Interactive Exploration and Refinement of Facial Expression using Manifold Learning

Rinat Adbrazhitov† Fanny Chevalier‡ Karan Singh†
† Department of Computer Science, ‡ Department of Statistical Sciences
University of Toronto
{rinat | fanny | karan}@dgp.toronto.edu

Figure 1. Interacting with high-dimensional face parameters using a variety of devices, is a challenging task (a), often resulting in implausible facial expressions: the eyelids inter-penetrate and the neck muscles are overstretched in (b). We address this problem by interactively projecting user created expressions onto a subspace of natural or stylized expressions, called a face manifold (c), learned using an autoencoder from existing facial animation data. Manifold projection enables rapid user exploration of the rich multidimensional subspace of expressive faces (d). These exploratory expressions can then be further adjusted interactively via a 2D embedding (e) that facilitates incremental refinement of expression on and off the manifold (f).

ABSTRACT
Posing expressive 3D faces is extremely challenging. Typical facial rigs have upwards of 30 controllable parameters, that while anatomically meaningful, are hard to use due to redundancy of expression, unrealistic configurations, and many semantic and stylistic correlations between the parameters. We propose a novel interface for rapid exploration and refinement of static facial expressions, based on a data-driven face manifold of “natural” expressions. Rapidly explored face configurations are interactively projected onto this manifold of meaningful expressions. These expressions can then be refined using a 2D embedding of nearby faces, both on and off the manifold. Our validation is fourfold: we show expressive face creation using various devices; we verify that our learnt manifold transcends its training face, to expressively control very different faces; we perform a crowd-sourced study to evaluate the quality of manifold face expressions; and we report on a usability study that shows our approach is an effective interactive tool to author facial expression.

Author Keywords
3D face modeling; blendshape; manifold learning.

INTRODUCTION
Emulating the complex expression conveyed by a human face, on digital 3D models, is a difficult task for actors and animators alike. Subtle mistakes in facial nuance can plunge an animated character into the Uncanny Valley, where the audience loses trust and empathy with the character [40]. The importance of realistic CG human faces, however, cannot be overstated. A diverse set of applications ranging from entertainment, medicine, education, and avatars, are all empowered by the ability to model, pose, and simulate human faces.

The facial rig is often comprised of a large (typically 30+) set of target blendshapes representing various facial configurations, for eg. those of the Facial Action Coding System (FACS) [17, 34]. Facial expressions result from a user controlled weight interpolation of these targets. The onus of manipulating these weights to author complex facial expressions, however, remains tedious in the hands of the user.

Individual blendshapes while semantically meaningful, often exhibit correlations, conflicts and significant redundancy. Even identifying a constituent subset of active blendshapes that model an expression, can be ambiguous and difficult. Sequentially experimenting with such a parameter space through independently operated weight sliders is particularly painful since only a small non-linear subspace of blendshape weights correspond to meaningful facial expressions. This makes unconstrained exploration of all blendshape weight configurations prone to generating implausible expressions. It is useful to restrict exploration to the subspace of plausible face configurations only. This allows users with limited understanding of blendshapes, face anatomy, or facial animation experience to produce complex expressions faster and easier.
Real-world high dimensional data (such as images or blendshape weights) lie on a low-dimensional subspace embedded in the high-dimensional space. In the data science literature that low-dimensional subspace is referred to as a manifold. We do not presume any user expertise in facial anatomy or animation. Instead, we capture this domain expertise as a manifold of “natural” (realistic or stylized) faces, as represented by a training corpus of sample expressive faces. Our face manifold is learnt from the corpus using a denoising autoencoder, and encoded by the latent space of the network [51]. Restricting interactive manipulated faces to this subspace or manifold of plausible configurations (see Fig. 1) addresses the above parameter space problems, and keeps facial expressions consistent for a given character. Current face manipulation interfaces ranging from a simple array of weight sliders, to image or video based performance capture [26, 52], direct manipulation [35], and sketch-based posing of faces [22], can all benefit from our approach by minimally altering their output to project onto a desired face manifold (see Fig. 1(a-c)).

Limiting faces to always lie within the potentially incomplete space of a data-driven manifold however, can be over constraining. We therefore allow iterative refinement of expression, by sampling the neighbourhood of an exploratory face, both on and off the manifold to create an interactive 2D embedding of nearby faces, with qualitative feedback on their distance to the face manifold (see Fig. 1(e)). The refined face can be used to produce subsequent neighbourhood embeddings, as well as bookmark specific faces for future use. A collection of expressive face bookmarks can be added to the data corpus to re-train the face manifold, bootstrapping the construction of a character face manifold.

Our implementation of manifold construction, projection and refinement is shown on a variety of inputs such as a face camera, game controller, and traditional sliders. Our manifold trained on a FACS-like rig, is also shown to be face agnostic, i.e. the learnt manifold of blendshape weights can meaningfully control face geometries quite different from the one used in training. We use a crowd-sourced study to evaluate the quality of the manifold in capturing expressive faces, and its impact on fixing unnatural faces. Finally, we perform a 10-user usability study that shows our approach to be an effective interactive tool to author facial expression, that compares favorably to the interactive control of independent face parameters.

Our main contributions are: 1) A data-driven model to capture a subspace of natural face expressions, and an operator to project faces onto this manifold; 2) A demonstration that interactive manifold projection can enable the rapid exploration of meaningful facial expressions by untrained novices; 3) A refinement interface to interactively manipulate the high-dimensional face space in the vicinity of an expression, both on and off the manifold.

RELATED WORK

We review frameworks and tools for creating facial expressions (Fig. 2.), and discuss prior work on denoising methods, and systems for the exploration of large dimensional spaces.

### Animating 3D face geometry

Blendshape rigs are arguably the most popular choice for realistic or stylized character animation [34]. Each blendshape is a target face, such as a smile, pucker, or squint, representing the result of facial action(s) on a neutral face. A facial expression or pose is the weighted linear combination of these blendshapes. Traditional face posing interfaces in software like Blender or Autodesk Maya®, are thus simple arrays of sliders and widgets, for animators to control blendshape weights (Fig. 3). Individually and sequentially manipulating blendshape weights, and even locating a desired target in the widget array is tedious. Over time animators thus memorize the function and layout of hundreds of sliders to efficiently pose a face. Professional 3D animation unsurprisingly requires about one hour of labor per second of animation [35].

### Face posing systems

Performance or motion capture systems use the motion of a human actor to drive a 3D face model [53]. These methods track 3D markers on an actor’s face via a multi-camera system, and compute time-varying blendshape weights that match the moving markers, to drive the animated expression of a synthetic 3D face [6, 16, 59]. High-end performance capture typically requires substantial imaging hardware and manual clean-up of the resulting face animation. Approaches based on ordinary web cameras [9, 19, 26] or low-cost depth cameras [7, 37, 38, 52] have been proposed to overcome cumbersome motion capture setup. These methods are easier to deploy, but produce coarser results and can be non-trivial to map to arbi-
try 3D face geometry. Motion capture methods are strongly dependent on camera hardware and actor performance. They also do not inherently restrict geometry to lie within any subspace of meaningful expressions, and can thus benefit from our work as a post projective step (Fig. 9a).

Sketch-based, and direct manipulation techniques have been proposed to overcome the tedium of sequentially controlling individual sliders. Face Poser [31] combines user input with a facial prior learned from a prerecorded facial expression database. Such systems use screen-space 2D point, stroke and curve constraints to pose a 3D facial model. Sketch Express [43] is a facial sketching interface based on drawing simple strokes in predefined 2D drawing regions. Direct manipulation methods [35] pose 3D faces by manipulating points on the geometry and inverse computing blendshape weights to match the points. Similarly, the motion of user specified points can be minimized, when optimizing blendshape weights to mitigate self-collisions and geometric mangling due to blendshape interference [36]. 2D portrait manipulation interfaces [48] map the manipulation of predefined curves of a 2D portrait, to drive 3D marker-based facial expressions. These approaches map user interaction to a subspace of plausible facial expressions using geometric and collision constraints. While these approaches are better than unconstrained manipulation, they tend to be face geometry specific and are unable to capture non-geometric or stylistic constraints. Our data-driven manifold approach addresses both these issues, and integrates well with existing animation workflows.

Face Manifolds
Face manifolds have been explored in various 2D contexts based on 2D landmarks or image pixel data [11, 12, 13, 45, 55], and based on the geometric sampling of the 3D facial surface in 3D [15]. In contrast, our approach is blendshape based: an animation-data-driven, sparse, and compact subspace of blendshape weights that capture facial expression.

Denoising autoencoders
Machine learning denoising methods are increasingly applied in computer graphics for a variety of purposes. Denoising is an unsupervised learning technique used to train autoencoders to detect structure in the given input [51]. Denoising autoencoders have been applied to problems such as in-painting [57, 58], denoising Monte-Carlo renderings [3, 10] and fixing corrupted animation data [23, 41, 54].

Our work is inspired by recent advances in skeletal-based animation [24, 25], that use convolutional autoencoders to learn a manifold from a motion capture database. Projecting invalid or corrupt data onto this manifold can fix motion errors and reconstruct natural movement. Additionally, high-level character constraints such as a locomotion trajectory or the target of a punch, can be mapped to the motion manifold, to produce motion that is both goal-directed and natural. We use a similar approach, utilizing denoising autoencoders to learn a manifold of static facial expressions.

Exploring of high-dimensional spaces
Exploring high-dimensional design spaces for creative authoring tasks is difficult. The baseline approach, is still to manipu-

late a collection of uni-dimensional sliders. Smart Suggestion and VisOpt Slider interfaces [29] use a goodness function, to interactively visualize the distribution of goodness in the design space. The goodness function is obtained via crowdsourcing pairwise comparisons between different parameter configurations. High-quality 3D models can be created by estimating the distribution of good models in a design space, where goodness is based on tracking the modeling activity of a distributed community of users [49]. In contrast, we directly learn the space of natural facial expressions from pre-existing datasets, an approach that is cost-effective and scales well to high dimensional parameter spaces.

Design galleries present suggestions, to iteratively guide users towards desired parameter configurations, by asking them to rank or choose from multiple candidate configurations [20, 33, 8, 60]. In contrast, we want the user to be able to rapidly explore the face space interactively and experiment with, or further refine their choices.

FACE MODEL
Creating a particular facial expression usually involves the manipulation of a face model — a reference face rig featuring a neutral expression that can be modified using a set of user controls, which are typically blendshape weights. Mathematically, such a face model is a simple vector sum as follows:

$$f = b_0 + \sum_{k=0}^{n} w_k (b_k - b_0)$$ (1)

where $f \in \mathbb{R}^n$ is the resulting face, $b_0$ is the neutral face, and $b_k \in \mathbb{R}^n$ are target blendshapes with weights $w_k \in \mathbb{R}$, (usually $[0, 1]$), that modify $b_0$.

Blendshapes are user intuitive but non-orthogonal, and sparse representation. This means that different blendshapes may be redundant in their action as well as conflict with one another, for eg. activating a “squint” and “eyebrows up” simultaneously (Fig. 5(right)). It is thus important to restrict exploration to the subspace of plausible face configurations, which we call the face manifold (Fig. 4).

MANIFOLD LEARNING
Our definition of realistic or stylized “natural” expressions is a subspace or manifold based on an input corpus of posed faces. Therefore, the degree of naturalness and plausibility is fully determined by the training data. Formally, given a facial expression $x \in \mathbb{R}^n$ where $x_i$ is the weight of each of
While our dataset contains a large number and variety of facial expressions, it is imbalanced as some expressions appear more often than others. For example, close-to-neutral expressions greatly outnumber smiling expressions. Imbalanced datasets can result in bias and overfitting, where the network learns to perform well on one set of expressions and ignore others. We mitigate this problem by first running a $k$-means algorithm [39] to separate the data into clusters. We force all clusters to have a similar number of samples by oversampling the minority clusters and undersampling the majority clusters, using the Synthetic Minority Over-sampling Technique (SMOTE) [14]. To undersample a cluster, we randomly pick samples to be removed. We use a balanced set of 10,000 samples to train our face manifold.

**Denoising Autoencoder for Learning Motion Data**

Autoencoders exploit the idea that the data is concentrated around a low-dimensional manifold or a small set of manifolds, and aim to learn the structure of those manifolds [21]. Autoencoders impose a bottleneck at the deepest hidden layer which forces a compressed knowledge representation of the original input, to capture any structural correlations in the training data. Denoising autoencoders are an extension of classical autoencoders but are exposed to partial corruption of the input pattern during training, to better handle noisy device input in our context.

We train 2 denoising autoencoders, one for the top and one for the bottom half of the face. The input size for the top is 10 dimensional and for the bottom is 21. Each autoencoder contains 3 hidden layers with different number of units (top: 7, 4, 7; bottom: 12, 7, 12). For a single hidden layer $k$, filter weights $u_k$ and biases $t_k$, the projection operation from input vector $x$ to hidden units is given by:

$$\Phi_k(x) = ReLU(u_k \cdot x + t_k)$$  \hspace{1cm} (2)

except for the last layer, where we use a sigmoid activation function (Fig. 6). The number of hidden layers and the size of each hidden layer in the autoencoder was determined during training by choosing the best combination that produced the smallest loss error (Eq. 3) on the validation data.

**Training**

Denoising autoencoders are typically trained to retrieve an original input, given an altered version of such input. To train our denoising autoencoder, we corrupt input vectors $x$ to a new vector $\hat{x}$, by adding Gaussian noise, and the network is trained to reproduce the original input $x$. We find network parameters $\theta = (u, t)$ for each layer such that the following loss function measuring the squared reproduction error and the $L_1$ norm of the output vector of activations ($\hat{x}$) is minimized. The $L_1$ norm term acts as a regularizer to ensure that the final output of the blendshapes weight is sparse. In other words we want the output expressions to be achieved by activating the least number of blendshapes as possible [34]. The amount of sparsity is controlled by the $\lambda$ parameter (0.1 in our experiments).

$$Loss(x) = ||x - \hat{\Phi}(\Phi(x))||_2^2 + \lambda \sum_i |\hat{x}_i|$$  \hspace{1cm} (3)
Overfitting can be a concern when the training dataset size is not large. While the size of our dataset is comparable to other training datasets described in learning literature [50, 2], we would like our approach to be applicable to smaller or evolving datasets of character animation data.

We address overfitting by: balancing the dataset to avoid over-representation of certain expressions; training two separate autoencoders for the top and bottom face, that have a lower dimensionality than a single autoencoder for the full face (Fig. 6); and using a Dropout [47] of 0.2 during training.

Design choices
Face manifolds tend to be non-convex in shape and capture complex non-linear relationships (Fig. 7) between blendshapes. Autoencoders provide a more powerful representation for such structure than baseline approaches such as PCA (Principal Component Analysis) (Fig. 8). Variational autoencoders learn the parameters of a probability distribution representing the training data. This learned distribution can then be sampled to produce novel data but is not well suited to our goal of denoising the corrupted input. We experimented with general neural networks and autoencoders (without Gaussian noise) but given that we want our approach to be applicable to a variety of interfaces and device inputs, we found denoising autoencoders to produce the smallest reconstruction error and show the most resilience to imperfect input.

FACE EXPLORATION
Given a face manifold trained on pre-existing data, we can project any face input onto the manifold. Faces may be posed using input from sources such as web cameras, hand or face trackers, game controllers, or a traditional weight slider based interface. The subspace of meaningful faces is small relative to space of all configurations and projecting input onto this manifold keeps a user from getting lost in a sea of meaningless and mangled faces. We experimented with 3 manifold projection augmented interfaces: a face camera, traditional sliders, and a gamepad controller.

Slider interface
Slider-based interfaces are dominant in facial animation practice. We augment such interfaces by automatically applying our manifold projection operator to ensure that users never leave the subspace of natural facial expression (Fig. 9). This enables rapid exploration on the manifold and also aids novices in understanding blendshape correlations when posing faces. For example, increasing “Jaw forward”, also visibly increases “Jaw drop” upon manifold projection, to capture the natural behavior of an open jaw. The current projected values used to pose the face are shown beside the slider, and as red bars in the slider background.

Camera interface
There is extensive research in analyzing facial expressions from from images [42]. OpenFace [4, 5] is able to estimate values for 15 blendshapes from an image of a face. Directly using these blendshape weights to pose 3D faces does not always produce plausible results since the outcome depends on quality of the image, lighting conditions, quality of the data set used for training, and the robustness of analysis algorithm. We apply our manifold projection technique to produce more natural facial expressions and fix corrupt input (Fig. 9).
Gamepad controller

Gamepad controllers can simultaneously control multiple blendshape weights to pose avatar faces in games. Figure 9 shows an example layout for controlling blendshapes using a gamepad. Simultaneous control of many blendshapes is fast but more susceptible to producing unnatural faces (see accompanying video), which are readily corrected by manifold projection. Our approach thus enables the use of a wider range of device input for posing faces.

FACE REFINEMENT

Most approaches in facial animation solely encompass a face exploration stage (without correction by a manifold). However, regardless of the input method used, and even with our correction through projection onto the manifold, face posing via manipulation of blendshape weights remains imprecise due to the complexity of combining multiple blendshapes. After posing the face, the user may find that the result is close to an envisioned facial expression, but may still require fine-tuning to fit their artistic vision. We thus propose a novel interface to face refinement by learning a latent space embedding only within the vicinity of an explored face. This embedding allows users to visualize and interactively refine the current facial expression or produce new highly exaggerated expressions.

Latent space embedding

We create a 2D latent space around a given facial expression where similar expressions can be embedded and intuitively explored. More formally, given a facial expression \( x_0 \), we generate a set of similar expressions \([x_1, x_2, x_3, \ldots, x_k]\) where \( x_i \in \mathbb{R}^n \). A naive approach to generating the set of similar expressions is to sample the subspace around a given expression. However, since face expressions are high dimensional, where \( x_0 \in \mathbb{R}^n \) with possibly \( n > 30 \), a lot of samples would need to be produced and most of them would not look natural. Instead, we sample the neighbourhood following these 3 rules:

1. We identify the list of activated blendshapes whose weight is above a certain threshold \( \tau = 0.15 \). We uniformly sample along each activated blendshape axis, such that if \( i \) is an index of the blendshape verifying \( x_i > \tau \), then the sampling boundary is defined as \( [x_i - \tau, x_i + \tau] \). We repeat the sampling for each activated blendshape individually to produce a set of samples \( S_1 \). This ensures that users will be able to refine their facial pose along each activated blendshape individually.

2. We build a set \( S_2 \) by uniformly sampling along all activated blendshape axes at the same time. This ensures that we have samples where multiple blendshape weights are varying.

3. We look at the set of previously user-created faces which we call bookmarks. We find a set of bookmarked faces that are within a small threshold away from the given facial expression to produce a set \( S_3 \). Adding bookmarks allows users to effortlessly bring previous faces that they considered plausible and quickly generate similar results.

The final set of neighbours \( S \) is the union of sets \( S_1 \cup S_2 \cup S_3 \). To enable interactive exploration of this space, we require an adequate representation that both conveys information to the user, while supporting easy navigation. An obvious solution consists of mapping this result space on a 2D representation that can be visualized and interacted with. Non-linear manifold learning techniques come as a natural choice for this kind of problem. Gaussian process latent variable models (GPLVM)\[32] is a probabilistic dimensionality reduction method that is able to find a lower dimensional non-linear embedding of high dimensional data.

Our projection operator can be used to measure the distance between the facial expression and the face manifold to determine how plausible the facial expression is. This enables an informative visualization of the latent space where the distance to the facial manifold presented in the form of color coding and highlights regions of the latent space that move far away or lie closer to the manifold (Fig. 10).

Interactive refinement in latent space

Given the 2D latent space visualized as a distance to manifold color map, we enable interactive navigation through dragging a pointer inside the 2D space to generate new latent coordinates which are projected back to the high dimensional blendshape space. This output is visualized dynamically, which provides real-time feedback supporting efficient exploration of the neighbourhood of the initial facial pose (Fig. 10).

To further guide the user in their exploration, we provide an additional visual aid in the form of images of the neighbouring faces. These images are shown when the current face position is clicked on, and aim to give an intuitive cue on how the face would change if the user were to move in that direction, guiding them the right way to obtain the desired pose. The location of each image is determined by sampling the neighbourhood along 5 different directions and determining the point along each direction that is the furthest distance away from the current pose. Above each image we show the name of the muscle that changes the most when moving to that area from the current position. If the user finds a plausible face but wants to see additional options, they can click the re-refine button to generate a new latent space from this face.

Design Choices

While the face manifold can be trained offline on a large dataset, the face refinement interface needs to be able to genera-
ate its latent space “on the fly” with little data of only the neighbouring faces (sets 51-3). Therefore, deep learning approaches used to build the face manifold are not suitable. GPLVM meets various criteria needed for our refinement interface: creates a non-linear mapping between 2D latent space and the blendshape space, requires little data (about 10 faces), requires short training time (seconds) to quickly re-refine around a chosen expression or add new custom expressions, interpolates between the embedded expressions for continuous exploration. PCA is not well suited to model the non-linear mapping from the very low dimensional latent space to the high dimensional blendshape space and other non-linear dimensionality techniques like Eigenmaps, t-SNE, LLE are less suited to interpolating between the embedded expressions.

MANIFOLD QUALITY ASSESSMENT
The quality of the produced manifold can be evaluated in multiple ways: a “good” manifold should be expressive, exclude unnatural faces, and generalize to a variety of rigs.

Expressiveness. Our manifold captures the expressive range of the given face, as represented by the input data. A limited dataset can unintentionally leave out meaningful expressions, or intentionally impose an expressive style on a face since the learned manifold is highly representative of the input data. For example, our training data is missing expressions where the *buccinator* muscle—the major facial muscle underlying the cheek that aids whistling and smiling—is highly activated. When applying the projection, the resulting face is closer to a neutral expression (see Fig. 11). Incomplete manifolds can always be improved with a richer data-set, or by adding novel expressions created by our refinement interface to the original training data and recomputing a more expressive manifold.

Unnatural faces. Another quality measure is the representation of undesired or unnatural faces by the manifold. Our manifold is trained by minimizing the error difference between facial expressions in the input data and their corrupted counterparts. We comprehensively evaluated the presence of unnatural faces in the manifold and the quality of the projection operator via a crowd-sourced study on Amazon Mturk (40 participants, excluding subjects who failed the understanding test below). Each participant was shown a different set of 100 images of facial expressions, one image at a time, and asked (yes/no) if the face looked natural (example faces were shown to provide meaning to the word “natural”). Images of facial expressions were generated as follows:

- **Random sample set**: 35 randomly sampled faces. Each expression \( f \in \mathbb{R}^n \) ws generated by uniformly sampling the \( n \) dimensional face space.

Table 1 shows a summary of the study results. Of the 1,400 random samples (40 participants presented with a different set of 35 random faces), only a few were found to be natural. In contrast, after projection on the manifold, these random faces were largely perceived as natural, suggesting that our projection operator finds the closest natural expression with an average distance in blendshape space of 1.8 between to original face, and its projected counterpart (Fig. 12a,b). Further, the faces from the random sample set that were found natural, were also found natural after projection, with an average distance of 0.24 between the pairs (Fig. 12c,d). This suggests that while our manifold is a good proxy for natural faces, there are also meaningful facial expressions that lie off the manifold, which can be reached by our refinement interface.

<table>
<thead>
<tr>
<th></th>
<th>Random set</th>
<th>Projection set</th>
<th>Projection set that became natural after projection</th>
<th>Random set that remained natural after projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of natural faces</td>
<td>8% (112/1400)</td>
<td>87% (1218/1400)</td>
<td>89% (1146/1288)</td>
<td>98% (109/112)</td>
</tr>
</tbody>
</table>

Table 1. Results of the crowd-sourced Amazon Mturk study evaluating the quality of the manifold based on the unnatural faces criteria.

Generality. Our manifold was trained on the animation data for the character in Figure 10. However, as long as the source and target rigs have corresponding/similar blendshape semantics, face manifold does generalize well across different human rigs (see Fig. 1, 8, 9, 13 and accompanying video).
**INTERFACE EVALUATION**

We conducted a study with participants who had a limited understanding of blendshapes, face anatomy, or any facial animation experience, in order to evaluate the potential of our approach in aiding such users in posing facial expressions. The study lasted 1 hour and we compensated participants with a $25 Amazon gift card.

**Apparatus**

We use the Pytorch Python library to perform data preprocessing tasks and train our autoencoder model. Once the model is trained it is serialized and ready to be loaded in a C++ application. The main exploration interface was written in C++/OpenGL using ImGui for the UI components. A separate Python-based application showed the refinement UI (the 2D latent space). Both applications were run on the same machine simultaneously, and utilized web sockets for intercommunication. The system ran on a computer with Intel Xeon 16 core 3.5Ghz CPU, GTX1080 GPU and 64GB of RAM with monitor at a 3840x2160 resolution. The participants used a mouse and a keyboard to interact with the interface.

**Participants**

Ten participants (6 female, 4 male; Age: 23-29) took part in our study: 5 graduate students and 5 participants who have full-time jobs not related to animation. Six participants reported having some knowledge about computer animation techniques but no experience with facial animation specifically.

**Study protocol**

Our study aimed to evaluate three aspects of our interface:  
**A1.** Discoverability: how easy it is for a user to use the interface with minimal to no guidance.  
**A2.** Usability: how effective it is to pose face expressions.  
**A3.** Expressivity: how hard it is to pose desired, diverse and meaningful expressions.

To this effect, we designed two tasks. **Task 1** compares two conditions: the projection+refinement condition, using our slider based interface with manifold projection and the refinement interface together, and the baseline condition, using the traditional slider interface without manifold projection. **Task 2** was an open-ended exploration task, where we only study the projection+refinement condition.

**Task 1** focused on **A1** and **A2**. Participants were given a reference facial expression (Fig.14(left)), and instructed to produce an accurate replica. Participants were not instructed to perform the task as fast as possible, but were given a time limit (3min) to perform the task. They could also submit their work before the countdown when satisfied with the result. Four different faces $f_1, f_2, f_3, f_4$ were shown to each participant as follows: $f_1, f_2$ were part of the training data set and belong to the manifold; $f_3, f_4$ were not present in the training set and were a small distance ($0.2$) off the manifold (Fig. 14). For every pair ($f_i, f_j$), the distance $d(f_i, f_j)$ was greater than a diversity threshold $1.0$. Each participant was randomly assigned the condition order of projection+refinement $\rightarrow$ baseline, or baseline $\rightarrow$ projection+refinement. The order of presentation of the faces ($f_i$) was generated randomly for each participant. Participants completed the task with the first condition, and then again with the other condition. We chose to use the same order of presentation of faces for each condition to maximize the number of trials between two same facial expressions, to mitigate any learning/memory effects. Each condition was prefaced with a 3-5 min training session for that interface. We administered a questionnaire after each condition using a 5-point Likert scale (1—strongly disagree, 5—strongly agree).

**Task 2** focused on **A3** using the projection+refinement condition only. There are 6 universal emotions: happiness, sadness, anger, surprise, fear, and disgust. Participants were each randomly given two emotions, and instructed to generate two different variations of each emotion within 5 min. The participant could generate a new face either from the last recorded face, or starting from a neutral expression. Figure 16 shows some of the user created faces.

**Results**

Overall, all participants successfully completed the tasks and created facial expressions within the time limit.

**Task 1 – Quantitative results**

Figure 15 (left) shows the mean distance error from each of the target faces $f_i$ and overall, per condition. Point estimate and 95% confidence intervals were computed using bootstrapping.
The higher values mean higher error, i.e. higher distance from the target face. The right figure contrasts our approach with the baseline: point estimates and intervals are estimates of the difference of mean distance error between the baseline and our method, computed for each participant (this is possible given our within-subjects design). In this figure, positive values mean that our method outperforms the baseline.

On average the participants were able to get closer to the target faces using our projection slider interface (Fig. 15, Table 2) with a smaller number of sliders being activated (Table 3). This can be explained by the manifold activating related muscles without the user needing to explicitly drag the sliders and manually activating them. The refinement interface further allowed participants to fine tune expressions closer to the target. Additionally, participants were able to reach the coarse expression resembling the target face (distance to target ≤ 1.0) faster with our projection slider interface (Table 4).

Discoverability
Participants found the system easy to use (4 strongly agree, 6 agree) and learn (5 strongly agree, 3 agree, 1 neutral, 1 disagree), and that the various functions in the system were well integrated (3 strongly agree, 5 agree, 2 neutral). As P5 noted, the "learning curve was not hard" and interacting with the system "felt like playing a game". P1 mentioned that they can see themselves "spending a lot of time playing with the system" and that the "lack of previous experience in facial animation didn’t cause any problems". Most participants found the interface “fun to use” (P1, P2, P3, P5, P7, P8).

Usability and Expressivity
Most participants felt that the interface allowed them to create the expression they wanted (4 strongly agree, 5 agree, 1 neutral). A common theme among users was that the projection slider interface allowed them to create a rough expression matching the target quicker and easier than the baseline (P1, P2, P3, P4, P5, P7, P10). P5 liked that the interface "adjusted related muscles for you, so you don’t have to think afterwards about additional muscles you need to activate" while P1 mentioned that there was "less tweaking involved" and that "they didn’t have to go through all the sliders to get the result."

The refinement interface proved to be useful for fine tuning the facial expression with 9/10 participants agreeing that it helped them create better expressions. P4 and P5 agreed that the refinement interface "gives you a lot of options that you didn’t think of before" and that those options were "better than what they originally created". The distance to manifold color coding received mixed feedback with 5/10 users saying that they factored the color coding into their final choice and that they found faces closer to manifold more natural. 5/10 users did not use the color coding or used it every other time and preferred just exploring the the whole space and didn’t mind choosing faces that were further away from the manifold. Majority of the participants (8/10) found images of neighbouring faces useful and often relied on them to navigate the latent space. Sometimes the images seemed similar and users mentioned that they relied on the label (indicating the main muscle that was changing).

The biggest contrast between the projection slider interface and the baseline was creation of unnatural faces when changing slider values. Users found that they were often creating unnatural faces with baseline interface (1 strongly agree, 6 agree, 2 neutral, 1 disagree) compared to manifold interface (2 strongly disagree, 7 disagree, 1 agree). While talking about the posing process, P3 mentioned that manifold projection "makes the slider adjustment feel more natural".

Slider responsiveness
Most participants (8/10) found that the projection slider interface was more difficult to use once they created a rough expression of the face and they wanted to make small isolated changes. They often found that making a small change to one muscle would change other muscles as well that they didn’t necessarily want. P4 mentioned that "some faces were harder to make than the others" and it would have been "helpful to move sliders in isolation". This was the most common feedback for faces $f_3$, $f_4$ that are not on the manifold.
Suggestions for Improvement

Study participants gave useful feedback on how we can improve our system. Some (P3,P4,P6,P7) mentioned that they wanted to be able to turn off projection for the isolated control of the individual muscles after posing the face. P5 suggested to add a toggle key-bind to the refinement interface to quickly switch between the current face and the original face to see how the face changed/improved. Some users (P2,P4,P8) wanted to see a more extensive list of muscles that are changing when moving to certain areas of the latent space instead of only knowing the one muscle that changes the most.

DISCUSSION

Our approach to the authoring and interactive control of facial expressions differs from prior work in two main ways. First, we provide dynamic auto-correction of unnatural faces into similar but meaningful facial expressions through projection onto a face manifold. Second, we support finer control of facial expression by interacting with a dynamically computed manifold that captures the neighborhood of the currently explored face.

Sense of control. Getting a strong sense of control is critical to creative authoring [46]. Constraining the user/artist to be on the space of the learnt manifold does take away an element of control over the output. During our evaluation however, participants commented on only wanting isolated control of the weights after coarse posing was done.

Our goal is to support users in their creative endeavor, but such support comes at the cost of taking away some decisions. There is a sweet spot between providing enough assistance so that the tool is actually helpful, and too much assistance that can take away sense of control and in some cases, sense of ownership of the created artifact. Prior work supporting guided authoring of sketches for novices, such as sketch-sketch revolution [18] and PortraitSketch [56] also discuss these trade-offs. In our case, however, the user can always refine the face to lie off the manifold.

Expressiveness. Our approach is limited by the quality and richness of the data used to build the manifold. While it can be seen as a constraint (and it certainly is in some cases), this also makes it possible for the user to use a specific dataset as a way to intentionally bias the manifold to generate expressions that conform to a particular style.

During an informal evaluation of our approach with a senior professional animator, he pointed out that “large-scale animation productions commonly impose a character style guide to ensure the integrity of a character authored by different animators”. He referred to this as animating “on-model” or “off-model” in industry parlance, and said it is precisely what our concept of “on/off-manifold” captures.

User Demographics. Our tool for interactive exploration on and around a face manifold serves two goals: first, manifold projection allows users to pose faces quicker, i.e. manipulating a few parameters is enough for remaining parameters to automatically conform to produce an expressive face; second, the manifold itself captures the desired range of expression or “on-model” for a character’s face. While expert animators certainly benefit from visualizing how their posed faces relate to the given “on-model” character style, they are already adept at posing nuanced and expressive faces quickly, and sometimes prefer to control all face parameters independently. We chose to formally evaluate our approach with novices and animation enthusiasts, who have little or no understanding of facial anatomy, to better measure the impact both on the quality of a posed face, as well the time speed-up that our tool offers.

We informally evaluated our approach with a professional animator. He provided a mix of high-level and detailed feedback. His overall feeling though was very positive. While he felt that our approach might not improve the quality of his work, or help him steer clear of unrealistic faces (he knows his rigs and its realistic subspace), it would certainly make him more efficient (exploring coarse expression rapidly). It would also allow him to work more effectively and uniformly with other animators on the same character, because the manifold would provide them with a common expressive baseline.

Limitations. The biggest limitation of our work is the reliance of our approach on a meaningful facial manifold. In the absence of a good representative data set of example face configurations, manifold projection will constrain the exploration to a small subspace of natural faces. Our definition of a “natural” expression is fully determined by the training data. To our knowledge, no open-source database of a large number of facial animations based only on FACS exists yet. Restricting the dataset to a single stylized character can introduce biases and force the manifold to adhere to the stylistic limitations. Ideally, to learn a general manifold, data should come from multiple characters with different styles. This would enrich a space of “natural” expressions and potentially improve the sense of control. However, there is risk of introducing conflicts between blendshape correlations which could result in the network outputting the average of different styles. This is a subject for future work.

Aside from using pre-existing animation data, the corpus can be built by showing crowd-sourced users random face configurations and asking them to accept or reject them as realistic. Importantly however, manifolds largely generalize across human faces and it may be easier to use existing face manifolds as a baseline for novel characters.

Another limitation is the lack of a guarantee that our refinement technique will precisely embed a desired facial expression. The embedding is however, constructed using a judicious sampling of the neighbourhood of a face including faces that are previously bookmarked, and those that lie both on and off the manifold. Users can further search this neighbourhood by incrementally re-embedding a refined face.

Given the importance of static expressive faces for applications like representative freeze or key-frames for film and animation, advertising billboards, custom emoji’s and profiles for avatars in social media and games, the modeling of facial prosthetics, this paper remained focused on static facial expressions. An extension of this work to a dynamic or animated face manifold is subject to future work.
CONCLUSION
We have presented a workflow to address the interactive exploration and refinement of static facial expressions. Our overall contribution is a workflow for face posing that uses a manifold learnt from existing facial expression data to constrain a user’s input, and a subsequent 2D interactive embedding of the neighbourhood of a face to refine its expression. This allows users with no previous experience in facial animation to produce natural faces poses with less effort. Our approach integrates seamlessly with a variety of high-dimensional input devices such as cameras, sliders and game controllers. Our evaluation shows the manifold to be a sound representation of the subspace of plausible faces, that can generalize across humanoid face models that are different from the training dataset. Finally, a user study shows our approach to face manifold exploration and refinement to compare favourably to independent slider manipulation. Faces are crucial to human visual communication and we see our approach to provide a novel and compelling workflow to aid users in understanding and authoring expressive 3D digital faces.

ACKNOWLEDGEMENTS
We wish to thank Chris Landreth for providing the data as well as facial rigs, the participants for their time and reviewers for their valuable feedback which helped improve this work.

REFERENCES


