map are used as tracking model, which results in a smooth as well as more complete geometry/texture (fig. 4) and allows for better tracking results in case of large head rotations. In order to capture only the facial geometry, we ignore all depth values that correspond to 3D points located outside the sampling cylinder.

Fig. 2. Hardware setup.

3. FACIAL TRACKING AND ANIMATION

As we use a texture based approach for facial animation, it is necessary to update the avatar's facial texture region at each time step. Though tracking information for skeleton joints and facial features is provided by the depth camera at each frame, its accuracy and reliability is not sufficient for retrieving video textures. Tracking errors would directly result in an unnatural and shaky motion of the replaced facial area, yielding an unrealistic facial animation. Therefore, a model based tracking approach is used to accurately track the user's head pose in realtime.

3.1. Model-Based Tracking

Our tracking method follows an analysis-by-synthesis scheme: based on a textured 3D model of the tracked object and initial motion parameters an approximate image of the current scene is rendered. By iteratively minimizing the difference between the captured image I and its rendered version $\hat{I}_{\mathbf{r},\mathbf{t}}$, we find an optimal parameter set that accurately describes the current scene. Our image formation model has 6 degrees of freedom, namely rotation $\mathbf{r} = [r_x, r_y, r_z]$ and translation $\mathbf{t} = [t_x, t_y, t_z]$. The main advantage of this tracking model is its low computational complexity allowing for realtime tracking even on CPU alone and its robustness which is especially useful when the subject covers only a small portion of the color-image, e.g. a wide angle lens is used or the person is further away from the sensor. The face tracking system simultaneously optimizes texture and geometry similarity by minimizing the following cost function:

$$\mathcal{E}(\mathbf{r}, \mathbf{t}) = \mathcal{E}_{tex}(\mathbf{r}, \mathbf{t}) + \lambda \mathcal{E}_{geo}(\mathbf{r}, \mathbf{t}), \tag{1}$$

with λ being a scalar factor that weights the influence of the geometric error term $\mathcal{E}_{geo}$ against the texture error term.

The first part of the objective function corresponds to texture similarity [17]:

$$\mathcal{E}_{tex}(\mathbf{r}, \mathbf{t}) = \sum_{x \in I} \left(I(x) - \hat{I}_{\mathbf{r},\mathbf{t}}(x) \right)^2, \tag{2}$$

with x being a face pixel. To make use of all available data captured by the RGBD camera, a geometric data-term $\mathcal{E}_{geo}$ is added that simultaneously registers the tracking geometry to the captured depth map. A projection based search [18] is used to find corresponding depth pixel d_i for each vertex $\mathbf{v}_i$ of the tracking model. A depth-buffer-based visibility test is performed to ignore all currently occluded model vertices. Knowing depth and camera calibration, the corresponding target vertex $\mathbf{u}_i$ is computed for each $\mathbf{v}_i$. The geometric error minimizes the point-to-plane distance between the

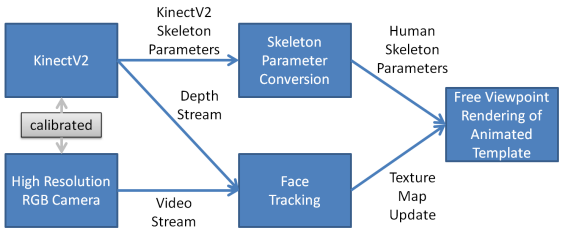

Fig. 3. System architecture.

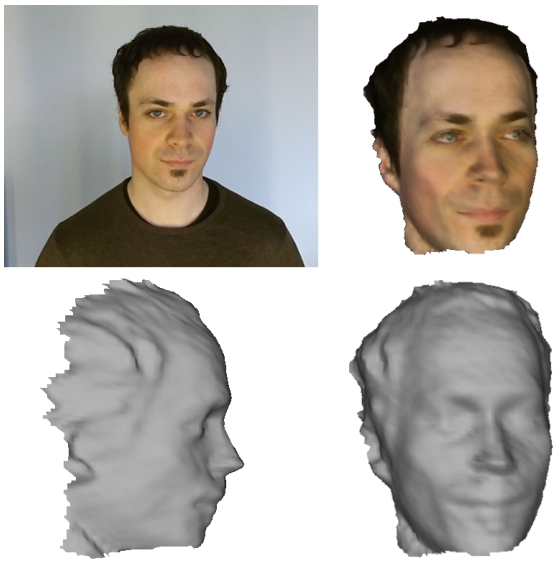

Fig. 4. Top: scanned subject (left) and final tracking model(right), bottom: tracking geometry without texture.

transformed model vertex $\mathcal{T}_{\mathbf{r},\mathbf{t}}(\mathbf{v}_i)$ and the tangent-plane of its corresponding target vertex $\mathbf{u}_i$ [19]:

$$\mathcal{E}_{geo}(\mathbf{r},\mathbf{t}) = \sum_i \left|(\mathcal{T}_{\mathbf{r},\mathbf{t}}(\mathbf{v}_i) - \mathbf{u}_i)\mathbf{n}_i\right|^2 \quad (3)$$

The surface normal $\mathbf{n}_i$ at vertex $\mathbf{u}_i$ is computed directly from the 4-neighborhood of d_i.

We use a non-linear optimization approach to iteratively minimize the cost function given in eq. 1. A fixed number of 7 Gauss-Newton iterations are run on each captured frame. In order to improve the tracking speed, $\mathcal{E}_{tex}(\mathbf{r},\mathbf{t})$ is evaluated on a scaled version of input image I, where the scaling factor depends on the distance between camera and user. If the user is closer than 2m, I is scaled down by a factor of 4 otherwise the input image is scaled down by 2.

3.2. Facial Animation

In order to transfer captured facial performances in realtime we exploit an image-based approach. The advantage of this approach is that facial expressions are reproduced faithfully with low computational overhead. In each time-step, we extract a texture from the current color image based on the tracking results. A predefined binary mask is used to transfer the facial texture region only. Finally, Poisson image editing [14] is used to integrate the current facial expression into the avatar's texture. We use an iterative parallelized solver to compute the new composite texture (fig. 5).

3.3. Tracking Recovery

Though the model based tracking works very robust even under large rotations and changing light situations caused by the user moving around and turning the head, the tracking may get lost, e.g. if the user walks out of sight, looks away from our system or moves very fast. To detect a tracking-loss, we observe several indicators: first, we measure the mean distance between the tracking model and the captured depth map. Second, the position of the head joint provides not an accurate but very robust estimation of the true head position. When our tracking result deviates too much from the position provided by the skeleton tracking, we also assume a tracking-loss. Finally, based on the 5 facial features (eyes, nose, mouth corners) we estimate a reference solution for the current tracking, i.e. rigid transform between the initial head pose and the current head pose, using Procrustes analysis [20]. If one of these indicators consistently reports a tracking-loss for more than 10 frames in a row, the tracking is reset to the last valid solution of the facial-feature-based tracking result. In order to test the reliability of the feature-based reference solution, we compute a rigid motion to transform the initial set of facial features to its current 3D positions. If the maximum distance between two corresponding feature locations exceeds a certain threshold we assume that the feature detection failed, e.g. due to occlusion, viewing angle, distance, etc., and facial features are not used to detect tracking-loss.

4. GEOMETRIC ANIMATION

For the animation of the geometric shape model we use the LBS/DLB skinning function [16] on a human model generated from the SCAPE dataset [15]. The resulting control structure is a typical skeleton with joints to animate the model. Since the skeleton of the chosen avatar differs from the one used by KinectV2, we have to convert the KinectV2 joint parameters in order to apply them properly to our model (fig. 6).

The global orientation of the user is calculated robustly from the KinectV2 input using the PCA of the KinectV2 joint locations of the waist, torso, and all shoulder and hip joints. Similarly, we determine the rotation of the breast and waist joint. The rotations of the remaining joints are calculated from the vectors pointing to the previous resp. next joint in the kinematic chain using the vector product for determining the rotation axis and the scalar product to determine the rotation angle. Since the facial region is tracked with high accuracy based on color information as described in sec. 3, we rely on the estimated face rotation for the head joint. The global translation to move our model onto the KinectV2 scan is calculated by subtracting the offset vector from the mean vertex location of the stomach from the kinematically animated template to the mean of the waist and torso joints of the KinectV2 parameters.

Since the KinectV2 joint parameter stream suffers from jitter, we apply a temporal filtering by averaging over the last n frames, where n is joint specific. In cases where some joints are not visible to the KinectV2, we keep the previously calculated joint rotation.

5. EXPERIMENTAL RESULTS

Our system setup consists of a KinectV2 and a Basler acA2040-25gc RGB camera with a resolution of 2048x2048. The KinectV2 and

Fig. 5. Left: input image, middle: animated avatar with original texture, right: animated avatar with updated face texture.

Basler sensors are driven with 30fps, therefore we reduced the resolution of the Basler camera to 2032x1600. The texture map we use is of size 1024x1024 and the facial region which is updated on every frame is of size 350x400.

The presented system has been tested thoroughly with different subjects (fig. 7). The user steps in front of the system which automatically triggers the initialization phase where an internal model of the head is generated for the face tracking unit. Then, the system automatically switches into animation mode, where the model is rendered in the pose captured with the KinectV2 while the facial region of the texture map is constantly updated with the RGB camera input.

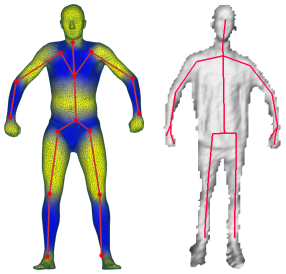

Fig. 6. The template model with its embedded skeletal control structure (left) and the KinectV2 output (right).

Fig. 7. System tested with different subjects. Left: fully animated avatar, right: corresponding input image.

In fig. 1, typical renderings of the animated full-body model are shown. A close-up view of the facial region of the rendered animated model is shown in fig. 8. Fig. 7, shows the rendering results for different subjects.

In fig. 8, we compare results where we generated the texture with the internal RGB camera of the KinectV2 as well as with a high-resolution RGB camera. On the one side, these results demonstrate the increase of the level of details of the facial region when using an additional high-resolution camera. On the other side, these results demonstrate the seamless embedding of the captured face into the static full-body texture map.

6. CONCLUSION

In this paper, we presented a model-based approach to capture and render a human in real-time and photo-realistic quality by combining image based rendering and computer graphics techniques. A KinectV2 is used to capture the pose of the subject in order to animate a rigged 3D shape model, while a high-resolution RGB camera captures the subjects facial area in order to update the texture map. This algorithmic combination provides photo-realistic animations in real-time. With experimental results, we evaluate the high level of visual quality achieved with this approach.

7. ACKNOWLEDGMENTS

The research that lead to this paper was supported in part by the European Commission under contract FP7-ICT-611761 ActionTV and H2020-ICT-687757 REPLICATE.

Fig. 8. Close up view of dynamically textured facial region captured with KinectV2-RGB-camera (top row), and with additional high resolution RGB camera (bottom row).

8. REFERENCES

[1] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H.-P. Seidel, "Seeing People in Different Light-Joint Shape, Motion, and Reflectance Capture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 663–674, July 2007.

[2] J. Carranza, C. Theobalt, M.A. Magnor, and H.-P. Seidel, "Free-viewpoint Video of Human Actors," in *ACM SIGGRAPH 2003 Papers*, New York, NY, USA, 2003, SIGGRAPH '03, pp. 569–577, ACM.

[3] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt, "Video-based characters: Creating new human performances from a multi-view video database," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 32:1–32:10, July 2011.

[4] A. Hilsmann, P. Fechteler, and P. Eisert, "Pose Space Image Based Rendering," *Comput. Graph. Forum (Proc. Eurographics)*, vol. 32, no. 2, pp. 265–274, May 2013.

[5] Peter Eisert and Jrgen Rurainsky, "Geometry-Assisted Image-based Rendering for Facial Analysis and Synthesis," *Signal Processing: Image Communication*, vol. 21, no. 6, pp. 493–505, July 2006.

[6] Philipp Fechteler, Wolfgang Paier, and Peter Eisert, "Articulated 3D Model Tracking with on-the-fly Texturing," in *Proc. Int. Conf. on Image Processing (ICIP)*, 2014.

[7] George Borshukov, Jefferson Montgomery, Witek Werner, Barry Ruff, James Lau, Paul Thuriot, Patrick Mooney, Stefan Van Niekerk, Dave Raposo, Jean-Luc Duprat, John Hable, Håkan Kihlström, Daniel Roizman, Kevin Noone, and Jeff O'Connell, "Playable universal capture," in *ACM SIGGRAPH 2006 Sketches*, New York, NY, USA, 2006, SIGGRAPH '06, ACM.

[8] Pablo Garrido, Levi Valgaert, Chenglei Wu, and Christian Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 158:1–158:10, Nov. 2013.

[9] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly, "Dynamic 3d avatar creation from hand-held video input," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 45:1–45:14, July 2015.

[10] M. Kettern, A. Hilsmann, and P. Eisert, "Temporally consistent wide baseline facial performance capture via image warping," in *Proceedings of the Vision, Modeling, and Visualization Workshop 2015*, Aachen, Germany, October 2015.

[11] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou, "3d shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 41:1–41:10, July 2013.

[12] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, 2015.

[13] Wolfgang Paier, Markus Kettern, and Peter Eisert, "Realistic retargeting of facial video," in *Proceedings of the 11th European Conference on Visual Media Production*, New York, NY, USA, 2014, CVMP '14, pp. 2:1–2:10, ACM.

[14] Patrick Pérez, Michel Gangnet, and Andrew Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, July 2003.

[15] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis, "SCAPE: Shape Completion and Animation of People," *Proc. ACM SIGGRAPH*, 2005.

[16] Philipp Fechteler, Anna Hilsmann, and Peter Eisert, "Example-based Body Model Optimization and Skinning," in *Proceedings of the 37th Annual Conference of the European Computer Graphics*, Lisbon,Portugal, 10th Mai 2016.

[17] P. Eisert and B. Girod, "Model-based 3d-motion estimation with illumination compensation," in *Image Processing and Its Applications*, 1997, vol. 1, pp. 194–198 vol.1.

[18] G. Blais and M.D. Levine, "Registering multiview range data to create 3d computer objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 8, pp. 820–824, Aug 1995.

[19] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, Apr 1991, pp. 2724–2729 vol.3.

[20] Peter Schnemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.