EUROGRAPHICS 2017 / L. Barthe and B. Benes
(Guest Editors)

Volume 36 (2017), Number 2

Makeup Lamps: Live Augmentation of Human Faces via Projection

Amit H. Bermano[1,2], Markus Billeter[3], Daisuke Iwai[4], and Anselm Grundhöfer[1†]

[1]Disney Research [2] Princeton University [3] Chalmers University [4] Osaka University

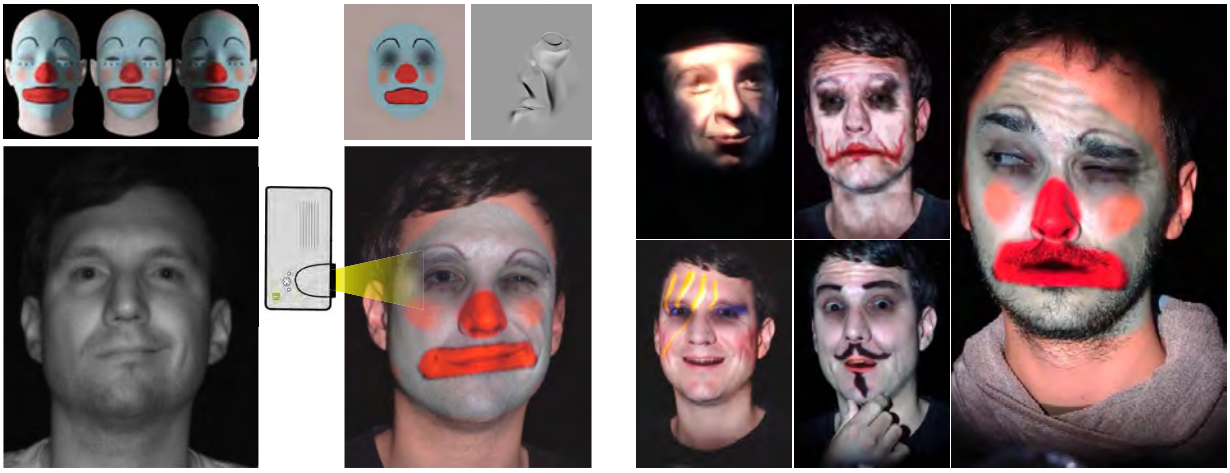

Figure 1: *Our system captures a live performance in the infrared (IR) spectrum (bottom left). A target appearance (top left) is rendered and saved as albedo and offsets based on expression and spatial position (top middle). These are blended and deformed to match the facial configuration and position (bottom middle), and projected back on the performer for appearance augmentation (right). The average system latency is* 9.8*ms, achieved through GPU optimizations, and is further compensated for through prediction.*

Abstract

We propose the first system for live dynamic augmentation of human faces. Using projector-based illumination, we alter the appearance of human performers during novel performances. The key challenge of live augmentation is latency — an image is generated according to a specific pose, but is displayed on a different facial configuration by the time it is projected. Therefore, our system aims at reducing latency during every step of the process, from capture, through processing, to projection. Using infrared illumination, an optically and computationally aligned high-speed camera detects facial orientation as well as expression. The estimated expression blendshapes are mapped onto a lower dimensional space, and the facial motion and non-rigid deformation are estimated, smoothed and predicted through adaptive Kalman filtering. Finally, the desired appearance is generated interpolating precomputed offset textures according to time, global position, and expression. We have evaluated our system through an optimized CPU and GPU prototype, and demonstrated successful low latency augmentation for different performers and performances with varying facial play and motion speed. In contrast to existing methods, the presented system is the first method which fully supports dynamic facial projection mapping without the requirement of any physical tracking markers and incorporates facial expressions.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [HCI]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Changing the appearance of objects and humans is observed throughout history, through painting, dyeing and coating. Specifically, face painting has been used for an extremely wide variety of purposes ranging from cultural ceremonies, through impersonation and camouflage to everyday make-up. Of course, traditionally, the appearance alteration is static, and requires manual labor to be changed. The digital era has given rise to new opportunities in this context, where one could observe how an

† e-mails:{bermano@princeton.edu; billeter@chalmers.se; daisuke.iwai@sys.es.osaka-u.ac.jp; anselm@disneyresearch.com}

object or a person would look like when painted or altered in different ways, both statically and dynamically. For physical objects, this can be done with the aid of projection-based illumination, bringing the flexibly of digital objects to physical appearance [RWF99, BRF01, BBG*13, AIS15, NWI15, ZXT*16]. Since introduced, these methods have been used extensively in education, training, entertainment and sales.

On the other hand, research on re-targeting or transferring facial expressions and performances from real actors to virtual characters has also made rapid strides in recent years [CXH03, WBLP11, BWP13, CWLZ13, CHZ14, CWS*13, LYYB13, TZN*15, TZS*16]. These technologies produce astounding and realistic facial augmentations, especially in applications such as film making and avatar-based video conferencing. However, these augmentations have happened mostly on traditional screens thus far. Leveraging these approaches to physical faces through projection has great potential for a variety of applications. In live theater performances or theme park attractions, stage actions can be enhanced such that the facial properties of actors (e.g., characters, expressions, ages, lighting conditions, etc.) can be changed or modified while acting. In cosmetics, a user can potentially try different maquillage before physical application. Face swapping between two people would open up a novel tele-existence application in which one can 'jack-in' to another distant person.

In this paper, we propose the first *live*, dynamic, markerless human face augmentation system, Makeup Lamps (Figure 1). While tremendous advancements have been made in fundamental technologies, such as geometric registration and radiometric compensation of projection images for dynamic objects, it is still hard to project geometrically consistent illumination on live human faces. In particular, the unavoidable latency of the whole pipeline produces perceivable misalignment of the projected texture to the face. All the face augmentation techniques described above either play predetermined imagery on a known performance, or display results on video monitors, where the original and augmented content is synchronized, possibly in real-time. For the latter, a slight delay between the real and displayed scenes is not perceived. In contrast, live augmentation is performed on the actual face, hence extremely low end-to-end computation time must be achieved to minimize misalignment artifacts.

To achieve the goal of keeping the overall latency at an imperceptible level, the potential complexity of the algorithms used has to be limited, and a high-performance implementation must be carefully designed. On the other hand, the augmentation must still be convincing and accurate enough for human observers. To this end, we apply 2D image interpolation and deformation, to directly match facial landmark positions captured by a high-speed camera, avoiding 3D processing altogether during run-time. With a coaxial projector-camera setup, where the camera and projector share the same optical axis, such simplification is sufficient for an accurate and consistent augmentation. Our system also offers to adapt the augmentation content based on the performer's global position, facial expression and time. In other words, it is able to simulate different lighting conditions and facial effects, such as wrinkles and accentuation, that are expression specific while allowing the performance to change over time. To facilitate such diverse augmentation and maintain the aforementioned timing restrictions, we propose to apply blendshape dimensionality reduction, and prediction through adaptive Kalman filtering. To evaluate our method, we have developed a software and hardware prototype, employing an optimized implementation combining multi-threaded CPU and GPU computations. We demonstrate our prototype's augmentation on several performers without individual training. We also highlight the potential contribution of each individual augmentation aspect (rigid motion, expression, and time), and carefully examine the system's latency and the prediction effect of Kalman filtering. While there is still room for improvement, we believe that demonstrating that projection onto arbitrarily moving faces is possible would have a strong impact on future projection mapping applications, and stimulate further developments for fields such as entertainment and advertisement.

The main contributions of this paper can be summarized as follows:

- A combined hardware and software system enabling robust markerless face tracking and live augmentation.
- A method to reduce processing needs during run-time through blendshape dimensionality reduction, and compensate for inevitable latency using an adaptive Kalman Filter.
- An optimized parallelized CPU/GPU-based framework to minimize overall system latency.

2. Related Work

Various dynamic projection mapping techniques have been developed in the past, both for rigid and deformable surfaces. In the earliest work, a magnetic tracker was applied to measure the pose and position of a rigid projection surface for an online geometric correction [BRF01]. Recent works applied computer vision techniques to estimate this geometric information without requiring any attached devices. For example, the estimation was performed using captured projected textures on the surface [ZSW13, RKK14], printed visual markers [AIS15], or using the information acquired by a depth camera [SCT*15, SYN*16, ZXT*16]. For deformable surfaces, an IR camera and retro-reflective markers were employed [FYT*15] to enable mapping onto a fabric. This method, however, requires a 6DOF (degree-of-freedom) geometric transformation from the camera to the projector, which is generally error-prone. By applying optically aligned projector-camera systems (procams), in which the optical axes of a projector and camera are coaxial with a beam splitter, the error can be reduced to sub-pixel accuracy [Ama14]. Some researchers have applied coaxial procams to realize dynamic projection mapping on deformable surfaces [PIS15, NWI15]. We also apply a coaxial procam system to accurately register the projection images onto the face.

Following pioneering work on projecting a facial animation onto a rigid white mask which was worn by a human actor [YNB*02], several projects have been carried out since on augmenting robotic, and human, faces by projected imagery. In robotics, human facial appearance was displayed on a uniformly white, mechanically controlled surface [HOI*06]. Lincoln et al. [LWN*09] presented a method to augment a fully-colored rigidly moving animatronic head using a projector. The same concept was extended for multiple projectors and flexible, non-rigid animatronic heads by Bermano et al. [BBG*13]; Here the animatronic head was augmented to

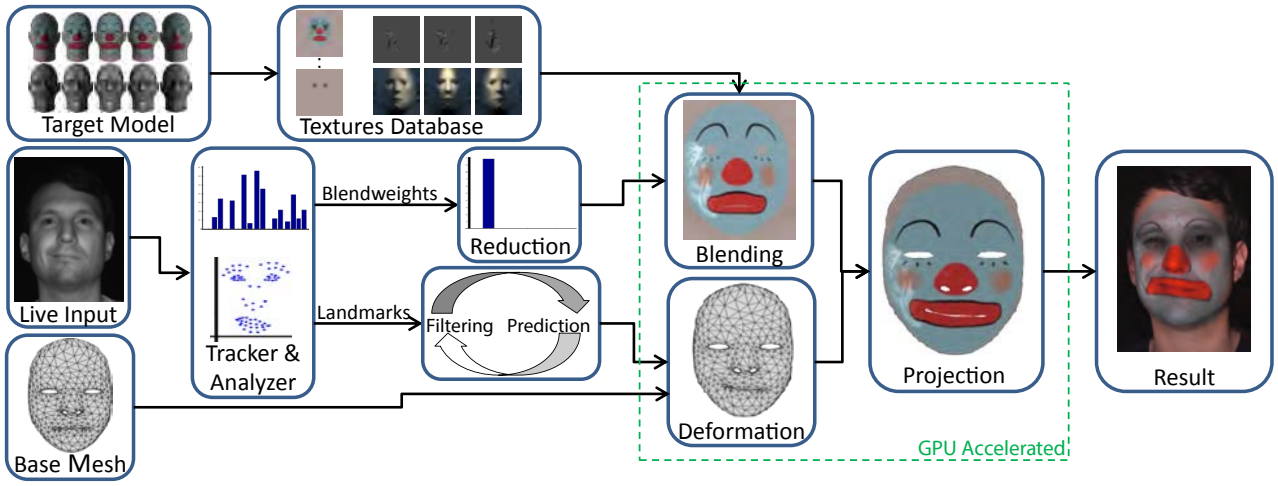

Figure 2: *An overview of our method. A facial rig is deformed and rendered under different illumination conditions. These are stored in an experssion-over-position table. In real-time, a live performer is captured, and the facial expression and configuration are analyzed. The estimated landmark positions are regularized and predicted through a Kalman filter, and are used to deform a 2D mesh. The current time, estimated expression, represented by a reduced set of blendweights, and global position are used to blend the pre-rendered images, and the result is laid over the mesh to create the final augmenting image.*

show more realistic facial expressions by projected imagery. In such robotic applications, a geometrically aligned projection can be achieved by leveraging information from the controlled robotic actuators. For the live scenario, this geometrical information must be measured or estimated in real-time. This was presented, for example, in media art shows of *Omote* and others [OMO16, Lal16, Kat16]. In these examples the facial motion is almost exclusively rigid, and retro-reflective markers were required to be attached in advance. Most recently, Hieda et al. [HC15] proposed a markerless, depth-sensor based approach to interactively paint on a user's face using projection. This method, like all previous ones, does not aim at imperceptible latency for convincing augmentations. In addition, none of these systems is able to estimate the human's expression and adapt the superposition accordingly.

Latency problems are unavoidable in procam systems as other virtual reality (VR) and augmented reality (AR) systems. Especially, the effects of latency are more noticeable in procam systems, as mis-registration between the physical surface and projected imagery is immediately obvious. In the VR and AR research fields, attempts were made to minimize the latency of display systems by both software and hardware approaches. On the software side, warping the images after rendering to the current position is an established approach [MMB97, SvLF08]. Motion prediction using Kalman Filters was also proposed to reduce latency [YNB*02, KBW15]. Custom hardware setups (DLP Discovery Kit and a high-performance FPGA) achieved the latency of less than 100 microseconds [ZWL*14, LBS*16]. In the procams research field, Sueishi et al. [SOI15] proposed a coaxial high speed procam system to realize a dynamic projection mapping on a moving rigid object with no perceptual latency. Such cutting edge equipment is not general, and the displayed images are still limited to primitive ones. In this paper, we apply high performance parallel CPU and GPU processing to minimize the system's latency and use Kalman filter based prediction to achieve a projected augmentation which is not perceived as lagging by the average human visual system.

3. Method

The task of real-time augmentation is a very demanding one. On one hand, the augmentation quality has to be high, especially since in our case the target is a human face. On the other hand, end-to-end computation time must be extremely low. Ng et al. [NLW*12] indicate that human observers, on average, do not perceive the inherent latency of displayed content if the overall latency of the system is below 6.04*ms* (standard deviation 4.33*ms*). Keeping latency levels within this time span limits the potential complexity of the algorithms used, and requires a careful, high-performance implementation. This section describes our proposed method balancing this trade-off.

3.1. Overview

This section gives an overview of the proposed method, and describes how the different steps are combined together to achieve a live facial augmentation system, as illustrated in Figure 2.

The target facial textures are synthesized for various contexts. We have produced them from a facial rig. The rig is rendered using different illumination conditions, expressions, and textures, as explained in Section 3.3. These images are tabulated accordingly and stored as texture maps. At run time, a live performance is captured by a high-speed camera under IR illumination. Employing IR light allows us to separate the actual performance from the one that is augmented by the projector in the visible spectrum (Section 4.1). The head position and configuration are analyzed in order to estimate the expression, through blendweights, and landmark positions. These estimates inevitably contain some noise and delay, so they are fed into a Kalman filter, which reduces noise and predicts the positions at the estimated time of projection. The corrected positions are used

to deform a 2D mesh which corresponds to the performer's face (Section 3.4). The blendweights, along with the face position and current time, are used to blend the texture maps, creating a single texture that is laid over the aforementioned mesh for the final projection image. In order to reduce computational costs and memory usage, we also propose a dimension reduction step, which improves performance and allows longer and more complex performances (see Section 3.3).

In the following, the individual steps of our proposed method is described, from pre-processing steps such as calibration, through the various processing steps to finally generating the projection image.

3.2. Calibration

Since the actor's facial orientation and expression are estimated through camera images, and are used to generate an adequately distorted projection image, both devices must be accurately aligned with each other. Optimally, both should share the same optical path, but practically this is hard to realize if the hardware is not fully customizable. We achieve a sufficient level of accuracy by combining optical alignment with a software based image warping step.

Optical Alignment. To minimize the deviation of the camera's and projector's optical axes, lenses with similar field of views are used for both devices. Furthermore, a beam splitter splits the light path of the projector to the camera, which is placed orthogonally. A custom built 6 DOF mechanical adjustment rig is used in combination with the method described by Amano et al. [Ama14] to achieve an accurate optical alignment.

Camera to Projector Mapping. Since it is impossible to exactly register both devices optically due to mechanical and optical imprecisions, a simple computer-vision based homography registration is applied. This maps the camera coordinates to the projector coordinates via a plane which is placed at the center target location.

3.3. Target Appearance Representation

The goal of our method is to consistently and convincingly augment a performer's facial appearance. To do so, the augmenting content must match the performer both in position and semantics. Section 3.4 explains how the former is achieved. To address the latter, we propose to enhance the expressiveness of the augmentation by adapting the augmenting image content, in contrast to position, according to the performer's intent. Specifically, we propose to alter the performance by taking three aspects into consideration, depicted in Figure 3. First, different facial expressions may be augmented differently. Expression dependent effects could emphasize facial features and lines to amplify an expression or simulate different skin behavior (e.g. older skin). Second, we offer to simulate different lighting conditions. This means that as the performer moves, the facial appearance should reflect the motion. In case of a spotlight, for example, only a small part of the face should be illuminated, depending of the head's position. Lastly, to fully exploit the power of projection based augmentation, we also allow the augmentation to change over time, which can help evolve a story as it is being told. For all of these scenarios, image deformation is not expressive enough, and the augmentation content must be adapted as well, such

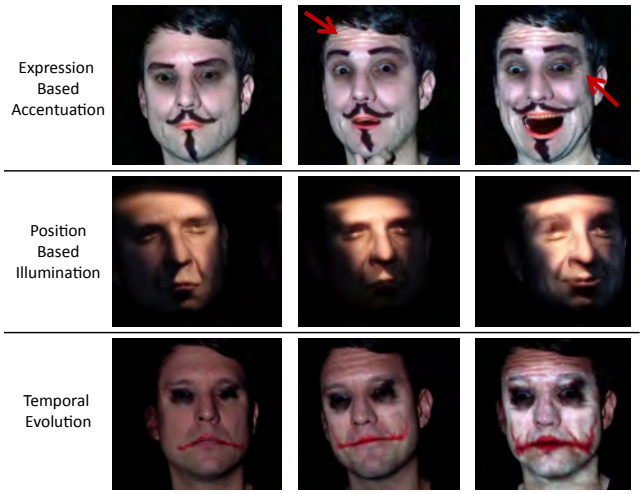

Figure 3: *The different handled augmentation aspects. Upper row: Accentuation of skin deformation details. Middle row: Spatially varying projection simulating a virtual light source. Bottom row: Time evolving augmentation clip of virtually painting the scary face.*

as expression-based accentuation of skin deformation details, simulation of position-based illumination by a virtual light source, and temporally evolving animation projection as shown in Figure 3.

To achieve these goals under the performance constraints, one would naively have to pre-render and store all the combinations of possible expressions in possible locations over time, which is of course not feasible. Instead we offer to sub-sample this multi-dimensional space and linearly interpolate the samples. In the temporal domain, we simply propose a fixed interval quantization of t_h. Meaning, for a performance of length T, we will have $n_t = \frac{T}{t_h}$ samples. In the spatial domain, which is primarily aimed at simulating the environment's lighting conditions, the appearance could dramatically change depending on the 6 DOF or a rigid transformation (Translation in X, Y and Z, and rotation about the 3 axes). Depending on the desired conditions and accuracy level, not all axes play a significant roll, and one can choose how many samples are necessary $n_i, 1 \leq i \leq 6$ for each DOF. We found that for most cases there is no need to sample all axes, i.e. $n_i = 1$ for of most $i = 1, \ldots, 6$ is sufficient (see Section 4.3).

Facial expressions pose a large multi-dimensional, non-linear space that has been explored by many [EF77, CHFT06, CWZ*14]. Hence, captured facial performances are typically used to augment and animate a single, or a set of, predefined meshes. Some approaches apply deformation to a template mesh from 3D motion capture data [BBA*07, ZNI*14], or from landmarks detected directly from a 2D image [CDC*12]. These approaches typically lack wrinkles and details which are expression specific. Most common approaches employ a set of pre-defined meshes, typically named blendshapes, and blend them together to achieve novel expressions [CXH03, ZSCS08, SLS*12].

We propose a combination of the two approaches — we employ both landmarks based deformation to achieve a desired facial configuration, and a blendshape model, for appearance augmentation.

The former enables handling a wide range of extreme expressions, while the latter allows for the accommodation of expression specific details. Furthermore, since the blendshape model is used for appearance only, we are able to drastically compress the model's dimensionality.

For a complete survey of the blendshapes model, we refer the reader to existing surveys [OPA12, LAR*14]. In a nutshell, a set of n_b+1 expressions $B_0,\ldots,B_{n_b}$ are used as a linear basis to span the expression space. A desired expressions is represented as a normalized weighted sum $\sum_{i=0}^{n_b} w_i \cdot B_i$, where the weights w_i are called the *blendweights*. Equivalently, the same basis can be considered as a neutral expression (also known as the *rest pose*) B_0 and a set of offsets $\hat{B}_i = B_i - B_0, i = 1,\ldots,n_b$. In this case, an expression is described as $B = B_0 + \sum_{i=1}^{n_b} w_i \cdot \hat{B}_i$. Due to the non-linear nature of the expression space, the quality of this approximation heavily depends on the number of used blendshapes. For performance optimizations, the compression of the blendshape space has been proposed, reducing the dimensionality of the blending problem [SILN11, LAR*14]. While these solutions offer some compression for general cases, a closer look at our problem setting can help significantly reduce the required amount of samples. As previously mentioned, the deformation step described in Section 3.4 ensures that the projected face shape matches the one of the performer. This means that the blendshapes are computed just to estimate the appearance of a point on the skin, and not its position. Since this task is significantly simpler, it is no surprise that in our experiments, we found that treating the appearance linearly, in a similar fashion to vertex positions, yields satisfying results.

Given the rest pose B_0, we render it under the desired illumination conditions, and convert it to a texture map I_0. Then, for each extreme expression B_i, we render it under the same conditions, and store the difference in illumination for every pixel $\hat{I}_i = I_i - I_0$. Note that since these are texture maps in UV space, pixels correspond to positions on the face, and not physical locations (see Section 3.4). In real time, the value of a texture map I will be computed as a weighted sum $I = I_0 + \sum_{i=1}^{n_b} w_i \cdot \hat{I}_i$, where w_i are the blendweights.

Under the proposed setting, we need to store a texture map for every expression for every sample point in time in every sampled rigid configuration. That is, we have to store $n_t \cdot (n_b+1) \cdot \prod_{i=1}^{6} n_i$ images for a full performance. While quite large, this number is already feasible for limited performances using current graphics hardware. In order to enable longer performances, we offer another aspect to exploit. If the temporal evolution of the augmentation does not aim to simulate a change in geometry, the temporal and spatial effects can be decoupled. This holds for many common cases, such as adding makeup, simulating blushing or face painting. We handle the position and expression space as previously described, disregarding temporal changes. For the temporal axis, we store only the desired varying albedo A_t, i.e. the color to be "painted" on the face for every point in time. At run-time, after linearly interpolating the temporal and spatial axes independently, the desired texture map is computed as a traditional albedo and lighting effects composition $I_t = A_t \otimes I_0 + \sum_{i=1}^{n_b} w_i \cdot \hat{I}_{i,0}$, where $\otimes$ is the component-wise multiplication operator. As illustrated by Figure 4, the required number of stored texture maps now is $n_t + (n_b+1) \cdot \prod_{i=1}^{6} n_i$, allowing performances to be practically arbitrarily long.

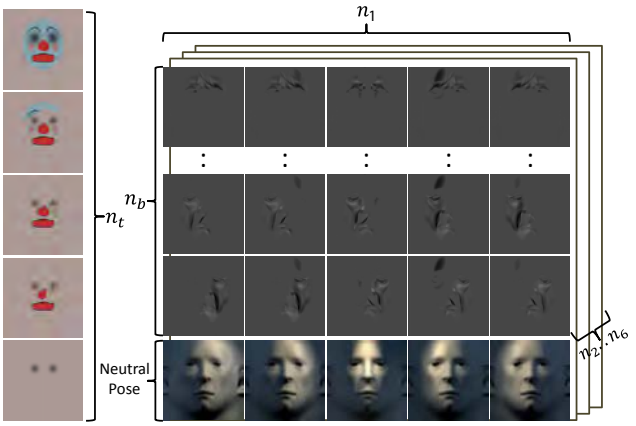

Figure 4: *Target appearance representation. An array of n_t albedo maps is stored to address temporal changes in skin appearance (left). The neutral pose is rendered for every point is space (bottom right). The spanning of the horizontal axis (n_1 samples) is demonstrated. The rest of the n_b blendshapes are rendered and stored as offsets from their respective rest pose (top right)*

Blendshape Dimensionality Reduction. In addition to the aforementioned reduction in memory consumption, we propose to exploit our problem structure even further, and reduce the necessary amount of distinct blendshapes. This reduction effects both the memory footprint of the method and its performance, as it effects the amount of data accessed in real time. Usual blendshape-based facial models employ dozens of base expressions to cover a wide range of facial mimics [LAR*14]. However, as previously mentioned, our task is simpler than traditional models, since positions are already determined by the deformation process (Section 3.4). Our model is only required to estimate the appearance of a point on the skin, or rather it should only reflect changes in skin details during an expression. These details are primarily wrinkling, which occur on the forehead, and around the eyes and mouth regions [MJC*08]. Anatomically speaking, facial expressions are created through the contraction of many muscles. However, not every facial muscular expression is independent. Facial movements can be generally divided into three major groups. The first covers the surrounding areas of the eyebrows, eyes and nose. The second includes muscles distributed around the eyeballs. The third group includes the muscles distributed around the mouth [JKT07, LHC16]. In terms of wrinkle formation, the second group is irrelevant, and the other two could be considered as the only controlling source. We therefore represent wrinkle formation by their main controlling muscles, and consider only the following expressions: eyebrow raising and lowering, lip corner pulling (raising) and lip tightening. One should also consider lip corner depressing (lowering), nose wrinkling, chin raising, and dimpling, however we do not due to limitations of our facial tracker. We found that including the jaw drop (open mouth) expression is also required for the successful mapping process described in Section 3.4. This expression is ignored in run-time (with an offset map of $\hat{I}_6 \equiv 0$), as its functionality is purely technical, and is not required for appearance augmentation. Eventually, for all our experiments we used 6 blendshapes, as depicted in Figure 5.

Figure 5: *The selected expressions of our reduced blendshape model, as projected on the performer. The stored offset image corresponding to each expression is inlaid. Left to right: left lip corner puller, right lip corner puller, eyebrow raiser, eyebrow lowerer, lip tightener and jaw dropper. The is added to facilitate the dimension reduction mapping, and therefore does not have any effect*

3.4. Data Acquisition and Processing

A key component of our system is the fast understanding of facial performances. Facial performance capture has been an active field of research since the dawn of computer graphics and computer-vision [Kan77]. Many approaches, however, aim at highly detailed reconstruction for offline applications [MHP*07, ZSCS08, BHB*11, GVWT13, SKSS14]. Of course, real-time facial tracking methods are more fitting. Some methods employ depth sensors [WBLP11, BWP13, LYYB13, CWS*13]. These sensors, however, are typically not fast enough for our live application. Recently, methods have started to emerge which employ only a single camera, are markerless and, since they depend on heavy priors, are very computationally efficient [CXH03, CWLZ13, CHZ14, Ima14]. Although these methods are restrictive in acquiring extreme expressions and specific details [CBZB15, TZN*15], these methods currently seem to be the only ones fast enough for our almost instantaneous processing needs.

Hence, for a captured frame, we start by analyzing it using one of the aforementioned marker-less facial trackers. We use an off-the-shelf one, as described in Section 4.2. The tracker is fed an image stream captured by a high-speed IR camera, unaffected by the visible light spectrum of the projection. A set of locations in the image is determined, corresponding to distinct facial features like the eyelids, nose and lips (known as *landmarks*). In addition, a set of *blendweights* is estimated, describing the current facial expression. This information is used to determine the projection images.

Facial Expression Estimation. Like most state-of-the-art methods, the tracker used employs a learned model to estimate facial configuration and expression. As described in Section 3.3, an expression is described by a multi-dimensional vector of blendweights, which is larger than required for our setting. To avoid elaborate re-training, we chose to map this high-dimensional space to the dominant wrinkle forming actions by which our target model is represented, instead of integrating them into the facial tracker.

We first experimented with an approximative dimensionality reduction method optimized for fast mapping operations [LSS14]. Since with this method, the mapping does not depend on the number of training samples, it requires relatively less computation time while preserving an acceptable mapping accuracy. However, after comparing accuracy and computation speed, we decided to switch to a method applying a non-linear mapping based on radial basis functions [WM04]. While these methods are able to generate more accurate mapping results, the required operations depend on the number of training samples and thus can be time consuming. Using a careful selection of training samples allowed us to use this method successfully with a comparable processing time but slightly higher accuracy. To train this mapping, the performer performs the reduced number of expressions individually and a few combinations, creating approximately ten short sequences of transitions between the neutral expression to extremities and back. The sequences are then manually annotated, although an automatic approach could be applied, similar to the one proposed by Weise et al. [WLGP09]. As mentioned in Section 3.3, during our experiments we have noticed that, for both methods, opening the mouth would be wrongly mapped to other expressions, and hence it is included as an additional dimension. Note that in addition to reducing the expression space dimensionality, this approach also enables the system to be calibrated to a specific performer and his or her motion gamut without re-training the full detection model. For example, some performers can raise their eyebrows less then others, and hence will never be considered as fully raising them by a general tracker. This is inherently taken into account through the explained process.

Facial Configuration Estimation. State-of-the-art facial trackers provide an indication of distinct facial features, such as eyelids, nose bridge and tips, lips etc. In addition, they typically also provide a similar estimate in 3D space. Beside the obvious performance advantage, we found the 2D landmarks to be natural and accurate to use, since our procam system is coaxial (see Section 3.2). To use the landmarks, we prepare the base 2D mesh by simply flattening the neutral pose of an existing facial rig onto a front looking projection plane. We disregard all back facing regions and preserve the original texture coordinates during the process. By running a rendered version of the flattened mesh through the facial tracker, we register the landmarks onto it. Later, at run-time, we use the landmarks as handles to drive a 2D deformation. We chose the handles to be soft constraints in a Laplacian deformation [Sor05], due to its light computational and implementation load, but any other fast deformation method would be equivalent. In other words, given the mesh $\mathscr{M} = (K,V)$, of the neutral pose, where K describes the connectivity and $V = v_1,\ldots,v_n$ describes vertex positions in $\mathbb{R}^3$. We define $\mathbf{L}$ as the Laplacian matrix such that:

$$\mathbf{L}_{ij} = \begin{cases} 1, & i = j \\ -w_{ij}, & (i,j) \in K \\ 0, & \text{elsewhere} \end{cases} \quad (1)$$

where w_{ij} are the cotangent weights. To register the landmarks $P = p_1 \ldots p_m$, we simply project them onto $\mathscr{M}$ using barycentric coordinates, i.e. we express every landmark $p_i = \sum_{j=1}^{3} u_{ij} \cdot v_j$, where $v_1,\ldots,v_3$ form the triangle in which p_i resides. denoting the Lapla-

cian coordinates of the neutral pose $\delta = \mathbf{L} \cdot V$, at run-time we solve in a least squares manner the system:

$$\tilde{\mathbf{L}} \equiv \begin{pmatrix} \mathbf{L} \\ \omega L_P \end{pmatrix} = \begin{pmatrix} \delta \\ \omega P \end{pmatrix}, \quad (2)$$

where L_P is the matrix that corresponds to the barycentric coordinates representing P in $\mathscr{M}$ and ω is a predefined weight ($\omega = 100$ in all our experiments). Note that this method is not scale invariant, and could generate artifacts as the performer moves to or away from the camera. Therefore, we scale the landmarks positions according to the bounding box of the detected face prior to solving this equation.

Prediction and Denoising. The task of live augmentation is a challenging one mainly due to performance issues. Although minimal, the process described in this section obviously induces some unavoidable latency from capture to projection. In addition, some noise is introduced in the process of capturing and analyzing the face. We address both of these issues by employing a minimum mean-square error (MMSE) estimator and predictor in the form of a Kalman filter [Gel74] on each coordinate of each landmark independently. The formulation we use is common in object tracking in avionics applications, and estimates a third-order dynamics model. We found this model to be adequately balanced in depth. On one hand, the acceleration dimension is required, as commonly practiced mainly due to physical effects interpretation such as force activation. On the other hand, modeling the force behavior even further includes involved and unnecessary considerations. The state vector $X \in \mathbb{R}^3$ is the physical configuration of the point - position, velocity and acceleration (x, v, a). The measurements are only the positions $Z \in \mathbb{R}$, and the estimation/prediction process is a Markov process with some degree of white noise. Following this formulation, we define the process transition $\mathbf{A}$, the process noise covariance $\mathbf{Q}$ and the measurement noise covariance $\mathbf{R}$ as:

$$\mathbf{A} = \begin{pmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & \alpha \end{pmatrix}, \mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_m^2 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \sigma_x^2 \end{pmatrix}, \quad (3)$$

where Δt is the time step, $\alpha = e^{\frac{-\Delta t}{\tau_m}}$, $q_m = \sigma_a(1-\alpha)^2$, and $\tau_m, \sigma_a, \sigma_x$ are parameters, describing the system's decay rate to white noise, standard deviation of acceleration, and of the position, respectively. While not common in graphics and vision applications, this formulation is well established and is relatively easy to control due to its low number of parameters. In all our experiments, we have used $\tau_m = 0.1, \sigma_a = 500, \sigma_x = 0.5$. These parameters were manually tuned, although one could use automatic methods, such as the one proposed by Berman [Ber15]. During run time, we update the process states for each captured image, adaptively changing Δt to be the difference between the time the last image was captured to the previous one. Before rendering each frame, we use the current timestamp to predict the process state at the time of projection. Note that updating Δt effectively means changing only two matrices, that are common to all the running filters, so it is a very lightweight operation. Figure 6 demonstrates how our adaptive prediction scheme is able to successfully predict the physical position of a point on the face, even though the incoming data stream is delayed and irregular. Except for a rapid change in motion direction, the prediction is able to bring the signal difference down to less than 3ms for most of the performance. As can be seen in the accompanying video, the prediction is essential to the perceived projection accuracy.

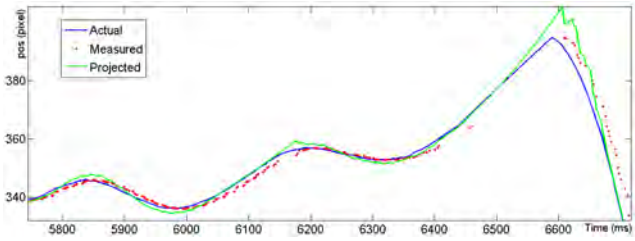

Figure 6: *Filtering and prediction effect. The x-coordinate of a single landmark on the eyebrow is depicted (blue), along with its augmented counterpart at projection time (green). In each frame the predicted value is updated according to the time elapsed from the last measured position (red). As can be seen, the predicted position mostly matches the actual one well, in spite of the delayed and irregular samples.*

3.5. Rendering

Having finally computed the vertex positions, transformed them onto the projector's image plane as described in Section 3.2, and computed performance parameters (blendweights, time, spatial positioning), the geometry is textured accordingly and projected out onto the performer's face. The final projection image is designed to match the performer's facial configuration through deformation, accentuate expressions through texture offsets, and depict color changes through usage of albedo. Note that this process does not take the performer's actual geometry or illumination effects into account. Since the actual geometry is similar to the target one, using albedo as our base of augmentation already compensates for some of the desired illumination effects. Some of the global illumination effects are also considered by baking them into the texture images during rendering. Higher precision registration and global illumination considerations, such as accounting for subsurface scattering or skin tone, still remains as future work, however.

4. Prototype

We set up a prototype to test and evaluate the proposed system. The main component is a projector-camera system, which is optically aligned using a beam splitter as described in Section 3.2. An IR-pass filter mounted in front of the camera lens separates the visible spectrum used by the RGB projection system from the invisible IR illumination. The latter is generated by twelve 4W IR LEDs with a peak wavelength at 850 nm. Figure 7 shows an annotated photograph of the prototype setup.

4.1. Active Hardware

For achieving the required low latency, off-the-shelf projectors are not suitable, and instead a high-frame rate projector with minimal latency is required. As the choice of such devices is significantly limited, we use a customized *Christie Mirage 4K35 DLP projection system* with custom firmware enabling the display of content with 2K resolution at an RGB bit depth of 8 bits and a refresh rate of 480 Hz.

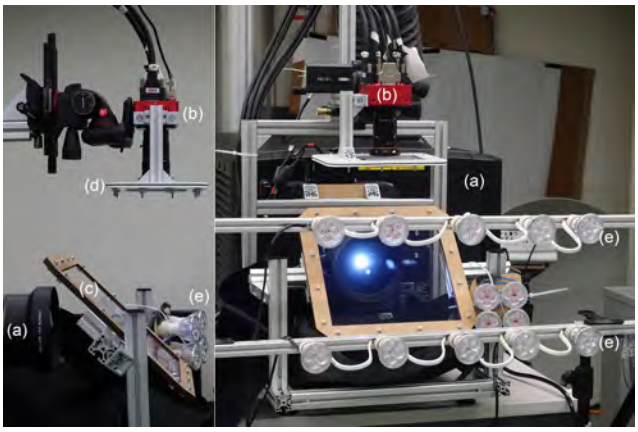

Figure 7: *The hardware prototype: The projector (a) displays 2K images at a frame rate of 480 Hz. The optically aligned high-speed camera (b) is mounted on top the beam splitter (c) placed in front of the projector lens. An IR pass-through filter (d) is used to separate projected light from the IR illumination (e) and capture system.*

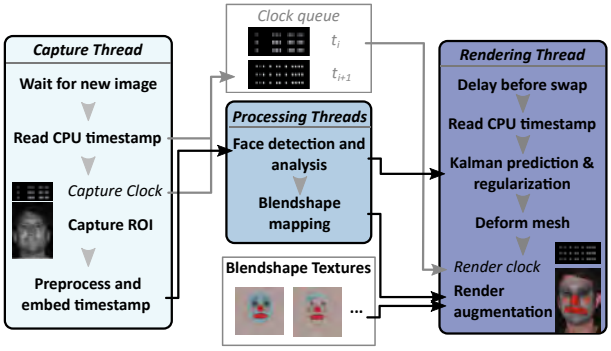

Figure 8: *Dataflow of the prototype. The three threads run asynchronously: the processing thread will discard input images if it cannot keep up with the input image rate. The rendering thread will extrapolate inputs for rendering in between results from the processing thread. Operations related to the clock may be omitted in a production system.*

The projector is driven by an *Nvidia Quadro M6000* GPU using four display ports configured in a synchronized mosaic display setup. The custom-built Intel Xeon workstation additionally contains two high speed *EPIX PIXCI E8* PCI-express frame grabbers which connect the system to an *Allied Vision Bonito CL-400 B* machine vision camera, configured to deliver grey scale images with a resolution of 2320×750 pixels at a frame rate of 1300 Hz. Since the spectra of the input and output devices do not overlap, there is no need to synchronize these two subsystems.

4.2. Software Implementation

The software was implemented using multi-threaded C++ code in combination with GPU-based processing via OpenGL shaders. The prototype has three main processing phases, as depicted in Figure 8, each of which uses one or more independent threads.

Capture. For each iteration, the capture thread waits for a signal indicating that a new image is available. Immediately when the signal is delivered, we record a CPU-timestamp, which is used throughout the different processing steps. Our implementation only reads a small Region Of Interest (ROI) of the image from the frame-grabber. This region is updated after each detection according to facial movement.

Optionally, we read an additional region containing a wall-clock with which we can measure the overall system latency. This image is stored with its associated timestamp in a queue, from which the rendering thread fetches it later. This mechanism is used for the offline latency measurements (Section 4.3).

Processing. Image analysis is performed by an off-the-shelf high performance facial tracker [Ima14]. The software returns (on success) a set of landmarks and animation controls. The former is forwarded to the rendering thread along with the source image timestamp. The latter constitutes the input to the *blendshape dimensionality reduction* (Section 3.3). Input frames may be discarded when the input image rate cannot be kept up with. After optimizing and tweaking the whole system, the face detection step remains the bottle neck in our prototype.

Rendering. For each display frame, the render thread employs the WGL_NV_delay_before_swap [BWR*13] extension to wait for a fixed amount of time (600 μs) before the next possible buffer swap event. This mechanism allows us to maintain a fixed, minimal delay between prediction plus rendering and display of the rendered image. After the delay, the amount of prediction time required is estimated by considering the current CPU timestamp, the timestamp of the last processed image, and a fixed factor (accounting for the latency from the camera, the projection system and buffer swapping).

The Kalman filters are updated if new landmarks were received in the current frame, and are in any case evaluated to predict new landmark positions. These form the input to the mesh deformation step (Section 3.4), which is computed on the GPU using OpenGL compute shaders. We render the deformed mesh using standard OpenGL vertex and fragment shaders. The latter combines the different textures using the blending parameters computed in the processing phase. We use bindless textures to avoid rebinding at each frame, and minimize state changes during rendering in general.

If offline latency measurements are enabled, we retrieve the clock image corresponding to the current input landmarks and display it at a predefined location. Comparing the actual and projected clock images, the total system latency can be measured.

4.3. Results and Performance

To evaluate our method, we experimented with several sets of textures, configurations and performers. Figure 9a,b,c and d show a few frames of performances where each one was done with a single distinct albedo map, unchanged by temporal or spatial considerations, only altered by the expression dependent offset maps. Note the consistency of the projection as it deforms and follows the face during the performance. Figure 9e shows the effects of spatial considerations, as we simulate a spotlight illumination. Moving the head horizontally causes the appearance to change in a manner that seems

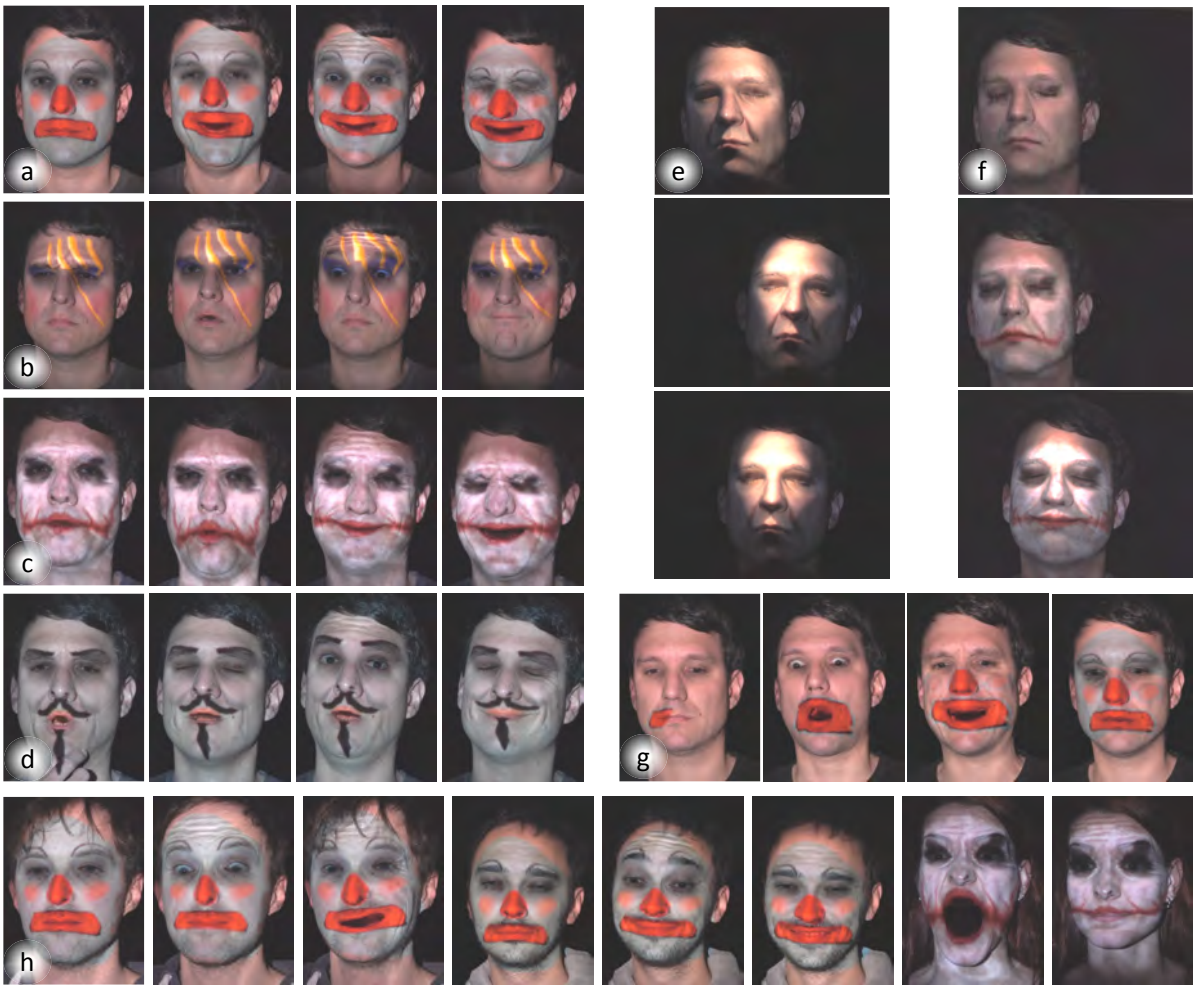

Figure 9: *Example frames from projection experiments. a,b,c,d: Expression dependent augmentation, disregarding space and time. e: Spotlight simulation - horizontal head motion changes the projected image. f: an augmentation which is affected by space, time, and expression considerations. g: a temporally evolving augmentation. h: different performers.*

like it is illuminated by a single narrow lamp. Figure 9g shows a temporally changing performance as it evolves, demonstrating the artistic freedom induced by digital augmentations, and Figure 9f depicts a performance in which all three aspects are considered together. To demonstrate the robustness of our pipeline, we have also captured several different performers (Figure 9h). No person specific training was done for any of the latter performers. For all the examples of this figure, we invite the reader to refer to the accompanying video for more results.

System Performance and Latency. The various processing phases were individually timed and the average results as well as standard deviations are summarized in Table 1. The face tracking performed using third-party software represents the majority of the processing time, and in fact overshadows all other steps. A more optimized face-tracking system could reduce overall latency significantly. Please note that the steps shown in Table 1 are executed in parallel, and hence cannot be directly summed to derive the overall processing latency.

Table 1: *Processing steps timings, over a sequence of* 130*K input frames. Bold text indicates GPU processing.*

Task:	Avg. time	Std. deviation
Image pre-processing	195.6 μs	22.9 μs
Face tracking	5671.3 μs	2851.9 μs
Blendshape reduction	25.4 μs	9.1 μs
Data filtering & **transform**	95.8 & **100.2** μs	27.8 & **2.0** μs
Textured rendering	**180.7** μs	**4.6** μs

We measure the overall latency in two ways. First, we estimate our implementation's latency via timestamps taken when an image is received, and before its final rendering (see Section 4.2). The

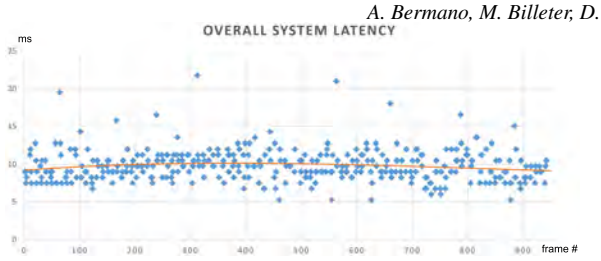

Figure 10: *Plot of the overall system latency measured with the LED-clock–based offline method. The average latency is* 9.8*ms.*

latency is on average 6.6 ms. While very low, this does not describe the end-to-end latency including acquisition and projection delays.

Additionally, we measure the latency in an offline manner. These measurements do include latency arising from the camera, display and projector hardware. To this end, we use an Arduino-based LED clock that displays time stamps encoded as gray codes in both IR and visible spectra [BRW*16]. The system captures an image of the clock together with the input image, and projects the clock back along with the matching input frame's results. Both the live LED-display of the device and the projected image are captured with an external camera (a *Sony RX100 IV*) at a frame rate synchronized to the projector. Each frame contains the current time shown on the image as well as the time at which the image was captured. Measured latencies are shown in Figure 10. The overall system latency is on average 9.8*ms* with a standard deviation of 2.1*ms*.

5. Summary and Discussion

In summary, we presented the first real-time, non-rigid dynamic projection method to augment human faces with adaptively projected content depending on facial semantics. Our system is primarily designed for high performance, minimizing overall latency through a combined multi-threaded CPU and GPU implementation. The developed algorithms, combined with the presented hardware, are able to produce convincing illusions, which are perceived as being fixed onto the mimics of the augmented human face. This is due to successful compensation of the unavoidable overall system latency through adaptive Kalman based filtering and prediction. Our method is not only able to track and match the captured facial shape, but also detects its semantics and efficiently adapts the rendered augmentation content accordingly. Hence, we have shown that the classic limitations to dynamic projection based augmentation (such as slow and rigid head movement, the use of optical markers, and displaying only non-adapting or fully precomputed content) are no longer necessary when generating convincing facial augmentations.

Limitations and Future Work. Although the current prototype is agnostic to the face tracking method used, the resulting augmentation quality strongly depends on the performance of this part of the pipeline. Observing the video results, one can clearly see the lack of facial boundaries and the tracking noise, which show the limitations of the used facial tracking library. New methods, focused on minimizing latency [IIGT12, RCWS14], reduced noise levels, and more detailed tracking can further improve projection quality. Since the current system is realized using a single frontal projection, the sides of the face cannot be fully illuminated. Overcoming this constraint to fully cover the facial surface is one of the future research directions to investigate. Note that projecting from the sides requires greater accuracy, due to depth discontinuities introduced especially around the nose. A significant improvement can be achieved through more elaborate compensation schemes. Skin tone, subsurface scattering and other global illumination effects can be accounted for, allowing to eliminate facial features, and not just adding them. This, however would require out-of-budget computation time, and an investigation of how to incorporate such considerations would be very beneficial and challenging. For the same reasons, comparing the results of our live application and previous ones, which aim for quality instead of latency, is not informative. It would be interesting to find a setting in which a user study or other forms of comparisons is meaningful. Another limitation of the current method is the face-dependent preprocessing step to generate the reduced set of blendshapes, and face boundaries. Generating them automatically on-the-fly or via a quick interactive calibration routine would make the system more flexible in situations where the projection should be carried out onto a variety of human faces. The demonstrated setup currently involves placing the actor close in front of the relatively bulky hardware. This choice is primarily related to the physical space available for our prototype installation. The procam system's capabilities allow the projection to take place from a much larger distance, especially when combined with the appropriate lenses. The angular region can be adjusted likewise, albeit doing so may affect resolution. Cost of the prototype setup is a concern, although we believe that the current trend towards higher frame rate displays is likely to bring down costs of projectors in the future as well.

Due to the direct projection onto an actor's face, (eye-)safety is a concern, and warrants an investigation, especially before use in a production setting. For our tests, we used minimal projection brightness, and further applied dark colors to the regions around the eyes. This setting was comfortable for the performer. However even without dark projection around the eyes, the experience is comparable to illumination from a bright scene-spot light (as used in various entertainment events and venues).

We imagine this system to be applicable exclusively for live audience performances. At its current configuration, it is restricted to a stage setting (as opposed to a street performance setting). A pan/tilt system (similar to a spotlight following the performer on stage) could probably be employed to expand the active region. The introduced motion would obviously be somewhat delayed compared to the actual performance. Intuitively, this should be possible to compensate for through our already existing tracking and prediction mechanism (the pan and tilt's acceleration would simply be added to the performer's). Of course, this requires careful examination and is an interesting aspect for future systems.

We believe that projection-based non-rigid expressive augmentation has the potential to give rise to a large variety of new creative application scenarios in the near future, and that the methods described here are the fundamental building blocks toward its realization.

Acknowledgements

The authors would like to thank Andreas Skyman for voicing the associated video.

References

[AIS15] ASAYAMA H., IWAI D., SATO K.: Diminishable visual markers on fabricated projection object for dynamic spatial augmented reality. In *SIGGRAPH Asia 2015 Emerging Technologies* (2015), ACM, pp. 7:1–7:2. doi:10.1145/2818466.2818477. 2

[Ama14] AMANO T.: Projection Center Calibration for a Co-located Projector Camera System. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), pp. 449–454. doi:10.1109/CVPRW.2014.72. 2, 4

[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M.: Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 33. doi:10.1145/1276377.1276419. 4

[BBG*13] BERMANO A., BRÜSCHWEILER P., GRUNDHÖFER A., IWAI D., BICKEL B., GROSS M.: Augmenting physical avatars using projector-based illumination. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 189:1–189:10. doi:10.1145/2508363.2508416. 2

[Ber15] BERMAN Z.: *Advances in Estimation, Navigation, and Spacecraft Control: Selected Papers of the Itzhack Y. Bar-Itzhack Memorial Symposium on Estimation, Navigation, and Spacecraft Control*. Springer Berlin Heidelberg, 2015, ch. Efficient Error Model Construction, pp. 191–208. doi:10.1007/978-3-662-44785-7_11. 7

[BHB*11] BEELER T., HAHN F., BRADLEY D., BICKEL B., BEARDSLEY P., GOTSMAN C., SUMNER R. W., GROSS M.: High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 75. doi:10.1145/2010324.1964970. 6

[BRF01] BANDYOPADHYAY D., RASKAR R., FUCHS H.: Dynamic shader lamps: painting on movable objects. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality* (2001), pp. 207–216. doi:10.1109/ISAR.2001.970539. 2

[BRW*16] BILLETER M., RÖTHLIN G., WEZEL J., IWAI D., GRUNDHÖFER A.: A LED-based IR/RGB end-to-end latency measurement device. In *Adjunct Proceedings of IEEE International Symposium on Mixed Reality and Augmented Reality* (2016), pp. 184–188. URL: https://www.disneyresearch.com/publication/a-led-based-irrgb-end-to-end-latency-measurement-device/. 10

[BWP13] BOUAZIZ S., WANG Y., PAULY M.: Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 40. doi:10.1145/2461912.2461976. 2, 6

[BWR*13] BRETON J., WERNESS E., RITGER A., JONES J., EVERITT C., CARMACK J.: NV_delay_before_swap. In *OpenGL Extension Registry*. 2013. URL: https://www.opengl.org/registry/specs/NV/wgl_delay_before_swap.txt. 8

[CBZB15] CAO C., BRADLEY D., ZHOU K., BEELER T.: Real-time high-fidelity facial performance capture. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 46. doi:10.1145/2766943. 6

[CDC*12] CHOI J., DUMORTIER Y., CHOI S.-I., AHMAD M. B., MEDIONI G.: Real-time 3-d face tracking and modeling from a webcam. In *IEEE Workshop on Applications of Computer Vision (WACV)* (2012), pp. 33–40. doi:10.1109/WACV.2012.6163031. 4

[CHFT06] CHANG Y., HU C., FERIS R., TURK M.: Manifold based analysis of facial expression. *Image and Vision Computing 24*, 6 (2006), 605 – 614. Face Processing in Video Sequences. doi:10.1016/j.imavis.2005.08.006. 4

[CHZ14] CAO C., HOU Q., ZHOU K.: Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph. 33*, 4 (July 2014), 43:1–43:10. URL: http://doi.acm.org/10.1145/2601097.2601204, doi:10.1145/2601097.2601204. 2, 6

[CWLZ13] CAO C., WENG Y., LIN S., ZHOU K.: 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 41. doi:10.1145/2461912.2462012. 2, 6

[CWS*13] CHEN Y.-L., WU H.-T., SHI F., TONG X., CHAI J.: Accurate and robust 3D facial capture using a single RGBD camera. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 3615–3622. doi:10.1109/ICCV.2013.449. 2, 6

[CWZ*14] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (March 2014), 413–425. doi:10.1109/TVCG.2013.249. 4

[CXH03] CHAI J.-X., XIAO J., HODGINS J.: Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 193–206. doi:10.2312/SCA03/193-206. 2, 4, 6

[EF77] EKMAN P., FRIESEN W. V.: Facial action coding system. 4

[FYT*15] FUJIMOTO Y., YAMAMOTO G., TAKETOMI T., SANDOR C., KATO H.: [POSTER] Pseudo Printed Fabrics through Projection Mapping. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2015), 174–175. doi:10.1109/ISMAR.2015.51. 2

[Gel74] GELB A.: *Applied Optimal Estimation*. MIT Press, 1974. URL: https://books.google.com/books?id=KlFrn8lpPP0C. 7

[GVWT13] GARRIDO P., VALGAERTS L., WU C., THEOBALT C.: Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 158–1. doi:10.1145/2508363.2508380. 6

[HC15] HIEDA N., COOPERSTOCK J. R.: Digital facial augmentation for interactive entertainment. In *Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on* (June 2015), pp. 9–16. URL: http://ieeexplore.ieee.org/document/7325479/. 3

[HOI*06] HAYASHI K., ONISHI Y., ITOH K., MIWA H., TAKANISHI A.: Development and evaluation of face robot to express various face shape. In *Proceedings of the IEEE International Conference on Robotics and Automation* (May 2006), pp. 481–486. doi:10.1109/ROBOT.2006.1641757. 2

[IIGT12] ISHII I., ICHIDA T., GU Q., TAKAKI T.: 500-fps face tracking system. *Journal of Real-Time Image Processing 8*, 4 (2012), 379–388. doi:10.1007/s11554-012-0255-8. 10

[Ima14] IMAGE METRICS LTD.: LiveDriver by Image Metrics. http://www.image-metrics.com/, 2014. 6, 8

[JKT07] JENKINS G. W., KEMNITZ C. P., TORTORA G. J.: *Anatomy and physiology: from science to life*. John Wiley and Sons Inc., 2007. URL: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-EHEP002147.html. 5

[Kan77] KANADE T.: *Computer recognition of human faces*, vol. 47. Birkhäuser, 1977. 6

[Kat16] KAT VON D: Live face projection mapping with Kat Von D, 2016. (visited 10/2016). URL: https://vimeo.com/143267919. 3

[KBW15] KNIBBE J., BENKO H., WILSON A. D.: Juggling the effects of latency: Software approaches to minimizing latency in dynamic projector-camera systems. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (2015), ACM, pp. 93–94. doi:10.1145/2815585.2815735. 3

[Lal16] LALWANI M.: Inside Lady Gaga's high-tech grammy performance, 2016. (visited 10/2016). URL: https://www.engadget.com/2016/02/18/inside-lady-gagas-high-tech-grammy-performance/. 3

[LAR*14] LEWIS J. P., ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. In *Eurographics (State of the Art Reports)* (2014), pp. 199–218. doi:10.2312/egst.20141042. 5

[LBS*16] LINCOLN P., BLATE A., SINGH M., WHITTED T., STATE A., LASTRA A., FUCHS H.: From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. In *IEEE Virtual Reality (VR)* (2016), IEEE. doi:10.1109/TVCG.2016.2518038. 3

[LHC16] LIN C.-Y., HUANG C.-C., CHENG L.-C.: An expressional simplified mechanism in anthropomorphic face robot design. *Robotica 34* (3 2016), 652–670. doi:10.1017/S0263574714001787. 5

[LSS14] LE Q. V., SARLÓS T., SMOLA A. J.: Fastfood: Approximate kernel expansions in loglinear time. *CoRR abs/1408.3060* (2014). URL: https://arxiv.org/abs/1408.3060. 6

[LWN*09] LINCOLN P., WELCH G., NASHEL A., ILIE A., STATE A., FUCHS H.: Animatronic shader lamps avatars. In *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality* (2009), IEEE Computer Society, pp. 27–33. doi:10.1109/ISMAR.2009.5336503. 2

[LYYB13] LI H., YU J., YE Y., BREGLER C.: Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 42–1. doi:10.1145/2461912.2462019. 2, 6

[MHP*07] MA W.-C., HAWKINS T., PEERS P., CHABERT C.-F., WEISS M., DEBEVEC P.: Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (2007), Eurographics Association, pp. 183–194. doi:10.2312/EGWR/EGSR07/183-194. 6

[MJC*08] MA W.-C., JONES A., CHIANG J.-Y., HAWKINS T., FREDERIKSEN S., PEERS P., VUKOVIC M., OUHYOUNG M., DEBEVEC P.: Facial performance synthesis using deformation-driven polynomial displacement maps. In *ACM SIGGRAPH Asia 2008 Papers* (2008), pp. 121:1–121:10. doi:10.1145/1457515.1409074. 5

[MMB97] MARK W. R., MCMILLAN L., BISHOP G.: Post-rendering 3D warping. *Symposium on Interactive 3D Graphics*, Figure 2 (1997), 7–16. doi:10.1145/253284.253292. 3

[NLW*12] NG A., LEPINSKI J., WIGDOR D., SANDERS S., DIETZ P.: Designing for low-latency direct-touch input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (2012), pp. 453–464. doi:10.1145/2380116.2380174. 3

[NWI15] NARITA G., WATANABE Y., ISHIKAWA M.: Dynamic projection mapping onto a deformable object with occlusion based on high-speed tracking of dot marker array. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology* (2015), pp. 149–152. doi:10.1145/2821592.2821618. 2

[OMO16] OMOTE: OMOTE – real-time face tracking & projection mapping, 2016. (visited 10/2016). URL: https://vimeo.com/103425574. 3

[OPA12] ORVALHO V., PARKE F. P., ALVAREZ X.: A facial rigging survey. In *Eurographics* (May 2012), vol. 32, pp. 10–32. doi:10.2312/conf/EG2012/stars/183-204. 5

[PIS15] PUNPONGSANON P., IWAI D., SATO K.: Projection-based visualization of tangential deformation of nonrigid surface by deformation estimation using infrared texture. *Virtual Reality 19*, 1 (Mar. 2015), 45–56. doi:10.1007/s10055-014-0256-y. 2

[RCWS14] REN S., CAO X., WEI Y., SUN J.: Face alignment at 3000 FPS via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1685–1692. doi:10.1109/CVPR.2014.218. 10

[RKK14] RESCH C., KEITLER P., KLINKER G.: Sticky projections — a new approach to interactive shader lamp tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Sept 2014), pp. 151–156. doi:10.1109/ISMAR.2014.6948421. 2

[RWF99] RASKAR R., WELCH G., FUCHS H.: Spatially augmented reality. In *Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes* (Natick, MA, USA, 1999), IWAR '98, A. K. Peters, Ltd., pp. 63–72. URL: http://dl.acm.org/citation.cfm?id=322690.322696. 2

[SCT*15] SIEGL C., COLAIANNI M., THIES L., THIES J., ZOLLHÖFER M., IZADI S., STAMMINGER M., BAUER F.: Real-time pixel luminance optimization for dynamic multi-projection mapping. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 237:1–237:11. doi:10.1145/2816795.2818111. 2

[SILN11] SEO J., IRVING G., LEWIS J., NOH J.: Compression and direct manipulation of complex blendshape models. *ACM Transactions on Graphics (TOG) 30*, 6 (2011), 164. doi:10.1145/2070781.2024198. 5

[SKSS14] SUWAJANAKORN S., KEMELMACHER-SHLIZERMAN I., SEITZ S. M.: Total moving face reconstruction. In *Computer Vision–ECCV*. Springer, 2014, pp. 796–812. doi:10.1007/978-3-319-10593-2_52. 6

[SLS*12] SEOL Y., LEWIS J., SEO J., CHOI B., ANJYO K., NOH J.: Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG) 31*, 2 (2012), 14. doi:10.1145/2159516.2159519. 4

[SOI15] SUEISHI T., OKU H., ISHIKAWA M.: Robust high-speed tracking against illumination changes for dynamic projection mapping. In *IEEE Virtual Reality (VR)* (March 2015), pp. 97–104. doi:10.1109/VR.2015.7223330. 3

[Sor05] SORKINE O.: Laplacian Mesh Processing. In *Eurographics 2005 - State of the Art Reports* (2005), The Eurographics Association. doi:10.2312/egst.20051044. 6

[SvLF08] SMIT F. A., VAN LIERE R., FRÖHLICH B.: An image-warping VR-architecture: Design, implementation and applications. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (2008), pp. 115–122. doi:10.1145/1450579.1450605. 3

[SYN*16] SAAKES D., YEO H.-S., NOH S.-T., HAN G., WOO W.: Mirror mirror: An on-body t-shirt design system. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2016), ACM, pp. 6058–6063. doi:10.1145/2858036.2858282. 2

[TZN*15] THIES J., ZOLLHÖFER M., NIESSNER M., VALGAERTS L., STAMMINGER M., THEOBALT C.: Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 183. doi:10.1145/2816795.2818056. 2, 6

[TZS*16] THIES J., ZOLLHÖFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Face2Face: Real-time face capture and reenactment of RGB videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* (2016). URL: http://www.graphics.stanford.edu/~niessner/thies2016face.html. 2

[WBLP11] WEISE T., BOUAZIZ S., LI H., PAULY M.: Realtime performance-based facial animation. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 77. doi:10.1145/2010324.1964972. 2, 6

[WLGP09] WEISE T., LI H., GOOL L. V., PAULY M.: Face/off: Live facial puppetry. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (Proc. SCA'09)* (August 2009). doi:10.1145/1599470.1599472. 6

[WM04] WILSON D. C., MAIR B. A.: *Sampling, Wavelets, and Tomography*. Birkhäuser Boston, 2004, ch. Thin-Plate Spline Interpolation, pp. 311–340. doi:10.1007/978-0-8176-8212-5_12. 6

[YNB*02] YOTSUKURA T., NIELSEN F., BINSTED K., MORISHIMA S., PINHANEZ C. S.: Hypermask: Talking Head Projected onto Real Object. *The Visual Computer 18*, 2 (2002), 111–120. doi:10.1007/s003710100140. 2, 3

[ZNI*14] ZOLLHÖFER M., NIESSNER M., IZADI S., REHMANN C., ZACH C., FISHER M., WU C., FITZGIBBON A., LOOP C., THEOBALT C., ET AL.: Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 156. doi:10.1145/2601097.2601165. 4

[ZSCS08] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: High-resolution capture for modeling and animation. In *Data-Driven 3D Facial Animation*. Springer, 2008, pp. 248–276. doi:10.1007/978-1-84628-907-1_13. 4, 6

[ZSW13] ZHENG F., SCHUBERT R., WEICH G.: A general approach for closed-loop registration in ar. In *2013 IEEE Virtual Reality (VR)* (March 2013), pp. 47–50. doi:10.1109/VR.2013.6549358. 2

[ZWL*14] ZHENG F., WHITTED T., LASTRA A., LINCOLN P., STATE A., MAIMONE A., FUCHS H.: Minimizing latency for augmented reality displays: Frames considered harmful. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Sept 2014), pp. 195–200. doi:10.1109/ISMAR.2014.6948427. 3

[ZXT*16] ZHOU Y., XIAO S., TANG N., WEI Z., CHEN X.: Pmomo: Projection mapping on movable 3D object. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (2016), ACM, pp. 781–790. doi:10.1145/2858036.2858329. 2