Synthetic prior design for real-time face tracking

Steven McDonagh*, Martin Klaudiny*, Derek Bradley, Thabo Beeler, Iain Matthews and Kenny Mitchell
Disney Research
{steven.mcdonagh, martin.klaudiny, kenny.mitchell}@disneyresearch.com

Abstract

Real-time facial performance capture has recently been gaining popularity in virtual film production, driven by advances in machine learning, which allows for fast inference of facial geometry from video streams. These learning-based approaches are significantly influenced by the quality and amount of labelled training data. Tedious construction of training sets from real imagery can be replaced by rendering a facial animation rig under on-set conditions expected at runtime. We learn a synthetic actor-specific prior by adapting a state-of-the-art facial tracking method. Synthetic training significantly reduces the capture and annotation burden and in theory allows generation of an arbitrary amount of data. But practical realities such as training time and compute resources still limit the size of any training set. We construct better and smaller training sets by investigating which facial image appearances are crucial for tracking accuracy, covering the dimensions of expression, viewpoint and illumination. A reduction of training data in 1-2 orders of magnitude is demonstrated whilst tracking accuracy is retained for challenging on-set footage.

1. Introduction

In recent years, real-time markerless facial performance capture has received a lot of attention both from academia and industry. Many of the proposed methods are generic in that they do not require any user-specific training upfront, and as such are extremely flexible in their use. The downside of these methods is however reduced accuracy, particularly when it comes to person specific features, but also in the overall shape and face appearance. Other methods promise greater accuracy when training the method for a particular person. Such training requires user specific input data, such as images and potentially geometry, as well as labels for that data. Since the variation of the facial appearance caused by changing expressions is rather substantial, a relatively large training set typically consisting of dozens of images has to be provided to train the algorithm for a single illumination condition from a single viewpoint. Unfortunately, restricting acquisition to a single known illumination and viewpoint is often too much of a limitation and most methods extrapolate poorly to footage acquired under different environmental conditions and/or from different cameras. Training the methods for large variation in lighting and viewpoint would lead to an unbearable amount of labour to acquire and label the required training data. Furthermore, the required training data can typically only be acquired under certain conditions, precluding training the model for different scenarios. An example is the established industry practice to capture an actor in a well calibrated, highly-constrained capture setup first to create a high quality digital double. This digital puppet is then controlled from video recorded under vastly different conditions, including different and time varying illumination as well as different camera viewpoints, optics and sensors.

Motivated by these challenges, we propose to instead synthetically generate the required training data, tuned to the expected environmental conditions and camera properties. We discuss in this paper how to effectively use fully synthetic training for the purpose of facial performance capture. While synthetic training reduces the burden to capture and annotate data significantly, and in theory allows generation and training on arbitrarily large amounts of data, practical realities such as training time and compute resources still limit the size of any training set. We show how to construct better and smaller training sets without sacrificing tracking performance by analyzing which facial appearances, covering the dimensions of expression, viewpoint and illumination, require denser sampling. Our insights will enable more informed decisions on how to construct an actor specific training set when given a defined budget of time, which is particularly important for on-set facial capture in film and video game productions.

2. Related work

Current methods for markerless 3D facial performance capture achieve very high quality 3D reconstruction [5, 8, 18, 23, 31, 34]. However, high accuracy typically comes

*Joint first authors.

at the cost of high computation time, and therefore recent methods have focused on real-time face tracking, generally sacrificing accuracy for speed. A common approach is to pre-train a system to recognize faces in different poses and expressions, and under different environmental conditions, and then infer the 3D facial performance given novel input video at run-time.

Generic real-time trackers are typically built from large databases of many different people posed in many different environments. Some databases consist of images (e.g. HELEN [24], LFPW [6], 300-W [29]), and others contain 3D face scans (e.g. BU-4DFE [38], BP4D-Spontaneous [39], FaceWarehouse [12]), but in both cases individual features of the face must be labelled to provide training data, which is tedious and time-consuming. Person-independent techniques include Active Appearance Models [3, 15, 26] and other deformable model fitting [30], regression-based methods that infer facial landmark positions [13, 20, 21, 28, 40] or feature trackers using prior motion capture data [14]. While exhibiting robust performance for general facial images, drawbacks of generic trackers are the lack of person-specific details and that they are often more prone to failures resulting from local minima.

To obtain higher fidelity tracking, several methods propose an offline preprocess to build a person-specific prior. Tracking of 2D facial features can involve learning a regressor on a training video of the subject [27] or incrementally re-training to adapt to the subject [4]. A common approach for 3D facial tracking is to build a person-specific facial rig. The 3D rig is driven in real-time according to structured light scans [36], RGBD images [32, 35] or monocular RGB stream [19, 33]. Cao et al. [11] regress to 3D landmark positions after training on images of person-specific expressions and head pose and then fit the facial rig. It is possible to regress directly to rig expression parameters and head pose as in [37] and also consider training images captured under varying illuminations. Alternatively, the person-specific facial rig may be built adaptively during real-time tracking, in so-called "online learning" methods, avoiding the apriori training process [7, 10, 25]. Cao et al. [9] build upon their previous work [10] by regressing to actor wrinkle detail, given generic 3D wrinkle training data, but the overall fidelity of actor-specific details is still far from that of offline methods. The main problem with person-specific tracking is the extreme difficulty and laborious task of capturing and labelling training data for all scenarios that might occur during runtime, including expression changes, camera viewpoint, and changing environment illumination.

Our key observation is to use synthetic training imagery, with inherent data labelling, for high-quality actor-specific facial tracking. Real-time head pose estimation from depth images using random forests [16] utilizes synthetic training by rendering depth maps of a face model undergoing large rotations. In the generic facial tracker of Jeni et al. [20], the regression is trained for a number of different viewpoints by synthetically rendering a database of facial scans. Feng et al. [17] use a 3D morphable face model to generate synthesized faces for regression-based training, adding head pose variations to augment real training data. While similar in spirit to our approach, these methods do not consider synthesizing training imagery tailored to camera properties and different illumination conditions in the target capture environment. We will show this is essential for achieving high-quality tracking in real-world environments. To our knowledge, ours is also the first facial tracking work to systematically investigate informed reductions of training set size, enabling reduced computation time while maintaining tracking accuracy.

3. Regression based face tracking

Our regression framework for real-time face tracking learns a mapping between images captured by RGB camera and an actor blend shape rig. In this regard our framework follows previous work [11, 37] which train actor-specific regressors using real imagery. Figure 1 depicts a high-level workflow illustrating our pipeline consisting of offline training and online tracking stages.

To train our framework we firstly construct a blendshape rig $\mathbf{B} = \{B_j\}_{j=0}^{J}$ for the target actor. The face rig model has shape and appearance components that enable generating realistic synthetic facial imagery for training purposes (see Section 4). Face model state $\mathbf{S} = (\mathbf{a}, \mathbf{r}, \mathbf{t})$ describes a facial expression $\mathbf{a}$ and camera pose $(\mathbf{r}, \mathbf{t})$ with respect to the face. The blend weight vector $\mathbf{a}$ defines the 3D shape of the expression $B = B_0 + \sum_{j=1}^{J} a_j(B_j - B_0)$. The pose of a camera with known intrinsic parameters is represented by 3D rotation vector $\mathbf{r}$ and 3D translation vector $\mathbf{t}$. A training sample pair $(I_n, \hat{\mathbf{S}}_n)$ consists of a synthetic face image I_n and the ground-truth face model state $\hat{\mathbf{S}}_n$ used to render it. Training set construction produces N such sample pairs that facilitate learning of the mapping (see Section 4).

Our framework predicts the change in face model state $\mathbf{S}$ between a preceding frame and the current frame when presented with the image I that corresponds to the current frame. This enables online tracking of consecutive frames. The functionality is implemented by augmenting input training pairs $(I_n, \hat{\mathbf{S}}_n)$ with a set of initial states $\mathbf{S}_m$ that model potential state transitions. The training samples provided to the algorithm therefore have the form $(I_m, \hat{\mathbf{S}}_m, \mathbf{S}_m)$. The first group of potential initial states describe expression transitions which are formed by the m_E expressions closest to $\hat{\mathbf{a}}_n$ in all N training pairs. Similarity of two expressions is computed as the sum of 3D Euclidean distances between the two face shapes. The second group of initial states describe camera pose transitions, where the ground-truth pose $(\hat{\mathbf{r}}_n, \hat{\mathbf{t}}_n)$ is locally perturbed

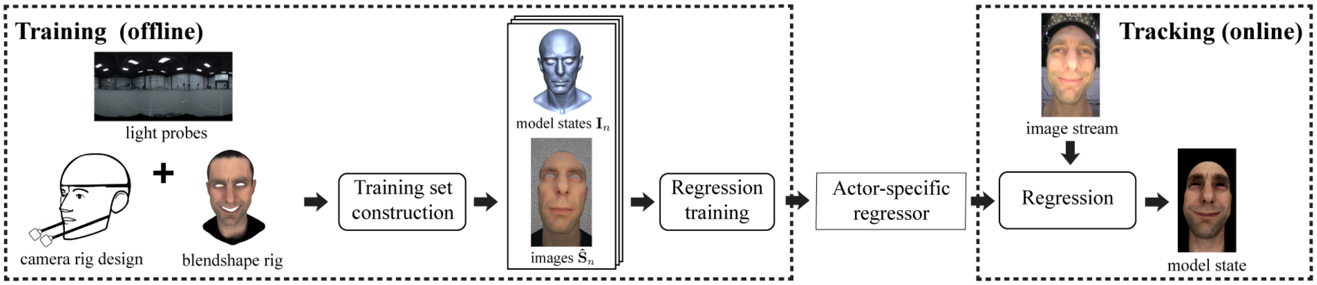


Figure 1: Overview of real-time face tracking framework.

m_T times. Specifically, we generate multiple spatial offsets from $(\hat{\mathbf{r}}_n, \hat{\mathbf{t}}_n)$ with constant step sizes along each translation axis and around each rotation axis. This sample augmentation strategy differs from previous methods [10,11,37], providing simpler design yet aiding regression stability. Our training set augmentation process expands the final number of training samples to $M = (m_E + m_T) \cdot N$.

The cascaded regression scheme introduced by Cao et al. [13] is used to learn the mapping between input image features sampled from I_m and transitions from $\mathbf{S}_m$ to $\hat{\mathbf{S}}_m$. This scheme consists of T stage regressors containing sequential chains of F random ferns. Input features for weak fern regressors are greyscale pixel differences projected and sampled from 3D point pairs randomly scattered across the facial rig. These features are mapped to learned increments of $\delta\mathbf{S}$. The D pixel differences are selected from a pool of U sampling points for each fern independently. Correlation across all M training samples is computed between possible pixel differences and the residuals $(\mathbf{S}_m - \hat{\mathbf{S}}_m)$ and the most correlated differences are selected. Because the output variables $(\mathbf{a}, \mathbf{r}, \mathbf{t})$ have different scales, in a vein similar to [15] we weight their influence on the correlation in a principled way in contrast to [37]. A unit change is applied to a particular target variable and we compute the resulting change on the rig mesh. The weights are determined according to normalised magnitudes of this 3D rig mesh change. We also enforce spatial locality of features [21], which increases their robustness against illumination change. We employ a simple 3D distance threshold ($\sim$ $10mm$) to reduce the pool of U^2 potential pixel differences prior to correlation computation.

Once trained the regressor can track faces online using a monocular image stream as input. The random fern approach provides real-time estimation of model state $\mathbf{S}$. To increase robustness and temporal coherence of the final solution, multiple independent regressions run in parallel at every frame and their results are averaged together. They are initialised from different model states $\mathbf{S}_l$ which are derived from the solution $\tilde{\mathbf{S}}$ in the previous frame. We search for the l_E closest expressions and l_T closest camera transforms to $\tilde{\mathbf{S}}$ in all training pairs $(I_n, \hat{\mathbf{S}}_n)$, applying the similarity metric based on vertex distances used during training. As a last step, we employ a light-weight Gaussian temporal filter with window size w to aid temporal smoothness.

4. Training set construction

The performance of any learning-based method is heavily influenced by the quality and amount of available training data. Construction of a substantial actor specific training set, consisting of conventional facial images, requires a lengthy capture session with an actor, whose time is typically limited. Subsequently, the images I_n need to be manually annotated with facial landmark positions to compute the corresponding ground-truth model states $\hat{\mathbf{S}}_n$. This process has inherent potential for inaccuracies that likely propagate to the learned model. Synthetic generation of training imagery using an actor-specific blend shape rig provides exact correspondence between I_n and $\hat{\mathbf{S}}_n$. Moreover, this allows flexibility to render a training set tailored to the actor's expression range, physical properties of a camera rig and environment lighting conditions. Such systematic customisation is highly impractical with real facial images. The tailored training prevents over-generalisation across many different conditions and yields better tracking performance for the individual. Also, this naturally limits the size of training data and therefore leads to faster learning. Reduction of training time is an important practical requirement, which we address by careful training set design.

Synthetic training set generation is based on an actor-specific blend shape rig $\mathbf{B}$. The rig is built beforehand using a high-quality offline facial capture system [5]. This provides high-resolution facial expression shapes and associated texture and normal maps. Facial appearance is derived from given blend weights $\mathbf{a}$ using local blending of texture maps for several key expressions. This approximation contains wrinkles and visible skin structure which can be relit under novel illuminations. The face rig considered in the current experimental work does not model eyes or inner mouth areas which can be considered a limitation. In addition to these obvious differences to real imagery, there exists several other sources that cause a discrepancy between real images and synthetic renders, such as sensor noise, subsurface scattering, etc. It is essential to prevent the regressor from overfitting to the synthetic render style, which we achieve by varying the appearance in the training set (discussed in Section 4.1). We utilize an advanced video game engine to render the facial rig, as real-time rasterization on GPU provides a good image quality and fast rendering. The engine allows variable placement of a virtual camera, modeled according to calibration data of a real head-mounted camera. Our virtual illumination of the facial rig consists of point lights simulating helmet-mounted illumination and an environment map models surrounding environmental lighting conditions, based on light probe data acquired at the target location. The background is rendered as white noise to ensure that the model learns the signal only from valid foreground regions (see Fig. 2).

4.1. Varying expression, viewpoint and illumination

The first semantic axis E, varying the appearance of training images, represents facial expressions. The blend weight component $\hat{\mathbf{a}}_n$ of the model state $\hat{\mathbf{S}}_n$ defines a facial shape and skin appearance for a given expression. An arbitrary number N_E of different blend weight vectors can be generated, however we chose to work only with the canonical blend shapes B_j in $\mathbf{B}$ (single, fully active weights in $\hat{\mathbf{a}}_n$ at a time). By the nature of the rig building process, these shapes represent actor range of expression well and constitute physically plausible facial expressions unlike most arbitrary points in the blendshape space. We find that restricting the expressions to B_j maintains a practical training set size whilst providing favourable expression tracking (experimental details follow in Section 5, Section 6). Example images rendered from the expression space are provided in Fig. 2.

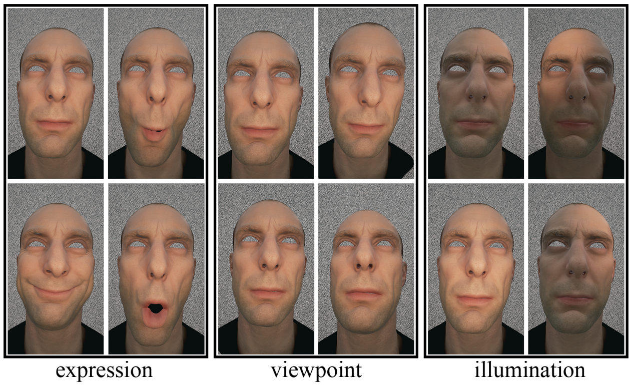


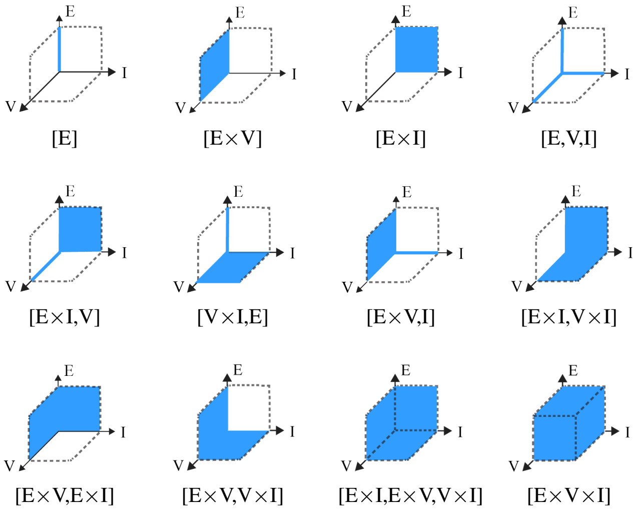
Figure 2: Synthetic training image variance - expression, camera viewpoint and illumination.

The second axis V varies camera viewpoint in training imagery. The model state $\hat{\mathbf{S}}_n$ contains a rigid transform $(\hat{\mathbf{r}}_n, \hat{\mathbf{t}}_n)$ that represents deviation from the camera rest pose. The rest pose, with respect to the face, is given by a real camera calibration together with fixed intrinsic parameters. We find that training for viewpoint variance brings robustness to camera movement at runtime. A model trained only on expression variation will incorrectly explain minor camera motion using facial expression change and training for viewpoint variance mitigates this. This is a requirement since helmet cameras may slide on the head in practice and camera-shake can become evident during rapid head motion. We analyse the motion of a real helmet camera relative to the face to determine ranges of rotation and translation with respect to the camera rest pose. Approximate anisotropic distributions over $\mathbf{r}$ and $\mathbf{t}$ are sampled uniformly to generate N_V distinct transforms.

The third axis I varies synthetic facial illumination. Note that lighting is not represented in our model state and is therefore not explicitly inferred during regression. However, different lighting conditions in the training set are necessary to handle illumination change of the face due to typical subject movement or changes of environmental lighting. Our regression framework does not extrapolate well to real data using only a single, stationary synthetic illumination. Furthermore, we find it important to derive the synthetic training illumination variance from reference lighting, defined by analysis of the target test environment. We typically capture light probe data at different locations in the capture volume and populate environment maps for rendering. Intensity and position of point lights are defined to emulate the physical helmet camera design and, by additionally varying environment map rotations, we can sample an informed set of N_I distinct illumination conditions and relight facial expressions with illumination conditions likely to occur at runtime.

4.2. Training set design

Training set construction involves sampling of $(I_n, \hat{\mathbf{S}}_n)$ from a space defined by the three semantic axes described previously. Imagery is rendered according to different combinations of blend weights, camera transforms and lighting configurations that are selected from the informed sets with sizes N_E, N_V and N_I. A naïve strategy involves generating a triple crossproduct [E×V×I] containing all possible combinations. This entails relatively expensive construction and subsequently long training times. As an alternative, we explore different design strategies that result in a significant reduction of training set sizes and, importantly, training times. The evaluated design strategies are illustrated in Fig. 3 and focus on different combinations of individual axes and the planes defined by sampling axis crossproducts. The origin of the sample space represents a facial image with a neutral expression ($\hat{\mathbf{a}} = \mathbf{0}$), the rest camera pose ($(\hat{\mathbf{r}}, \hat{\mathbf{t}}) = \textit{identity}$) and the reference illumination. We consider this a base appearance likely to occur at runtime. As a notation example, the training set design [E×I,V] contains a full crossproduct of all expressions and illumination axis samples viewed from (only) the rest camera viewpoint, combined with the neutral expression under the reference illumination observed from all viewpoint axis samples. This set design results in $(N_E \cdot N_I + N_V)$ training images.



[E] [E×V] [E×I] [E,V,I]

[E×I,V] [V×I,E] [E×V,I] [E×I,V×I]

[E×V,E×I] [E×V,V×I] [E×I,E×V,V×I] [E×V×I]

Figure 3: Different training set designs. The axes represent the three semantic dimensions - facial expression E, camera viewpoint V and illumination I.

5. Results on synthetic image data

This section reports evaluation of our real-time face tracking with synthetic test image sequences. Experiments are designed to assess systematic training set design strategies according to tracking accuracy. By generating test imagery from keyframe animations of a facial rig $\mathbf{B}$, used for tracking, we allow for per frame comparison between a regressed model state $\mathbf{S}$ and the ground truth. More concretely, a facial mesh B is reconstructed from the estimated weights $\mathbf{a}$ and transformed rigidly according to $(\mathbf{r}, \mathbf{t})$. An error, with respect to the ground truth mesh, is computed as a mean of 3D distances between corresponding vertices.

Experiments are performed on a synthetic sequence *HelmetRigSynth* which simulates performance capture with a head-mounted camera on a real set. The sequence contains large expression changes together with large movement of a helmet camera and dynamically varying illumination derived from real light probe data. Fig. 2 gives visual examples analogous to the level of appearance variance observed at test time. None of the test data is present in the training sets. We refer the reader to our supplementary video, containing the complete test sequence. The characteristics of test imagery are 720p at 60fps to match our following real data experiments Section 6.

The parameter configuration used in our experiments is as follows. Offline training: $T = 7, F = 200, U = 800, D = 5, m_E = 20, m_T = 26$; online tracking: $l_E = 10, l_T = 10, (w = 5)$. Note that temporal filtering post-processing is switched off for all quantitative evaluations. Training computation time has been measured on a standard PC; Intel Core i7 5960x (3GHz) CPU and 32GB RAM. Our framework is implemented using parallelised OpenMP C++ without the use of CUDA or other GPU specific code. Online regression takes $4.5ms$ per frame.

5.1. Informed training image axis selection

Facial expressions: As described in Section 4, a set of N_E facial expressions from the blend shape rig constitute our discretized expression axis. In the following sections we define $N_E = |\mathbf{B}| = 73$ unless otherwise stated. This simple model involves full activation of each blendshape weight a_j individually. We find that this strategy generates sufficient expression variation and allows our method to generalize well to novel plausible expressions.

Camera viewpoint: Our initial experiment investigates the importance of varying camera viewpoint during training image generation. Robustness to camera movement at runtime is necessary even when considering head-mounted cameras. Moderate head motion or helmet adjustments result in a noticeable change of viewpoint which affects tracking accuracy. A variant of test sequence *HelmetRigSynth* containing continuous camera motion, but without illumination change, is utilised in this experiment. We train two regressors on different image sets and provide error analysis for comparison. The first regressor is trained without camera motion (design [E]) and the second regressor is trained on an informed design [E×V] where the set of $N_E = 73$ expressions and $N_V = 72$ camera transforms form a full crossproduct of model states. We sample rigid camera transforms with magnitudes representative of the restricted 6 DOF motion present in our physical helmets, experimented with in practice. These result in translation, rotation sampling ranges on the order of $\sim 4cm$ and $\sim 16^\circ$ degrees respectively. Table 1 (rows 1, 2) compares regression accuracy and numerically shows that the design [E] misinterprets viewpoint changes as expressions.

Illumination: Sampling the illumination axis is hypothesized to add robustness to dynamic lighting changes at runtime caused by actor movement or environmental changes. To investigate this point we use a variant of test sequence *HelmetRigSynth* exhibiting dynamic lighting change but without camera movement. We train on three different training set designs and provide error analysis for comparison. The design [E] makes use of only a single reference lighting condition for all 73 expressions. The design $[\mathrm{E} \times \mathrm{I}]_{arb}$ utilizes $N_I = 75$ arbitrary lightning environments to relight the face. Environments are obtained from publicly available lighting databases [1, 2]. The design $[\text{E} \times \text{I}]_{probe}$ uses $N_I = 75$ different illumination conditions derived from light probe data captured on-set (the same lighting information used to derive *HelmetRigSynth*). The two training sets with illumination variance make use of full $N_E \times N_I$ crossproducts to form training samples. We find that synthetic training sets containing multiple lighting conditions perform, as expected, significantly better on test data containing illumination change. It can also be observed in Table 1 (rows 3,4,5) that tailoring training samples according to the target lighting environment has a large impact on quantitative error. This training variance proves essential for bridging the visual gap between synthetic training imagery and live test imagery, allowing accurate tracking (see Section 6).

Training set			Testing sequence appearance variance			error (mm)	std
Design	N	M	E	V	I		
[E]	73	1606	✓	✓		5.47	2.06
[E×V]	5256	241776	✓	✓		0.93	0.53
[E]	73	1606	✓		✓	1.92	2.28
$[\text{E} \times \text{I}]_{arb}$	5475	120450	✓		✓	0.62	0.63
$[\text{E} \times \text{I}]_{probe}$	5475	120450	✓		✓	0.26	0.33

Table 1: Training set designs with varying image appearance sampling and per-frame vertex error.

5.2. Exhaustive axes combination

Generating training pairs $(I_n, \hat{\mathbf{S}}_n)$ by combining individual semantic axes provides an obvious route to improving accuracy performance as observed in Section 5.1. As a further axis-combination baseline experiment we generate a full triple cross product of all expressions, camera transforms and illumination conditions (see the design [E×V×I] in Fig. 3). This naïve training set construction generates exhaustive appearance combinations, however the uniform, and uninformed, nature of construction leads to large image sets and consequently computationally expensive training. While broad training data can often aid the learned prior's ability to generalise, it does not result in a prior with the best tracking accuracy. Subsets of a tailored appearance space potentially allow for computational savings using smaller image sets whilst retaining regression precision.

Due to computational constraints, in this experiment we sample the three axes by using subsets of the discrete sample points used previously in Section 5.1. We use $N_E = N_V = N_I = 30$ which results in $30^3 = 27000$ exhaustive appearance combinations in the training set. We create training sets of various size by randomly downsampling training pairs from the full [E×V×I] according to a *uniform* distribution. Regressors learned on these training sets are tested on *HelmetRigSynth*. In order to obtain robust statistics, priors are repeatedly learned for each training image count $N \in \{90, 450, 900, 1800, 2700, 3375, 6750, 13500\}$, by randomly drawing training sets five times. Fig. 4 reports statistics of mean vertex distance averaged over the whole test sequence. It can be observed that error rates converge using priors trained on the order of several thousand images. Larger training image counts $>= 3375$ have little effect on the error that converges to $\sim 2.3mm$ and exhibits consistent variance across training set size. We include in Fig. 4 also the full training set to demonstrate lack of improved accuracy in spite of larger image count.

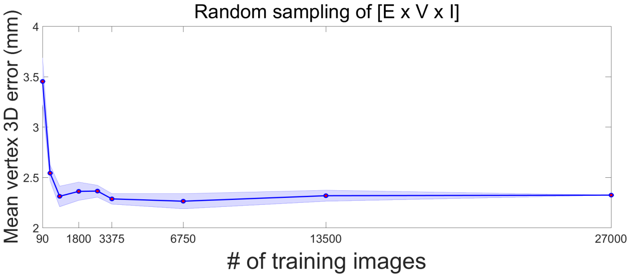


Figure 4: Random sampling of [E×V×I]. Mean error to ground-truth per training image count over the *HelmetRigSynth* test sequence.

5.3. Comparison of training set designs

Section 5.2 showed that it is possible to drastically reduce full axes combination by uniform random downsampling without losing tracking accuracy. In this section, we progress to selecting samples from the appearance space in a more informed fashion while retaining small training set size and high regression precision. We investigate non-uniform sampling by systematically combining crossproducts of different axes. Fig. 3 illustrates the considered designs representing different combinations of planar crossproducts and individual axes. The intuition here is that some axes may require a full crossproduct, but for others this could be reduced to a 1D sampling along the axis. The constructed training sets are again evaluated using the sequence *HelmetRigSynth* and resulting mean errors and standard deviations are reported in Table 2.

Dense sampling of the expression and illumination plane together with a minimal axis set of all viewpoints, [E×I,V], provides the leading quantitative performance. This design is able to reduce quantitative error by 46% in relation to strategies [E×V,I], [V×I,E] which have comparable training set size. The intuition is that two factors contributing most to facial appearance are expression and illumination. Minor variation in camera viewpoint alters the image information significantly less in comparison. Also, supersets of the design [E×I,V] such as [E×V,E×I] exhibit comparable but slightly higher error while containing many more images. This may be explained by the process of extending training set size to include less relevant image samples “diluting” the focus of the learned prior. Lastly, almost all designs achieve lower error than the exhaustive design [E×V×I] from Section 5.2. This shows that an informed design can provide better regression capabilities with a smaller image budget. We find both non-uniform sampling in the appearance space and informed selection of discretized axes sample points to be instrumental in training set construction.

Design	N	M	error (mm)	std	time (min)
[E×I,V]	5547	255162	**1.05**	**0.66**	178.2
[E×I,V×I]	10875	500250	1.09	0.66	397.4
[E×I,E×V,V×I]	16131	742026	1.10	0.66	623.2
[E×V,E×I]	10731	493626	1.11	0.69	408.0
[E×V,V×I]	10729	493534	1.59	1.17	366.6
[E×V,I]	5404	248584	1.94	1.38	212.4
[E,V,I]	292	13432	2.13	1.43	2.9
[V×I,E]	5473	251758	2.43	2.04	207.9

Table 2: Training set design comparison on synthetic test sequence *HelmetRigSynth*. Ordered according to mean vertex error. Training time per design provided in minutes.

5.4. Training set downsampling

We consider further training set size reduction by random downsampling which proved beneficial on the naïve exhaustive design from Section 5.2. The best informed design [E×I,V], found in Section 5.3, is evaluated here in terms of image count vs. regression accuracy trade-off. The training pairs are randomly drawn from the crossproduct plane E×I and the viewpoint axis is left complete. This is repeated over 5 trials for each target image count. In Fig. 5 it is observed that downsampling the training set by an order of magnitude (922 vs. 5547 images) results in a mean regression error increase of only 10%, in turn resulting in minimal visual difference in tracking. For reference the design [E,V,I], utilizing only 220 images, can be seen as an extremely downsampled version of [E×I,V]. The related training time savings resulting from this accuracy trade-off can be assessed in Fig. 6. To summarise, we find training set size reduction is possible and the best reduction found experimentally is the scheme [E×I,V]. By further reducing this synthetic training set through downsampling to just 922 images, we create an effective training set using only $\sim$ 3.5% the image count of the exhaustive [E×V×I] set, with corresponding linear training time savings.

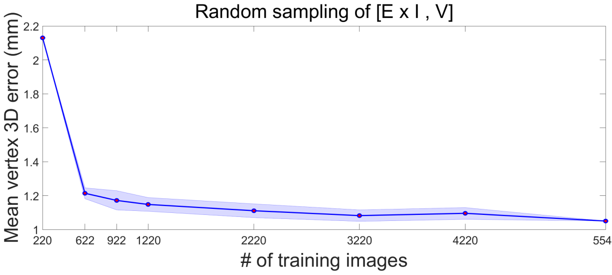


Figure 5: Random downsampling of the design [E×I,V] (5547 images). Mean and standard deviation for ground-truth error over the whole *HelmetRigSynth* test sequence.

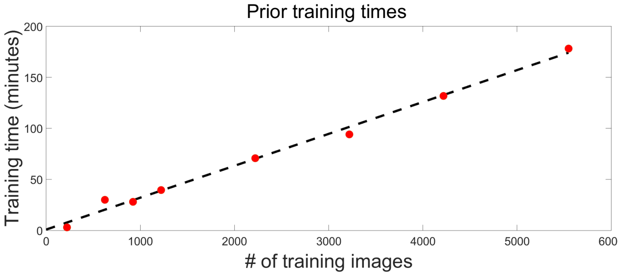


Figure 6: Linear relationship between training set size and training time demonstrated on random downsampling of the best design [E×I,V].

6. Results on real image data

We assess our approach on real test imagery and provide evidence that synthetic experimentation provides valid prediction of tracking quality on real data.

6.1. Quantitative evaluation

To evaluate performance on real imagery we obtain high-quality tracking using the offline system of [5]. This allows numerical comparisons analogous to the experiments with synthetic sequences. Fig. 7 shows an example frame from a real test sequence *StaticRig* captured using a static multi-camera system [5]. Training viewpoint workout is informed by the natural movement of the actor’s head against a head rest ($N_V = 73$). Training illumination variance is narrow ($N_I = 20$) because of fixed uniform capture lighting. We construct training sets according to the designs in Fig. 3. Tracking accuracy of our trained regressors is compared using per-frame vertex distance to the tracked meshes of [5]. Aggregate errors over time and training characteristics are shown in Table 3. The design [E×I,V] achieves the lowest error with the relatively small number of training images requiring short training time. This matches the outcome of synthetic experimentation in Table 2. There is a general positive correlation in the ranking of individual designs on synthetic and real data, confirming that our synthetic testing sequences act as reasonable barometer for accuracy on live imagery. A lack of perfect ranking correlation may be caused by disparity in the nature of the employed sequences *HelmetRigSynth* and *StaticRig* (e.g. the amount of illumination change over time).

Comparison to Beeler et al. [5] The experiment described above naturally provides a quantitative and qualitative comparison to the high-quality offline tracking technique which performs reconstruction using 7 cameras (see supplementary video). Our real-time method displays accurate facial tracking and recovers the performance in a fraction of the offline computation time ($4.5ms$ vs. $15min$ per frame).

Design	N	M	error (mm)	std
[E×I,V]	1533	88914	**4.57**	**2.26**
[V×I,E]	1533	88914	5.10	2.75
[E×I,E×V,V×I]	8249	478442	5.24	2.86
[E×I,V×I]	2920	169360	5.33	2.51
[E×V,E×I]	6789	393762	5.65	2.88
[E,V,I]	166	9628	5.80	2.73
[E×V,V×I]	6789	393762	6.37	2.72
[E×V,I]	5349	310242	7.27	3.12

Table 3: Training set design evaluation using a real test image sequence *StaticRig*. Results ordered according to mean vertex error.

6.2. Qualitative evaluation in virtual production

We evaluate fully synthetic training for real-time facial tracking using a professional virtual production scenario. Training set designs, assessed previously on synthetic data in Section 5.3, have been constructed for a live helmet camera scenario. Synthetic viewpoint and illumination training ranges are informed by the real head-mounted camera and light probes of the set volume, respectively. A test sequence *HelmetRig* depicted in Fig. 7 contains substantial changes of helmet illumination strength. The same regressors learned using the training sets in Table 2 tracked the sequence *HelmetRig* with similar qualitative results. The synthetic prior designs performing the best on *HelmetRigSynth* yield visually comparable tracking on live data, with our [E×I,V] sampling strategy resulting in the shortest training time. Further training speed-up can be provided by the variant of [E×I,V], downsampled to 922 images which is able to maintain very similar visual tracking quality. Our supplementary material exhibits real test sequences that demonstrate accurate tracking of large expression change, expressive speech and robustness to illumination change.

Comparison to Kazemi et al. [21] We qualitatively compare to a recent real-time, sparse 2D tracking method that detects a set of 68 facial landmarks in every frame. An implementation of this regression-based technique, from the publically available Dlib library [22], was employed using a regressor trained on the iBUG 300-W dataset [29]. The supplementary video shows visual comparison to 2D tracks of face rig vertices that were manually selected to correspond with the detected 2D landmarks. Our result is more stable over time, handles large expression changes more successfully and is robust to dynamic illumination change. This highlights the limitations of learning using generic facial imagery, commonly available in annotated face datasets yet not suitable for every scenario. Our relatively cheap to construct synthetic imagery, tailored to a particular target and capture conditions, can provide considerably better tracking quality.

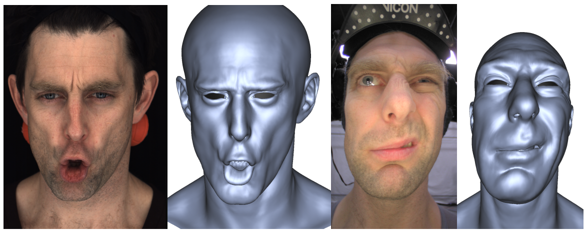


Figure 7: Example tracked frames from test sequences *StaticRig* (left) and *HelmetRig* (right).

7. Conclusion

In this paper we present an investigation of fully synthetic training for real-time, actor-specific facial tracking. To achieve high accuracy on real imagery, training data are synthesized using sets of facial expressions, camera viewpoints and lighting conditions tailored to a camera setup and capture volume illumination. We find that non-uniformly sampled training set designs are important not only for data size reduction but also achieve higher regression precision than naïve, exhaustive strategies. Experimental results show that the best design strategy can reduce training image counts by 1-2 orders of magnitude and result in proportional computational savings with no visible loss of tracking accuracy. The proposed approach presents a large step towards practical real-time markerless performance capture that allows flexible and efficient adaptation of a prior to conditions found at runtime. Our findings are applicable for guiding synthetic image generation strategies under various learning-based tracking techniques.

Acknowledgements We appreciate the help of Jose Iglesias-Guitian with video preparation and the constructive feedback of anonymous reviewers. We thank Industrial Light & Magic, London for creating actor blendshape rigs and Jethro Skinner for his acting performances. This work was funded by the InnovateUK project - Real-time Digital Acting (# 101857).

References

[1] http://www.unparent.com/photos_probes.html. Accessed: 2016-09-14. 6

[2] http://www.hdrlabs.com/sibl/archive.html. Accessed: 2016-09-14. 6

[3] B. Amberg, A. Blake, and T. Vetter. On compositional image alignment, with an application to active appearance models. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[4] A. Asthana and S. Zafeiriou. Incremental Face Alignment in the Wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[5] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 30:75:1–75:10, 2011. 1, 3, 7, 8

[6] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[7] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 32(4):40:1–40:10, 2013. 2

[8] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 29:41:1–41:10, 2010. 1

[9] C. Cao, D. Bradley, K. Zhou, and T. Beeler. Real-time high-fidelity facial performance capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 34(4):46:1–46:9, 2015. 2

[10] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 33(4):43:1–43:10, 2014. 2, 3

[11] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 32(4):41:1–41:10, 2013. 2, 3

[12] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, Mar. 2014. 2

[13] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2887–2894, 2012. 2, 3

[14] J.-X. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Symposium on Computer Animation*, 2003. 2

[15] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001. 2, 3

[16] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool. Random Forests for Real Time 3D Face Analysis. *International Journal of Computer Vision*, 2012. 2

[17] Z.-H. Feng, G. Hu, J. Kittler, B. Christmas, and X. Wu. Cascaded Collaborative Regression for Robust Facial Landmark Detection Trained using a Mixture of Synthetic and Real Images with Dynamic Weighting. *IEEE Trans. on Image Processing*, 7149(c):1–1, 2015. 2

[18] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 32(6):1–10, nov 2013. 1

[19] P. Garrido, M. Zollhoefer, D. Casas, L. Valgaerts, K. Varanasi, P. Perez, and C. Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph.*, 35(3):28:1–28:15, 2016. 2

[20] A. Jeni, J. F. Cohn, and T. Kanade. Dense 3D Face Alignment from 2D Videos in Real-Time. In *Face and Gesture*, 2015. 2

[21] V. Kazemi and S. Josephine. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 3, 8

[22] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 8

[23] M. Klaudiny and A. Hilton. High-detail 3d capture and non-sequential alignment of facial performance. In *3DIMPVT*, 2012. 1

[24] V. Le, J. Brandt, Z. Lin, L. Boudev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[25] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 32(4):42:1–42:10, 2013. 2

[26] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 2

[27] E. Ong and R. Bowden. Robust facial feature tracking using shape-constrained multiresolution-selected linear predictors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1844–1859, 2011. 2

[28] S. Ren, X. Cao, Y. Wei, and J. Sun. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[29] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshops*, 2013. 2, 8

[30] J. M. Saragih, S. Lucey, and J. F. Cohn. Face alignment through subspace constrained mean-shifts. In *International Conference on Computer Vision (ICCV)*, pages 1034–1041, 2009. 2

[31] F. Shi, H.-T. Wu, X. Tong, and J. Chai. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graphics*, 33(6):1–13, nov 2014. 1

[32] 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), 2015. 2

[33] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[34] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under un-

controlled lighting. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 31(6), 2012. 1

[35] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 30(4):77:1–77:10, 2011. 2

[36] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *Symposium on Computer Animation*, pages 7–16, 2009. 2

[37] Y. Weng, C. Cao, Q. Hou, and K. Zhou. Real-time facial animation on mobile devices. *Graphical Models*, 76:172–179, 2014. 2, 3

[38] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. In *Automatic Face Gesture Recognition*, pages 1–6, 2008. 2

[39] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2013. 2

[40] S. Zhu, C. Li, C. Change, and X. Tang. Face Alignment by Coarse-to-Fine Shape Searching. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2