# FaceForge: Markerless Non-Rigid Face Multi-Projection Mapping

Christian Siegl, Vanessa Lange, Marc Stamminger, Frank Bauer and Justus Thies

Fig. 1. Results of our face projection system. All images in this work are captured with a DSLR camera.

**Abstract**— Recent publications and art performances demonstrate amazing results using projection mapping. To our knowledge, there exists no multi-projection system that can project onto non-rigid target geometries. This constrains the applicability and quality for live performances with multiple spectators. Given the cost and complexity of current systems, we present a low-cost easy-to-use markerless non-rigid face multi-projection system. It is based on a non-rigid, dense face tracker and a real-time multi-projection solver adapted to imprecise tracking, geometry and calibration. Using this novel system we produce compelling results with only consumer-grade hardware.

**Index Terms**—Face Projection, Mixed Reality, Multi-Projection Mapping, Non-Rigid Face Tracking.

## 1 Introduction

Projection mapping is a popular technique to alter the appearance of real-world objects. It is used for different applications in art, design, marketing and teaching. The target objects range from planar-like to arbitrary complex geometry. In this work we focus on augmenting deforming human faces as projection targets. This is especially desirable for art applications. For example, at the 2016 Grammy Awards, Lady Gaga integrated such a projection into her performance [2]. Another project from 2015 featured Kat von D in a Madrid live performance [28]. Both systems are built upon a marker-based face tracker, relying on a high quality scan of the person's face. However, the tracking only considers rigid head motions. In particular, they cannot handle dynamic facial expressions.

In this paper we introduce expression aware face projection. We combine a markerless non-rigid tracking system with correctly blended multi-projection. Furthermore, we reduce the hardware requirements to a minimum (consumer-grade projectors and RGB-D sensor) and demonstrate an easy-to-use system.

Our tracking algorithm is based on a parametric face model. Using a simple and fast initialization step, the system adjusts the model to match the face of the current user within seconds. At runtime, the model is dynamically adapted to the user's expression and pose in real-time. Since all art assets are designed for the unwrapped average face, they are transformed accordingly.

Projection mapping (most notably multi-projection mapping) relies on a very precise model of the target geometry. While the parametric face model simplifies asset generation, it is not a perfect representation of the actual face. Therefore, we adapt an existing projection mapping solver to compensate for the resulting errors.

To fulfill the real-time requirements of such a face projection system, all computationally intensive tasks are parallelized and computed on a modern graphics card.

## 2 Previous Work

Our project covers both face tracking and projection mapping.

### 2.1 Face Capture and Tracking

There exists a variety of markerless face reconstruction and tracking approaches. We will concentrate on real-time methods that employ a 3D parametric face model, like the morphable model of Blanz and Vetter [7]. A 3D parametric face model has the advantage, that it can be rendered from disparate views (projectors), and different faces share the same texture space.

Recently, methods were published that only rely on the input of a single RGB camera to reconstruct a parametric face model. They are using trained regressors [10, 9, 12] or a dense analysis-by-synthesis reconstruction schema [26]. Thies et al. [26] demonstrate how to virtually modify the appearance of a face and even allow to transfer the expression of one person to another. In contrast to these virtual changes, we tackle an even harder problem, we alter the appearance of a face in the real world using projection mapping. Since the above mentioned methods purely rely on the color of the face, they are not capable to track the face robustly under changes in the appearance of the face

- *All authors are with the Computer Graphics Group of the University of Erlangen/Nuremberg.*
- *Christian.Siegl@fau.de.*
- *Vanessa.Lange@fau.de.*
- *Marc.Stamminger@fau.de.*
- *Frank.Bauer@fau.de.*
- *Justus.Thies@fau.de.*

*Manuscript received 18 Sept. 2014; accepted 10 Jan. 2015. Date of Publication 20 Jan. 2015; date of current version 23 Mar. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.*

during projection mapping. We therefore rely on an RGB-D camera, using the unaffected depth channel during projection mapping. Beside the additional depth channel, state-of-the-art real-time RGB-D face tracking approaches [27, 8, 16, 14, 25] consider also the RGB channels of the camera, using sparse facial markers or dense color consistency terms. Using all available observation data, these methods try to get the optimal tracking results. However, they assume a constant skin appearance with a smooth illumination. This assumption does not hold in the projection mapping scenario, where high-frequency textures are projected onto the face. We therefore propose a method that is based on the approaches of Thies et al. [25, 26] to reconstruct a dense face model in a short initialization phase, considering both color and depth data of the camera. During projection mapping we then adapt the face geometry solely based on the unaffected depth channel.

### 2.2 Projection Mapping

There has been a wide body of work for using projection mapping in augmented and mixed reality applications. For an overview see Bimber et al. [6, 5].

Shader Lamps [21] showed compelling results for projection mapping onto arbitrary geometry, even using multiple projectors. However, a manual static calibration was employed, highly limiting the areas of application.

Mine et al. [18] describe various projection mapping applications for theme parks in static setups. While they do not get into too much algorithmic detail, the real-world challenges of projection mapping (calibration, blending, luminance correction, etc.) become apparent. We will show solutions to some of these problems in this work.

Sueishi et al. [24] presented a projection mapping system that is able to project onto moving objects with very low latency. While they showed very impressive results, they use highly specialized hardware (1000 fps camera and a projector behind a galvanometer mirror) and do not support multiple projectors.

Bermano et al. [4] alter the appearance of an animatronic head for richer details and expressions. They show convincing results and use multiple projectors, however they do not face the challenges of a really dynamic setup. Thanks to their animatronic projection target, they know the exact geometry for every actuator configuration and can therefore precompute the projector blending.

Asai et al. [2] achieve impressive results by projecting artistic animations onto a real human face using a marker-based face tracking (Lady Gaga, see Introduction). An interactive facial projection painting application was presented by Hieda et al. [13]. The big goggles however that are used for tracking break the immersion for the user.

Lincoln et al. [17] designed a teleconferencing system based on animatronic heads. The face of one person is transferred onto a styrofoam animatronic head at a different location. They recover the face geometry and texture similar to our approach, however rely on a head mounted tracking. The projection on the target geometry uses only a single projector and is based on a static setup.

In their recent work, Bermano et al. [3] show very impressive results in projecting onto a human face. Their tracking is non-rigid and has very low latency. However, they only use a single co-axial projector and highly custom and expensive hardware (480 Hz projector, 1300 Hz camera, framegrabber, etc.). In our work we achieve non-rigid multi-projection with only commodity hardware.

## 3 System Overview

The workflow of our face projection system is depicted in Figure 2. Figure 3 shows the actual setup. We utilize an RGB-D camera to reconstruct and non-rigidly track the user's face. Since we add color to the scene via projection mapping, the camera's color image is used only during an initialization phase. The tracking algorithm relies solely on the depth image (see Section 5).

Given the tracked geometry and the calibrated setup (see Section 4), the target color information is rendered from the viewpoints of the projectors. To compute the correct blending in the overlapping regions, we need geometric information. Additional buffers supply synthetic depth maps and normal information from the projectors' viewpoints. The multi-projection system then computes the correct luminance value for every projector ray (see Section 6). By applying these luminances to the target object colors, we obtain the final projected colors sent to the projectors.

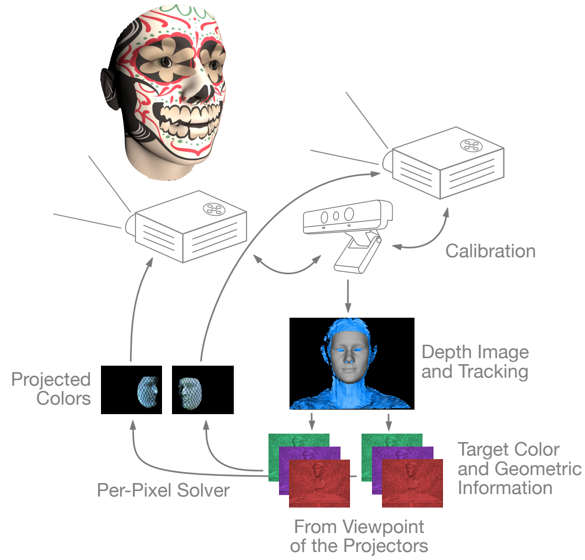


Fig. 2. The steps performed for each frame.

## 4 Calibration

To correctly render the reconstructed face from the viewpoints of the projectors, all components of the system need to be calibrated. The RGB-D camera offers an internal alignment of the color and depth images. Using this alignment the two cameras have the same intrinsic parameters which are provided by the camera vendor through an API function. Distortion can be neglected, since it is also handled by the camera API. This camera calibration is used in the initialization phase to reconstruct the face geometry.

For intrinsic projector calibration, we use the method presented by Moreno et al. [19]. A series of gray-code patterns is projected onto a checkerboard in multiple orientations and observed by a color camera. Using this information we reconstruct the intrinsic projector parameters.

In order to render synthetic views of the face, we need an additional extrinsic calibration step between the depth camera and the projectors. For this step we rely on a semi-automatic procedure. Using a crosshair pointer, the user identifies and selects predefined geometric landmarks on a rigidly tracked calibration target object. Given the intrinsic parameters of the projector, the correspondences and the 3D world positions of these landmarks, a non-linear system is solved using the Levenberg-Marquardt algorithm. To account for the different scales of rotation and translation, these parameters are computed in a flip-flop manner. For optimal accuracy, a final joint optimization step is added. Using these correspondences we recover the extrinsics of the projectors relative to the depth camera without the detour over using the color image.

The calibration is designed to be fast and easy to use. This is important in a real-world application, since the calibration quality decreases over time, especially from mechanical shifts of the projectors. An additional problem is the movement of the optics within the projectors due to material distortion from heat. Thus, we need a multi-projection system that is resilient to small calibration errors.

## 5 Face Reconstruction

A core component of our system is the reconstruction and non-rigid real-time tracking of a human face. To enable a robust reconstruction,

Fig. 3. Proposed face projection setup: Two projectors are directed towards the target person and project the desired content onto the face. The non-rigid face tracking is based on a commodity RGB-D camera.

we utilize a multi-linear face model as a prior (see Figure 4). Based on the statistical face model of Blanz and Vetter [7], we reconstruct the identity shape of a person. The model consists of 53490 vertices and 106466 triangles. We use the first 80 principle components for shape ($E_{\mathrm{id}} \in \mathbb{R}^{(3 \cdot 53490 \times 80)}$) and albedo ($E_{\mathrm{alb}} \in \mathbb{R}^{(3 \cdot 53490 \times 80)}$) of the statistical model. The facial expressions are modelled with 76 delta blend-shapes ($E_{\mathrm{exp}} \in \mathbb{R}^{(3 \cdot 53490 \times 76)}$) which are generated based on the data of Alexander et al. [1] and Cao et al. [11]. Using this basis, a face can be described with $2 \cdot 80$ identity (shape and albedo) and 76 expression parameters. The geometry of a face with shape parameters $\vec{\alpha}$ and expression parameters $\vec{\delta}$ is computed by:

$$\mathscr{G}(\vec{\alpha}, \vec{\delta}) = \vec{a}_{\mathrm{id}} + E_{\mathrm{id}} \cdot \vec{\alpha} + E_{\mathrm{exp}} \cdot \vec{\delta} \ . \tag{1}$$

$\vec{a}_{\mathrm{id}}$ is the average face geometry of the statistical model. The albedo of the face is synthesized using:

$$\mathscr{A}\left(\vec{\beta}\right) = \vec{a}_{\mathrm{alb}} + E_{\mathrm{alb}} \cdot \vec{\beta} \ . \tag{2}$$

Where $\vec{a}_{\mathrm{alb}}$ is the average face albedo of the statistical model. Additionally, we model the illumination using the first three bands of spherical harmonics as proposed by Ramamoorthi et al. [20], resulting in 27 illumination parameters $\vec{\gamma}$.

The described face model has the advantage, that every reconstructed face shares the same topology and also the same texture space. This allows artists to create an animated texture, that can be applied to arbitrary persons. Since the texture coordinates are attached to the vertices of the face model, the projected texture deforms with the model's surface (see Figure 5).

Our system includes a short initialization phase, where the shape and the texture of the face is estimated. The face reconstruction exploits the analysis-by-synthesis principle, thus, computing the model parameters such that the observations of the RGB-D camera are matched. During the initialization phase, the projectors do not emit light, enabling us to use RGB data beside depth information. Following the non-rigid bundling schema of Thies et al. [26], we reconstruct

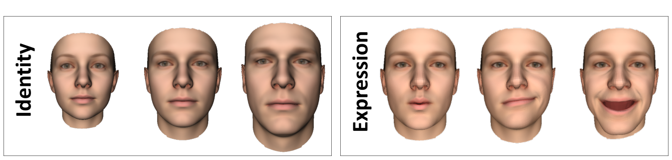



Fig. 4. Parametric Face Model: Our face model is a combination of a statistical model that represents the identity and blend-shapes that model the expressions.

(a) (b) (c)

Fig. 5. The texture from (a) is mapped to the tracked face model without (b) and with (c) an expression. Note, how the texture adapts to the deforming surface.

the shape in a few seconds. We choose a frontal and two lateral image frames for bundling. Each of these three frames captures a color $C_i$ and a depth image $D_i$ (reprojected into the camera space ($D_i(p) \in \mathbb{R}^3$)). The bundling energy term $E_{bundling}(\mathscr{P})$ is defined as following:

$$E_{bundling}(\mathscr{P}) = \sum_{i=1}^{3} E^i_{rgb}(\mathscr{P}^i) + E^i_d(\mathscr{P}^i) + w_{reg} E_{reg}(\mathscr{P}^i) \ . \tag{3}$$

$\mathscr{P}$ is the parameter vector containing all unknowns of the per frame unknowns $\mathscr{P}^i$ (consisting of shape, albedo, expression, illumination and rigid pose parameters). The energy function $E^i_{rgb}(\mathscr{P}^i)$ of the $i$-th frame measures the pixel-wise difference between the input image $C_i$ and the synthesized image $\hat{C}_i$. In addition, the *RGB* term measures the difference between detected facial landmarks $f_j$ and the corresponding vertices $v_j$ of the face mesh:

$$E^i_{rgb}(\mathscr{P}^i) = \frac{w_{col}}{|\mathscr{V}|} \sum_{p \in \mathscr{V}} \left|\hat{C}_i(p) - C_i(p)\right|_2 + w_{lan} \sum_{j=1}^{66} \left|f_j - \Pi(v_j)\right|_2^2 \ .$$

The synthesized face images $\hat{C}_i$ are generated using a full perspective projection $\Pi(\vec{x})$ in the standard graphics pipeline. We use the calibrated intrinsics of the RGB-D camera provided by the camera API. To compute the set of rendered pixel positions $\mathscr{V}$ we apply a GPU-based scan. The facial landmarks are tracked using a commercial implementation of Saraghi et al. [22]. The depth term of every single frame is defined as:

$$E^i_d(\mathscr{P}^i) = \frac{1}{|\mathscr{V}|} \sum_{p \in \mathscr{V}} w_{point} \left|\hat{D}_i(p) - D_i(p)\right|_2^2 + w_{plane} \left|(\hat{D}_i(p) - D_i(p))^T \cdot \hat{N}_i(p)\right|_2^2 \ .$$

$\hat{D}_i$ and $\hat{N}_i$ are the generated depth and surface normal images of the synthesized model. To prevent degeneration of the face reconstructions, we regularize the reconstruction by additionally measuring the distance of the shape to the average face and the distance of the expression to the neutral pose in parameter space using:

$$E_{reg}(\mathscr{P}^i) = \sum_{i=1}^{80} \left[\left(\frac{\vec{\alpha}_i}{\sigma_{id,i}}\right)^2 + \left(\frac{\vec{\beta}_i}{\sigma_{alb,i}}\right)^2\right] + \sum_{i=1}^{76} \left(\frac{\vec{\delta}_i}{\sigma_{exp,i}}\right)^2 \ .$$

$\sigma_{id}$ and $\sigma_{alb}$ are the standard deviations of the statistical face model, $\sigma_{exp}$ is the deviation of the expressions, set to a constant value ($= 1$). The bundling energy term $E_{bundling}(\mathscr{P})$ is minimized using the GPU-based Gauss-Newton framework proposed by Thies et al. [25]. The weights of the energy formulation are set to $w_{col} = 100.0$, $w_{point} = 2000.0$, $w_{plane} = 10000.0$, $w_{lan} = 100.0$ and $w_{reg} = 0.00125$.

Figure 6 depicts an example reconstruction of the RGB-D bundling schema. We compare our reconstruction against a high-quality structured light scan, yielding a Hausdorff Distance of only 2.39 *mm* with an RMS of 4.28 *mm*.

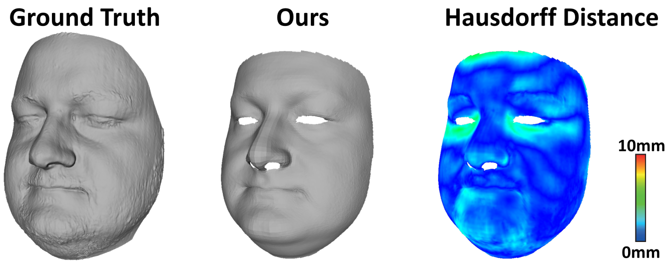



Fig. 6. Reconstructed geometry (middle) using the proposed RGB-D bundling schema in comparison to a high-quality (left) structured light scan (David SLS-3).

During projection mapping, we track the face frame-by-frame, considering changes in expression and rigid pose. Since the projection mapping alters the color of the face, we rely only on the depth data of the camera. Thus, the tracking energy term is defined as:

$$E_{tracking}(\hat{\mathscr{P}}) = E_d(\hat{\mathscr{P}}) + w_{reg} E_{reg}(\hat{\mathscr{P}}) \ . \quad (4)$$

Using the estimated parameters of the previous frame as an initial guess, we minimize the energy using 7 Gauss-Newton iterations.

Since we only rely on noisy depth data, the tracking needs to be filtered. We add parameter smoothing using a history of 4 unfiltered parameter sets with an exponential weighting.

## 6 Projection Mapping

Projection mapping with a single projector is a rather easy task. However, using only a single projector highly limits the quality of the projection mapping. Especially for target objects that exhibit more complex geometry (like a human face), multiple projectors are required to prevent self shadowing. Similarly, movements (from either user or observer) benefit from a multi-projector setup to cover the full surface. However, blending multiple projectors in a dynamic setup is a challenging problem.

We show the first dynamic multi-projection mapping system that handles a non-rigid target object.

### 6.1 Multi-Projection Mapping

Our multi-projection mapping system is based on the work introduced by Siegl et al. [23]. Their system optimizes a luminance for every projector ray to correctly blend between projectors in overlapping regions. This results in a uniform illumination on the surface. To compute the correct luminance per ray, they propose an optimization problem that consists of the following terms:

- Physical Term: represents the physical attenuation of light (distance and Lambert's Law)
- Balance Term: gives more contribution to the projector with the best possible projection quality
- Laplacian Term: smooths and increases numerical stability
- Bounding Term: keeps luminances in the possible 0 to 1 range

With their real-time solver on the GPU, dynamic multi-projection mapping is solved by estimating the best possible projection on the target geometry. Using a markerless rigid tracking algorithm allows them to move the object. While they show impressive results, this system is highly dependent on a precise 3D reconstruction of the target object and a close to perfect calibration. Since these are hard to achieve in real-world applications, the authors have extended their system to blend more smoothly between projectors. Thereby they become more resilient to calibration errors [15].

Projecting on non-rigid real-world faces poses a new challenge: The face reconstruction based on a statistical model is much less accurate than the very high quality static 3D scans used before. Using a single

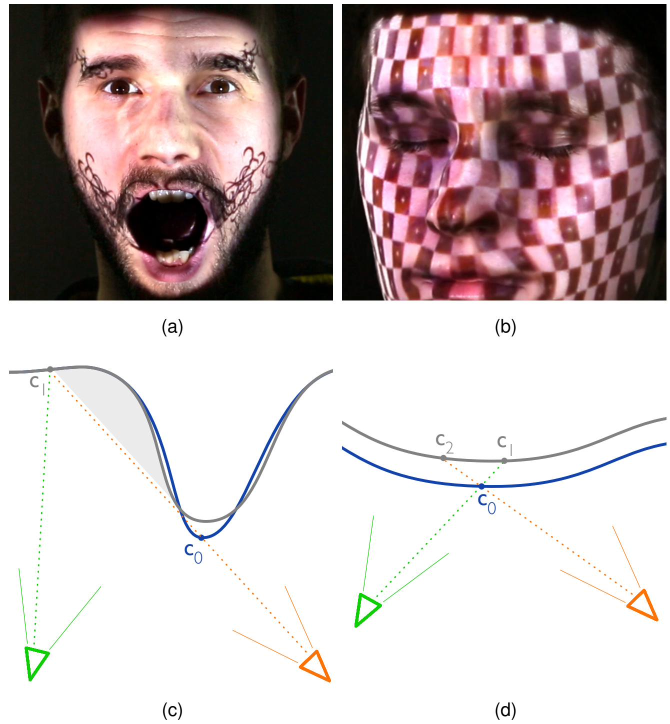



Fig. 7. The effects of a slightly inaccurate model, calibration, or tracking on multi-projection mapping (c, d). The actual model is depicted in gray, the tracked geometry in blue. Notice the bright artifact below the eye in (a) and the blurry projection in (b).

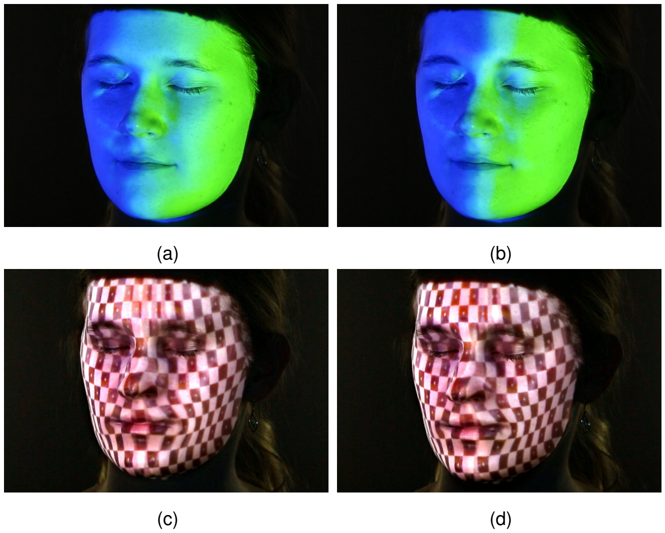
projector setup, an imperfect calibration and model results in a shift of the projection on the target object. This may cause the projection to miss the geometric feature it is supposed to illuminate. In general though, for smaller errors this effect is largely invisible to the viewer. However, in case of multiple projectors, blending is computed based on the tracked 3D model by taking the interaction between rays into account. Since our reconstructed 3D face model does not accurately match the real-world face (compare Figure 6), different rays than anticipated may interact.

Figure 7 depicts the resulting problems. The gray line in (c) and (d) represents the real world object, the blue line the tracking geometry. In case of Figure 7 (c), the orange projector ray is assumed to hit the surface-point $c_0$. As a result, $c_1$ is supposed to be illuminated by only the green projector. In reality, both projectors illuminate $c_1$, causing a visible artifact in (a) below the eye due to the added brightness (*brightness seam*). A second scenario causing artifacts under these circumstances is depicted in Figure 7 (d). The green and orange projector rays are expected to hit the target surface at $c_0$. Assume, that we want to project a high frequency texture detail at this point. Since the real-world (gray) geometry is located further in the back, the projection aiming for $c_0$ instead hits $c_1$ and $c_2$. As a result, the luminance computation is wrong and therefore the high frequency detail will be projected at two surface points ($c_1$ and $c_2$), causing a ghosting artifact (see Figure 7 (b)). These two kinds of artifacts catch the viewer's attention immediately and lower the perceived projection mapping quality.

### 6.2 Zebra Technique – Projector Blending

The scenario shown in Figure 7 (c) results in the *brightness seams* visible in (a). A partial solution to this problem was described by Lange et al. [15], where they introduce a new balance term $E_{\text{bal}}$. If multiple projectors illuminate the same surface spot, a projector's contribution (the luminance computed by the previously described solver) should be equal to the corresponding ray quality:

$$\frac{p_i}{\sum_{r \in R_i} p_r} \stackrel{!}{=} \frac{s_i}{\sum_{r \in R_i} s_r}$$



(a) (b) (c) (d)

Fig. 8. Zebra Technique: (a) and (b) show the color-coded contribution of two projectors without (a) and with (b) the *zebra technique*. (c) and (d) show the corresponding projections. Note, how the projection on the forehead in (d) is substantially sharper.

$p_i$ is the contribution of projector pixel $i$ (this index runs across all pixels of all projectors) and $R_i$ is the set of reprojections. $s_i = \langle \vec{n}_i, \vec{i}_i \rangle$ describes the heuristic for measuring the projection quality of a pixel. $\vec{n}_i$ is the normalized surface normal and $\vec{i}_i$ the normalized negative light direction.

The authors further added *blend weights* $w_i$ to steer the contribution of projectors:

$$E_{\text{bal}} = \sum_{i=1}^{N} \left( p_i \cdot \sum_{r \in R_i} s_r w_r - s_i w_i \cdot \sum_{r \in R_i} p_r \right) \stackrel{!}{=} 0 \tag{5}$$

These *blend weights* are used to unilateraly dim down a projector that will lead to poor projection quality at a given surface point. This is especially the case around depth discontinuities and the frustum borders from the viewpoint of the projectors. Therefore, the *blend weights* are set to zero at these points. By dilating and smoothing the resulting *blend weight maps* (*blend weights* per projector on a per-pixel level), a smooth blending between projectors is ensured. The *blend weights* serve as one input into the luminance solver, more precisely the balance term. The solver then ensures correct illumination at every surface point. For a more detailed description of *blend weights*, we refer the reader to Lange et al. [15].

Analogous to these *blend weights* we add an additional algorithmic step to the pipeline in order to remedy the blurry projections that can be seen in Figure 7 (b) and Figure 8 (c). As described before, ghosting and blurry artifacts do not occur when using only a single projector. Also, slight mismatches between projection and actual geometry are imperceivable. To this end, it is highly beneficial to illuminate every given surface point only with a single projector. Following the concept of *blend weights* ($w_i$), we introduce *zebra weights* ($z_i$). Every projector ray illuminating a surface point is compared to all other rays hitting the same point. Only the projector ray that will lead to the highest possible projection quality is assigned a *zebra weight* of 1, all others 0. For prediction of the projection quality we use the heuristic presented by Siegl et al. [23] (incident angle and distance of projector ray). This ensures consistent quality assessment, producing coherent patches per projector.

These new *zebra weights* are added to the existing *blend weight map* (before the dilation and smoothing) by choosing the minimum between the two. This means, that if in doubt, a projector is rather dimmed down. The combined map is then – as in the original paper – smoothed to get continuous transitions between 0 and 1 areas. By modifying the original *blend weight map*, the *zebra weights* require no further algorithmic steps and the error term is only slightly updated with the modified *blend weights* $w_i^*$. The computed error $E_{\text{bal}}$ is – as before – given to the luminance solver, guiding it towards the best projection quality.

With this additional step, each surface patch on the target face is illuminated by as few projectors as possible (based on the projector setup and the actual geometry). Figure 8 shows the effect of the *zebra weights*. In (a) and (b), the projector contributions are color-coded. Note how the blending between projectors in (b) displays a rapid falloff. Therefore, a much smaller area is illuminated by both projectors. The resulting projection for a high frequency texture is shown in (c) and (d). As can be seen in (d), the blending area of the projectors is reduced to a small region, resulting in an overall increased sharpness compared to the untreated projection shown in (c). This is especially visible on the forehead.

We call this the *zebra technique*, as the projectors' contribution form a zebra pattern on the target geometry (in case of more than two projectors).

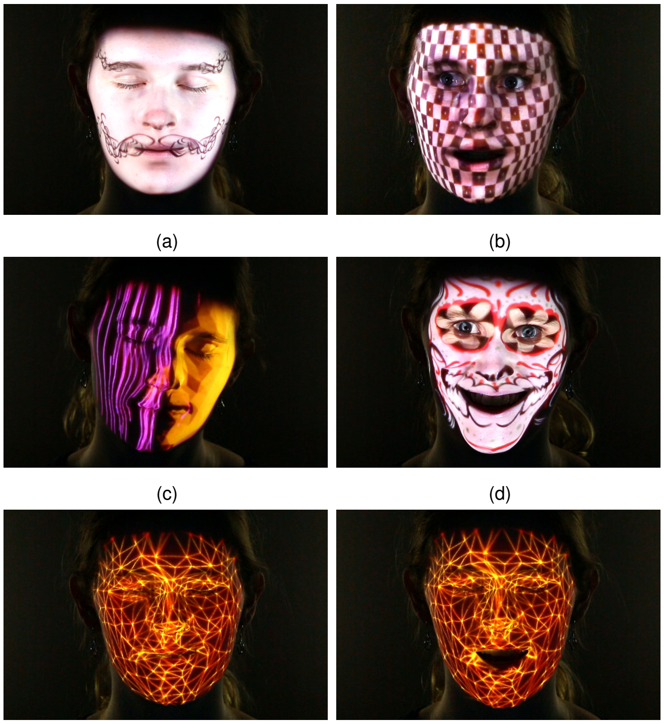
Fig. 9. Different people showing various facial expressions with the same projected texture.

## 7 Results

This section shows the results of our novel face multi-projection system on a variety of faces with different expressions, poses and target textures. We highly encourage the reader to watch the supplementary video for seeing our system in live action.

### 7.1 Hardware

Our face projection setup (see Figure 3) consists of two `NEC NP-P451WG` projectors with a resolution of 1280 by 800 pixels. An `Intel RealSense SR300` camera provides color and depth information. These components are connected to a standard desktop workstation with an `Intel Core i7 4771 (3.5 GHz)` CPU,



(a) (b) (c) (d) (e) (f)

Fig. 10. One person showing different textures. Note, how the texture is correctly deformed by the facial expression in (e) and (f).

32 GB of RAM and an `NVidia GeForce GTX 1080` graphics card. For the intrinsic projector calibration (see Section 4) we observe the calibration pattern with a `Canon 5D Mark III` DSLR camera. This camera also captured all images and videos in this work.

### 7.2 Performance

Using the above mentioned hardware the system runs in less than 20 ms per frame, thus being real-time. Tracking the user's face takes an average of 10.4 ms. Rendering the scene for the two projectors adds 2.7 ms and solving the luminances another 6.2 ms.

### 7.3 Qualitative Results

Figure 9 shows multiple persons and facial expressions using the same target texture. We want to stress that our initialization phase for each person is only a few seconds enabling us to quickly adapt to any face. As can be seen, the reconstructed face model fits the real person's face. Also the expression are precisely captured. The applied flat shading texture is deformed according to the real-world surface. The projection appears to be glued to the face, thereby altering the user's appearance convincingly. Beside the reflection, neither the tracking nor the projection is disturbed by glasses.

In Figure 10 various textures are projected onto a single person. All shown textures are animation sequences, best seen in the video. The result in (a) shows the quality of our proposed blending technique, in the extreme case of a nearly uniform white texture. Figures (b, d, f) demonstrate how our facial tracking system deforms the different textures to fit the person's expression. Straight lines bend and follow the natural curvature of the face (c). The wireframe texture in (e, f) highlights the dynamic face reconstruction.

Given a texture that leaves the eye region dark (b, d) allows the person to open the eyes. For brighter textures (e.g. in (a)), the two utilized 4500 lumen projectors would blind the person.

## 8 Limitations and Discussion

Our system is robust against imprecise calibrations and models. Given the setup used throughout this work, a projection error of two pixels results in an offset of about 1 mm on the target object. From the other side, even small errors of the face reconstruction will lead to a misalignment of pixels. A typical error of the face reconstruction can be seen in Figure 6 with a Hausdorff Distance of 2.39 mm. In Figure 8, the mismatch between the projected checkerboard patterns shows an error of 2-3 mm on the surface. This misalignment varies over the surface, since it is dependent on both the calibration and reconstruction error. However, this is an extreme case that we chose for demonstrating the effectiveness of the presented *zebra technique*. In general, the error on the face surface lies within 1-2 mm, as can be seen in Figure 10 (b). The new *zebra technique* minimizes ghosting by substantially shrinking overlapping projection areas. Without this technique, the two projectors (blue and green) in Figure 8 (a) show significant overlap (cyan), resulting in intrusive ghosting (see Figure 8 (c)). As can be seen in Figure 11 fast head motions (a) can lead to inaccurate tracking. (b) shows an expression that is not covered by the blendshape model. In both cases the assumed face geometry does not match the real-world face causing visible artifacts (misalignment, ghosting and stretching). Our system works best with non-occluded faces. Facial hair as well as hair overlapping from the sides may interfere with the face tracking.

(a) (b)

Fig. 11. Limitations of the system. Rapid head movement (a) or extreme facial expressions (b) cause a misalignment with the face.

We are limited to projecting onto the part of the face that is covered by the parametric face model. Though, by using a model that covers a larger part of the human body, a larger projection target area is possible.

Another limitation of our system we want to discuss is latency. As described above, our system runs well within the limits usually described as real-time. For projection mapping however, the frame-rate is much less important than latency, since this will break the immersion of the viewer. Our system is constructed with consumer-grade hardware, the RGB-D camera introduces latency due to the on-chip processing and the rather slow USB connection. Additional latency is introduced by the commodity projectors which are not geared for low latency. The overall latency becomes visible as a misalignment of the projection when the user moves very rapidly (see Figure 11 (a)). However, while not reaching the extremely low latency of the highly customized system presented by Bermano et al. [3], the overall latency of our commodity hardware setup is still manageable.

## 9 Conclusion

In this work we presented the first real-time fully dynamic face multi-projection system. We combined a markerless face capture and tracking system with a dynamic multi-projection system. By adjusting the dense face tracking algorithm to rely only on a depth map of the scene, the tracker is not disturbed by the projected color. In order to minimize visible projection artifacts, caused by the imperfect target geometry, we enhanced the existing multi-projection solver. Overlapping and thereby misaligned regions are reduced, since we use as few projectors as possible for every given surface point. The resulting system is the first of its kind, performing multi-projection mapping on non-rigidly deforming target geometry.

## Acknowledgments

We would like to thank Chen Cao and Kun Zhou for the blendshape models, as well as Volker Blanz, Thomas Vetter, and Oleg Alexander for the provided face data. The facial landmark tracker was kindly provided by TrueVisionSolution. This research is partially funded by the German Research Foundation (DFG), grant GRK-1773 Heterogeneous Image Systems.

## References

[1] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. The Digital Emily Project: photoreal facial modeling and animation. In *ACM SIGGRAPH Courses*, pages 12:1–12:15. ACM, 2009.

[2] N. Asai. http://www.nobumichiasai.com/post/139912267747/lady-gaga-david-bowie-nobumichi-asai58th, 2016.

[3] A. Bermano, M. Billeter, D. Iwai, and A. Grundhöfer. Makeup lamps: Live augmentation of human faces via projection. In *Eurographics*, 2017.

[4] A. Bermano, P. Brüschweiler, A. Grundhöfer, D. Iwai, B. Bickel, and M. Gross. Augmenting physical avatars using projector-based illumination. *ACM Trans. Graph.*, 32(6):189:1–189:10, Nov. 2013.

[5] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhöfer. The visual computing of projector-camera systems. In *Computer Graphics Forum*, volume 27. Wiley Online Library, 2008.

[6] O. Bimber and R. Raskar. *Spatial augmented reality: merging real and virtual worlds*. AK Peters Wellesley, MA, 2005.

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[8] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM TOG*, 32(4):40, 2013.

[9] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM TOG*, 34(4):46:1–46:9, 2015.

[10] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM TOG*, 33(4):43, 2014.

[11] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3D facial expression database for visual computing. *IEEE TVCG*, 20(3):413–425, 2014.

[12] C. Cao, H. Wu, Y. Weng, T. Shao, and K. Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.*, 35(4), July 2016.

[13] N. Hieda and J. R. Cooperstock. sharedface: Interactive facial projection mapping. In *Proceedings of the 2015 Virtual Reality International Conference*, VRIC '15, New York, NY, USA, 2015. ACM.

[14] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *Proc. CVPR*, 2015.

[15] V. Lange, C. Siegl, M. Colaianni, M. Stamminger, and F. Bauer. Robust blending and occlusion compensation in dynamic multi-projection mapping. In *Eurographics*, 2017.

[16] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM TOG*, 32(4):42, 2013.

[17] P. Lincoln, G. Welch, A. Nashel, A. State, A. Ilie, and H. Fuchs. Animatronic shader lamps avatars. *Virtual Reality*, 15(2):225–238, 2011.

[18] M. R. Mine, J. van Baar, A. Grundhöfer, D. Rose, and B. Yang. Projection-based augmented reality in disney theme parks. *IEEE Computer*, 45(7):32–40, 2012.

[19] D. Moreno and G. Taubin. Simple, accurate, and robust projector-camera calibration. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 464–471, Oct 2012.

[20] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 497–500, New York, NY, USA, 2001. ACM.

[21] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, London, 2001. Springer-Verlag.

[22] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011.

[23] C. Siegl, M. Colaianni, L. Thies, J. Thies, M. Zollhöfer, S. Izadi, M. Stamminger, and F. Bauer. Real-time pixel luminance optimization for dynamic multi-projection mapping. *ACM Trans. Graph.*, 34(6):237:1–237:11, Oct. 2015.

[24] T. Sueishi, H. Oku, and M. Ishikawa. Robust high-speed tracking against illumination changes for dynamic projection mapping. In *IEEE Virtual Reality Conference (VR2015)*, 2015.

[25] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183, 2015.

[26] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[27] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. In *ACM TOG*, volume 30, 2011.

[28] wildbytes. Kat von d's live face projection mapping. http://wildbytes.cc/work/kat-von-d-live-face-projection-mapping, 2015.