A Learning-Based Approach to Parametric Rotoscoping of Multi-Shape Systems

Luis Bermudez
Intel Corporation
luis.bermudez@intel.com

Nadine Dabby
Intel Corporation
ndabby@gmail.com

Yingxi Adelle Lin
Intel Corporation
adelle.lin@gmail.com

Sara Hilmarsdottir
Intel Corporation
madame.sara@gmail.com

Narayan Sundararajan
Intel Corporation
narayan.sundararajan@intel.com

Swarnendu Kar
Intel Corporation
swarnendu.kar@intel.com

Abstract

Rotoscoping of facial features is often an integral part of Visual Effects post-production, where the parametric contours created by artists need to be highly detailed, consist of multiple interacting components, and involve significant manual supervision. Yet those assets are usually discarded after compositing and hardly reused. In this paper, we present the first methodology to learn from these assets. With only a few manually rotoscoped shots, we identify and extract semantically consistent and task specific landmark points and re-vectorize the roto shapes based on these landmarks. We then train two separate models – one to predict landmarks based on a rough crop of the face region, and the other to predict the roto shapes using only the inferred landmarks from the first model. In preliminary production testing, 26% of shots rotoscoped using our tool were able to be used with no adjustment, and another 47% were able to be used with minor adjustments. This represents a significant time savings for the studio, as artists are able to rotoscope almost 73% of their shots with no manual rotoscoping and some spline adjustment. This paper presents a novel application of machine learning to professional interactive rotoscoping, a methodology to convert unstructured roto shapes into a self-annotated, trainable dataset that can be harnessed to make accurate predictions on future shots of a similar object, and a limited dataset of rotoscoped multi-shape fine feature systems from a real film production.

1. Introduction

Rotoscoping is a technique used to obtain object segments from a video that can subsequently be extracted for use in compositing or further editing in the visual effects post-processing pipeline. These object segments must be spatially and temporally smooth in order for compositing artifacts to be imperceptible to the human eye, and they must be parametric so that an artist can iteratively modify them until the composite meets their rigorous standard. The high standards and stringent requirements of state-of-the-art rotoscoping still requires a significant amount of manual labor from live action and animated films.

We apply machine learning to automate rotoscoping in the post-production operations of stop-motion filmmaking, in collaboration with LAIKA, a professional animation studio. LAIKA has a unique facial animation process in which puppet expressions are comprised of multiple 3D printed face parts that are snapped onto and off of the puppet over the course of a shot, leaving unattended *seams* and *chatter* that must be removed in post-processing. We present a methodology to help accelerate the seam removal and holdout mask creation process. The architecture of our AI-enabled rotoscoping solution is described in Figure 1. The rotoscoping task in this example is to find and create roto shapes around eyes, eyebrows and visible seams.The subsequent compositing task is to repair the seams while keeping the eyes and eyebrows intact.

Our methodology upends the need for additional annotations beyond what artists already generate as part of their compositing task. We confront the practical problem of how to repurpose these one-time use roto-assets that were created without machine learning in mind. Additionally, we are able to integrate this methodology and our trained AI models directly into the roto artist's existing Nuke workflow, such that the artists are able to interact with and manipulate the intermediate and final inferred outputs of the models.

In the VFX segmentation field real data sets are hard to come by and difficult to share due to intellectual property constraints. Although we tested our method on multiple tasks, we are only able to publicly share the results of one character task to demonstrate our method. In addition to the method we present here, we are releasing our train-

ing dataset (consisting of 5 shots) as well as a test dataset consisting of 200 frames [3]. We believe this is the first dataset of rotoscoped multi-shape fine feature systems to come from a real film production to be released to the community. Missing Link movie and character images copyright 2019-2020 Laika, LLC and licensed under CC BY-ND 4.0: https://creativecommons.org/licenses/by-nd/4.0/

Although we demonstrate the idea of end-to-end roto learning for a specific stop-motion animation task, this methodology is directly and generally applicable to any visual effects operation that requires a multitude of interacting, consistent objects to be rotoscoped. While we highlight our methods for one character, the studio has used our techniques for many new characters and new shapes (e.g. mouth, nose, ears). We hope that this work will motivate other researchers in the community to consider in-production roto assets as sources of supervision for future VFX machine learning work.

2. Background

2.1. Related Work

Many existing rotoscoping tools are semi-automated: they allow the artist to label a few frames of an image sequence, and then the tool infers the labels for the rest of the frames. The target region is usually labelled by creating the region's boundary with a shape tool. Researchers have proposed ample approaches to rotoscoping [8, 25, 2, 19, 14]. Similar to other rotoscoping work [33, 34], we also incorporate our rotoscoping solution into the artists' existing interaction flows; however, we do not require them to do any manual rotoscoping in real time.

In this work, we do not explicitly address smoothness between frames. However, tracking is a common technique that is leveraged by artists to maintain spatio-temporally consistent predictions between frames [33, 1, 21], and our tool allows the use of tracking after inference is performed.

There is a significant overlap between our work and single image inferences such as semantic segmentation and facial landmark predictions in the computer vision community. The goal of semantic segmentation is to detect the pixels that represent semantic regions, such as humans, trees, bicycles, and more [13, 12, 32, 23, 5, 17, 31]. Our semantic focus is to detect facial features, such as eyes, eyebrows, puppet seams, and mouths. While we share a similar goal with most semantic segmentation work, we do not perform per-pixel inference. Instead, we leverage facial landmark predictions [28, 29, 38, 36, 39, 26]. Traditional facial landmark predictors used cascade regressors [15], but they have been surpassed by CNN methods and heatmap estimations [28, 38]. After we complete landmark predictions, we leverage the landmarks to run regression methods [22, 35, 6, 30] and output the semantic shapes that we are interested in [4]. Facial landmark prediction research has also been conducted on videos [7].

The work described in [19, 24, 20] predict shapes based on landmarks and tangents. The work in [9, 37, 11] predict shapes based on pixel predictions or correspondence. While [19] does not make image aware predictions, our work and that presented by [24, 20] base predictions on the image as input. The predicted points are not semantic in [24, 20] while the landmarks in our work are semantic with respect to the object in the image. Additionally, our work is the only one that predicts a system of shapes simultaneously.

2.2. Design Considerations

Data Efficiency: The raw stop-motion footage generated by the studio requires a considerable amount of post-processing work to remove seams and other artifacts from the puppet faces in order to transform the footage into a film. Training a key point model normally takes a dataset with millions of images [28, 16, 18]. Task specific annotation is another major hurdle, and if the task changes slightly, much of the annotation may have to be redone. While pre-trained facial landmark detection models (trained on thousands of human faces) exist, those models do not generalize to puppet faces. Thus a major design consideration for solving this problem is to maximize the performance of a deep learning model in a data efficient way.

Domain Matched Data: In the rotoscoping domain, a good output shape is not necessarily determined by the mean point-to-point error of facial landmarks inferred by our tool. Rather, success is determined by how closely the AI-generated shapes would match a professional artist's work on a particular character, or how much time an artist may save by using this tool. With that in mind, we decided to focus on training models for individual characters as opposed to training a general model that would apply to all characters, but perform less well on all of them. We use synthetically generated data and data augmentation to complement a few shots of artist-generated rotoscoped data for a particular puppet character, to train a model that can then be used to assist in the labeling of many more raw shots of that character. With as few as 5 professionally rotoscoped shots for a particular character, we can assist in the rotoscoping of hundreds of additional shots featuring the same character.

Interactivity and Vector Shapes: Our main goal in this work was to create an AI-inference tool that would reduce the manual work of roto post-processing. In order to meet this goal we made the conscious choice to integrate our inference algorithms directly into the artist's workflow instead of building a separate stand-alone tool. Our trained character models can be imported directly into Nuke via a plugin and, as a result, the artist does not have to disturb their workflow to gain the benefit of using our AI models.

Much of the previous academic work in the area of rotoscoping has focused on algorithms to the detriment of readily usable output (a notable exception is the work presented in [19]). For a tool to be adopted by the roto artist, it should output shapes in a format that can be easily manipulated.

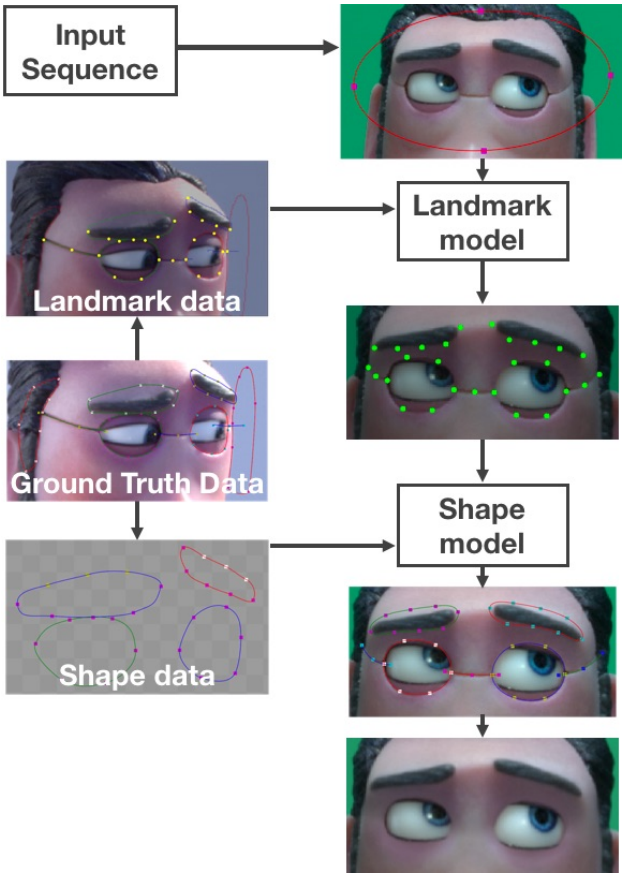

Figure 1. An overview of our solution illustrates how we mine the ground truth dataset for landmark points (used to train the Landmark model) and shapes (used to train the Shape model). A sequence of images is input into our tool, accompanied by a rough crop of the face, and fed into the Landmark model. The resulting inferred landmark points are next fed into the Shape model and the roto shapes are output.

3. Technical Approach

3.1. Overview

Our solution (Figure 1) has the following steps:

(1) The input to the system is a raw dataset containing the image $\mathcal{I}$ of the object and unstructured multipart rotoshapes $\mathcal{S}_{raw}$. This dataset is not yet usable for training. Details about our training dataset can be found in Supplementary Section 1 or the project page [3].

(2) A series of feature engineering steps are performed on this raw data. Landmark points are extracted based on morphological and geometric operations on $\mathcal{S}_{raw}$, we denote this set of points $\mathcal{P}_{all}$ (see Section 3.2). $\mathcal{S}_{raw}$ is then re-vectorized into a canonical shape (denoted $\mathcal{S}$) by using points from $\mathcal{P}_{all}$ and regressing for tangents. Note that $\mathcal{S}$ (comprised of points in $\mathcal{P}_{all}$ and their tangents to the shape) are self-annotated from $\mathcal{S}_{raw}$. No new annotation cost is incurred.

(3) Even though $\mathcal{I} \rightarrow \mathcal{S}$ is now trainable, in the next step we deliberately introduce the semantically consistent subset of landmarks $\mathcal{P} \in \mathcal{P}_{all}$ into the learning chain: $\mathcal{I} \rightarrow \mathcal{P} \rightarrow \mathcal{S}$. Rather than building a black box inference system that cannot be tweaked, we insert these landmark points into the process as they allow the user to assist the system in correcting mis-predicted points that would down the line result in a bad shape. The roto artist can easily manipulate the inferred points $\mathcal{P}$ before the system outputs the inferred roto shapes. Thus instead of learning $E[\mathcal{S}|\mathcal{I}]$, the system instead learns $E[\mathcal{P}|\mathcal{I}]$ and $E[\mathcal{S}|\mathcal{P}]$ separately for the sake of predictability and interactivity. This provides early intervention opportunities to the artist with fewer errors being propagated down the chain in the case of mispredictions due to covariate shift.

(4) Finally, a landmark model $E[\mathcal{P}|\mathcal{I}]$ and shape models $E[\mathcal{S}|\mathcal{P}]$ for each roto shape are trained.

3.2. Feature Extraction

We extract landmark features from existing rotoscoped shots. This enables the inexpensive collection of domain-matched data and forms a crucial part of our solution. As part of the rotoscoping task, an artist creates vector shapes as in Figure 2. We call these "ground truth shapes". Even though an artist uses several points and tangents to create and manipulate these shapes, the point spacing, ordering and numbering are largely arbitrary and inconsistent from shot to shot.

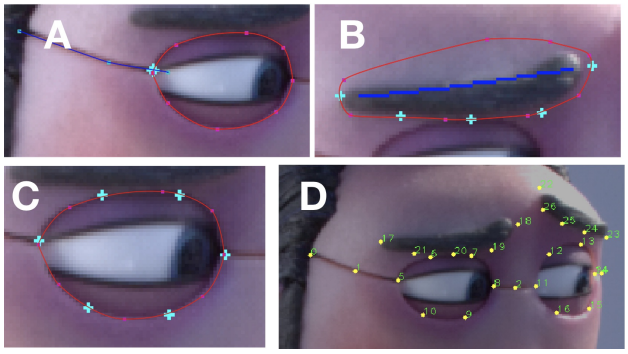

Figure 2. Landmark extraction process from roto shapes to landmark points. (A) Corner points are defined by the intersection of two curves. (B) An eyebrow shape is skeletonized to define its axis. (C) Corner points are registered and the remaining landmark points are created along the contour with uniform spacing. (D) The extraction process results in a set of semantically consistent points.

We use a combination of morphological and geometric techniques to derive landmark points from these ground truth shapes as follows: (1) Corner points are defined by the intersection between two shapes, (e.g., between seams and eyes), as shown in Figure 2A. (2) For objects with oblong shapes (e.g., eyebrows) we first rasterize and skeletonize the shape. The intersection of the axis with the original shape induces a pair of extreme points. We define a few points in the middle based on equal spacing along the shape contour (Figure 2B). (3) For other solid objects (e.g., eyes), we use corner points to form an axis. The vector shape is then rasterized and traversed in a clockwise direction, whereby equidistant points along the contour are further defined as shown in Figure 2C.

Note that roto shapes often appear imperfect because certain portions of the shapes may not have been relevant to the eventual compositing goal. Consequently, a subset of extracted landmarks may be visually inconsistent, that is, they cannot be defined from any visible corners or edges in the image (e.g., the top portions of eyebrows as shown in Figure 2B). We discard these points.

3.3. Landmark Prediction

3.3.1 Face region localization

The artist creates a garbage matte (rough crop) over the region of interest and animates it over all the relevant frames in a shot. We derive a coarse crop of size 224×224 from this garbage matte. We require the region of interest to be fully inside this crop and roughly within 50%-90% of the maximum possible scale. We trained our model to be robust to this imprecision in order to enable fast face localization.

3.3.2 Model

We use a straightforward extension of the ResNet50 model [10] to directly regress on the landmark positions within the 224×224 crop. We removed the final classification layer the network and replace it with a densely connected layer and a landmark output layer. Alternative architectures can potentially be adopted for this purpose with similar modifications.

3.3.3 Loss function

Our custom loss function for all n points is a generalized version of Mean Absolute Error

$$\mathcal{L}_{\text{MAE}} = \sum_n \varepsilon_n, \varepsilon_n = \Big|\big|\hat{x}_n - x_n^{\text{GT}}\big| + \big|\hat{y}_n - y_n^{\text{GT}}\big|\Big|, \quad (1)$$

that also takes into account the unique features of our dataset. Here $\hat{x}_n, \hat{y}_n$ denote predicted points and $x_n^{\text{GT}}, y_n^{\text{GT}}$ are the corresponding ground truth points. Due to the mixed nature of our dataset, which includes both synthetically generated data and actual rotoscoped data, our loss function must handle potential inconsistencies in labeling methods and utilize all possible metadata available. In particular, we include three additional weights per point to equation (1).

$$\mathcal{L}_{\text{roto}} = \sum_n w_{\text{valid},n} w_{\text{part},n} w_{\text{point},n} \varepsilon_n \quad (2)$$

A validity parameter, $w_{\text{valid},n} \in \{0, 1\}$, allows us to handle diverse data in which some labeled points are occluded at particular rendered angles. We can pass whether or not a point is valid into the loss function with this parameter. Thus the ε_n of the occluded points will be multiplied by 0, and their contribution to the loss calculation will not be backpropagated during training.

We also include a point-specific scaling parameter, $w_{\text{point},n}$, that allows us to handle the relative importance of individual points that are more semantically consistent across poses and other conditions, by conferring additional weight to them (e.g., the corner points at the intersection of the eyes and seams).

$w_{\text{part},n} = \frac{1}{\max(\text{scl}_n, s_n^0)}$ allows us to (1) impart scale and pose invariance to landmark point deviations and (2) handle the relative importance of shapes. This part-specific scaling parameter is assigned by feature and is used to weight the loss function by group. This method of scaling is similar to using the inter-ocular distance to normalize the error in facial landmark detection when training is performed on faces of varying scale, but differs in that we utilized distinct scaling factors for each shape on which we are training. scl_n is a scalar variable that signifies the size of the shape to which the point n belongs. s_n^0 is a scalar that describes the minimum shape weighting factor and is introduced in order to avoid over-weighting points in smaller ground truth shapes and to improve numerical stability. $w_{\text{part},n}$ is particularly useful in training the model to accept a range of input crops that allows the model to be more robust to the artist's input.

For training details and software optimizations that we performed, see Supplementary Section 2.

3.4. Shape Regression

Our shape models take the landmark points inferred from the landmark prediction model as input and output shape predictions. Figure 3 illustrates the steps involved in our methodology. A typical roto shape from an artist's project is depicted in Figure 3A. Even though the shapes are parametric, i.e., defined in terms of both points and tangents, these parameters have no fixed definition with respect to the image other than the fact that the contour it induces encapsulates the object. In the absence of a fixed definition, it is impossible to learn directly from these vector shapes.

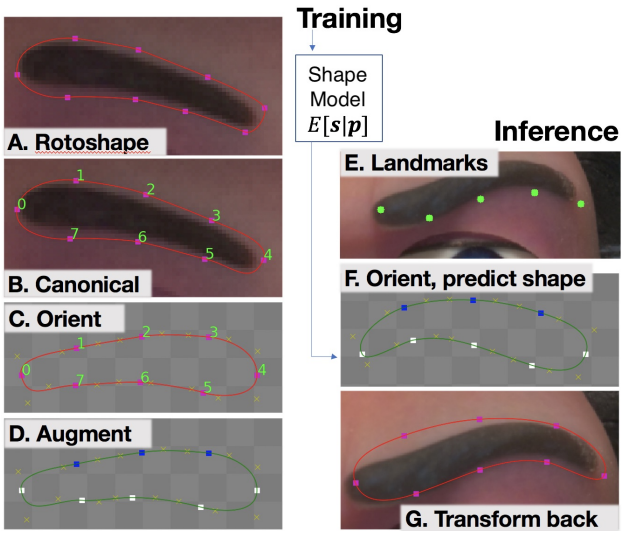

Figure 3. Shape regression method: (A) From the rotoshape, (B) computes the canonical points. (C) Applies preprocessing transformations on the canonical points for location and scale invariance. (D) Applies shape augmentation. (E) Based on landmarks, (F) predict the tangents (and any remaining landmarks) using our model, and (G) transform the shape back.

3.4.1 Canonical representation

In order to utilize a roto shape we must first derive the canonical re-parameterization of that shape based on landmarks identified earlier in Section 4.2. We define additional points as they are needed to complete the shape. For example, in Figure 3E, the 4 landmark points along the lower edge of the eyebrow are insufficient to describe the eyebrow shape, so additional points along the top of the shape are defined. Again in Figure 3B, points $(0, 7, 6, 5, 4)$ are semantically consistent landmark points for an eyebrow across all of the image data, so we define points $(1, 2, 3)$ at this stage to complete the shape in a consistent way. Once the point selection is complete, we compute the tangents as follows: (1) **Rough tangents:** We regress on individual segments to find the control points $\mathbf{c}_1, \mathbf{c}_2$ and latent variables $\{\alpha_m\}_{m=1}^{M}$ such that the contour points can be parameterized as $\mathbf{c}(\alpha) = (1-\alpha)^3\mathbf{c}_0 + 3(1-\alpha)^2\alpha\mathbf{c}_1 + 3(1-\alpha)\alpha^2 c_2 + \alpha^3\mathbf{c}_3$, where $\alpha \in [0, 1]$ is the monotonically increasing stretch parameter and $\mathbf{c}_0, \mathbf{c}_3$ are end points which are fixed in our case. The latent variables $\{\alpha_m\}$ for selected contour points are assigned uniformly spaced values during initialization and are updated in an Expectation-Maximization loop until convergence. (2) **Finalize directions:** For points where smoothness is desired, the adjacent tangents are projected onto the line with minimal projection error while guaranteeing that these projections are in opposing directions. We treat the direction of tangents after this stage as frozen. (3) **Finalize tangents:** For tangents that were tweaked during Step 2 above, we apply a final regression to the *magnitude* of the tangent and leave the end points and tangent directions unaltered. Notationally, this is the same as Step 1 with regression performed on β_1, β_2 where $\mathbf{c}_i \leftarrow \beta_i \mathbf{c}_i / ||\mathbf{c}_i||$. All three steps above are computations that solve convex problems, which result in a robust estimation of the parameters.

3.4.2 Preprocessing

In order to preprocess the canonical shapes for location and scale invariance, the shapes undergo an isotropic motion transformation based on aligning a couple of *axis* points. Optionally, if two shapes are equivalent with respect to a reflection (e.g., right and left eyes) one is flipped during this step. For example in Figure 3C, the right eyebrow shape from Figure 3B is flipped and oriented so that points $(0, 4)$ are interchanged and mapped to pre-defined points.

3.4.3 Shape Augmentation

Even though we train our landmark model $\mathbf{E}[P|\mathcal{I}]$ to predict points that are uniformly spaced along the contour connecting two axis points, we must consider two practical issues, (1) there are errors in landmark prediction which deviate from the uniform spacing assumption, and (2) even after an artist interactively manipulates the landmark predictions, we have observed that these points, while guaranteed to lie along the contour, can remain non-uniformly spaced. These two practical considerations result in a shift between training and domain distributions that we can account for with an additional *shape augmentation* step prior to training. We apply two kinds of shape augmentation to the shape training data: (1) random tweaking of points along the contour, up to 30% of the length of the segments on either side of the points, followed by re-fitting of Bézier curves on both sides of the points and 2) random affine transformations of small magnitudes.

3.4.4 Shape Models

We evaluated three strategies for shape completion with increasing levels of complexity.

Cusping: For shapes where all the points are available (e.g., eyes), we try cusping as a baseline technique. This converts a piecewise linear curve to a piecewise cubic spline with tangential continuity. Here, interpolating splines are fit on X and Y coordinates separately using normalized cumulative pairwise distance $d_k = \sum_{i=2}^{k} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$, $\tilde{d}_1 = 0, \tilde{d}_k = d_k / d_K$ as the common interpolating parameter. To achieve closed shapes and smoothness at the ends, splines are computed on stacked set of points, $\mathbf{p}_{\text{stack}} = [\mathbf{p}_1, \ldots \mathbf{p}_K, \mathbf{p}_1, \ldots \mathbf{p}_K]$ and only the middle segment is retained. The splines are typecast to Bézier after computing the tangents from the spline derivatives. For shapes where

other points are needed to complete the shape (e.g., eyebrows), cusping is not an option. Note that cusping is a static technique which is not influenced by or learned from data.

For data driven predictions using linear and neural-network models, we denote the Bézier shape as

$$\mathbf{S} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_{K_0} & \cdots & \mathbf{p}_K \\ \mathbf{t}_{l,0} & \mathbf{t}_{l,1} & \cdots & \mathbf{t}_{l,K_0} & \cdots & \mathbf{t}_{l,K} \\ \mathbf{t}_{r,0} & \mathbf{t}_{r,1} & \cdots & \mathbf{t}_{r,K_0} & \cdots & \mathbf{t}_{r,K} \end{bmatrix}, \quad (3)$$

where $\mathbf{p}_k = (x_k, y_k)$ are 2D points, $\mathbf{t}_{l,k}, \mathbf{t}_{r,k}$ are corresponding left and right tangents, K_0 is the number of known points (a subset of landmark points) and $K \geq K_0$ is the total number of points in the complete shape. We want to model the conditional estimate $\mathbf{E}[\mathbf{S}|\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{K_0}]$ through a function $\mathbf{y} = F(\mathbf{x})$, where $\mathbf{x} \triangleq [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{K_0}]$ is a vector of known parameters and $\mathbf{y}$ is a vector of the remaining parameters in $\mathbf{S}$.

Linear: A linear model for $F(\cdot)$ with only $2K_0$ regressors - (x, y) for K_0 points was found to be inadequate. However we found a featurized version, with $K_0(K_0-1)/2$ pairwise distances among the shape points added to the list of regressors, to have better predictive performance. We refer to this as the linear model in comparisons below.

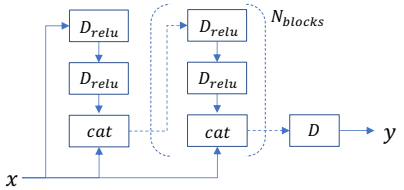

Figure 4. Neural network model for predicting shapes from landmarks.

Neural network: We also try the more expressive model in Figure 4, where D_{relu} and D stands for dense linear layer with and without rectified-linear activation, and cat refers to concatenation of variables. Here $N_{blocks} = 3$ and width$(D_{relu}) = 100$ are model hyperparameters. This model is motivated by the relative success of the featurized linear model and designed to have ample skip connections to encourage polynomial calculations among the regressors. The loss function is constructed to focus on tangent endpoint errors (normalized with respect to the ground truth) for semantically consistent points with a small L_1-norm contribution from the overall shape deviation,

$$\mathcal{L}_{\text{NN}} = \mathcal{L}_{\text{EPE}} + \gamma \left\| \mathbf{S}^{\text{GT}} - \hat{\mathbf{S}} \right\|_1, \text{ where} \quad (4)$$

$$\mathcal{L}_{\text{EPE}} = \sum_{k=1}^{K_0} \sum_{\mathbf{t}_l, \mathbf{t}_r} \frac{\left\| t_k^{\text{GT}} - \hat{\mathbf{t}}_k \right\|}{\max\left(\left\| t_k^{\text{GT}} \right\|, \epsilon \right)}. \quad (5)$$

Here $\epsilon = 0.01$ is used for numerical stability and $\gamma = 0.1$ is used to weigh the relative contribution of L_1 shape deviation.

4. Experimental results

Our training dataset consists of 1438 frames across 5 annotated shots. The test dataset consists of 240 annotated frames across 12 shots (from the same film but different from the training shots), balanced across 4 classes (face poses: Up, Down, Left and Right) with 60 frames in each class. The over-all result of using our method can be seen in Figure 5, which shows the raw images, and the roto shapes and seams output by our method. We ran several experiments on the test dataset.

4.1. Shape Regression Results

In order to evaluate our shape learning methodology, we choose the average tangent endpoint error $\mathcal{L}_{\text{EPE}}$ (Equation 5) as our metric for comparison. We use all the points for eyes and only the bottom points for the eyebrows in the $\mathcal{L}_{\text{EPE}}$ computation. We process the test dataset to obtain canonical equivalents (Section 3.4.1) so that we can measure the error on both canonical shapes (where middle points are equivalently spaced along the contour) and raw artist tweaked shapes (where there is no such guarantee). The four techniques we compare here are 1) cusping, which applies only to eyes 2) a linear model with no data augmentation 3) a linear model with data augmentation and 4) a neural network model with data augmentation.

	eye	eye-can	brow	brow-can
cusp	0.23	0.24		
lin-noaug	0.48	0.13	0.61	0.26
lin-aug	0.31	0.30	0.20	0.26
nn-aug	0.16	0.15	0.17	0.22

Table 1. Our neural network shape regression model has a lower error than the linear model or the cusping model. Additionally, the use of shape augmentation decreases the error. The error metric used in the table is the average deviation of the inferred tangent relative to the ground truth tangent (Equation 5).

The results (Table 1) clearly reveal the need for *shape augmentation* (Section 3.4.3) in training the shape models. For prediction on canonical shapes only, the linear model seems adequate, but does not perform well when the contour points are not uniformly distributed. The neural network model performs better. Some examples of shape prediction are presented in Supplementary Figure 3.

4.2. Landmark Prediction Results

In order to evaluate the facial landmark results, we computed the average Mean Absolute Error (MAE) of inferred landmark points relative to their respective points on an artist's defined shape. We also calculate the contribution

Figure 5. Top: Raw images of puppet faces with seams. Bottom: Roto shapes for seams and hold-out masks inferred by our method.

to total MAE of each type of shape (e.g. seams, eyes, eyebrows, and total) across various experimental conditions pertaining to key design choices relevant to our methodology: (1) data efficiency, (2) the use of synthetic data, (3) image data augmentation and a scale-aware loss function. We summarize the results below. For a full discussion of the results with accompanying data and visualizations, see Supplementary Section 3.

Data Efficiency: Supplementary Figures 4, 5 and Table 2 show that we have significantly increased the effectiveness of one shot of ground truth data by adding synthetic data and data augmentations. A model trained only on raw, unaugmented ground truth data is not able to fit the landmark points onto the correct facial features, nor is it able to scale appropriately. Adding synthetic data or augmentations alone significantly improves the results.

Given synthetic data and augmentations, but no ground truth data, the model outperforms the case in which the model is trained on one shot of augmented ground truth data without synthetic data (MAE of 42.64 versus 60.26). These results suggest that our rendered synthetic data is approximately equivalent to a single shot of ground truth data. When trained with a single shot of augmented ground truth data without synthetic data, the model's inferred results are significantly improved 10-fold over the raw, unaugmented single shot scenario, from a total MAE of 602.22 to 60.26. Training a model on a single shot of ground truth data with both synthetic data and augmentations results in a more than two-fold and three-fold improvement in the total MAE as compared with training on synthetic data alone (from a total MAE of 42.64 to 17.31) or on one shot of augmented ground truth data alone (from a total MAE of 60.26 to 17.31), respectively. Taken as a whole, adding synthetic data, augmentations, and a scale-aware loss function to only 1 shot of ground truth data improves performance by almost 35X over training on raw data alone (from a total MAE of 602.22 to 17.31).

We can clearly observe the improvement of model performance when trained with each additional set of ground truth data, and that beyond 2 shots, the effect on the MAE of each additional shot decreases (see Supplementary Figures 6, 7 and Table 3). While one shot of ground truth data has a significant effect on minimizing the domain gap between synthetic data and real data (e.g. the total MAE decreases from 42.64 to 17.31), each additional shot results in a decremental reduction of the total MAE, by comparison.

Synthetic Data: We conclude from Supplementary Figure 7 that more training data results in better performance as long as it increases the variance in the data. As more data is added to the training set, the model's performance starts to plateau. The synthetic data only uses 8 facial expressions with many camera angles. These results lead us to hypothesize that if we add more facial expressions to the synthetic data then it would increase variability and decrease error up to a point. Our results illustrate that synthetic data can be a very powerful tool when the *domain gap* is addressed.

Augmentations: We can qualitatively observe that models trained with background and occlusion augmentations

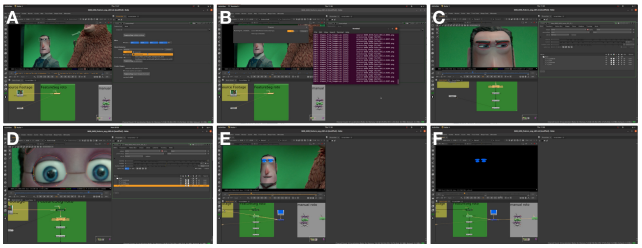

Figure 6. Tool Workflow: (A) A shot is imported into Nuke, a rough crop of the face is placed on the frames and a model is selected for landmark inference. (B) The plug-in calls an external subprocess that runs inferencing on the shot. (C) Landmark points and initial shapes are output across all frames. (D) The points in each shape can be displayed and tweaked, and a revised shape output. (E) The final splines are overlayed on the face. (F) We can also visualize the final roto matte. These finished shapes can be exported to a JSON file and used to further train the model.

perform better when inferencing on character images that contain small occlusions such as hats (see Supplementary Figures 8, 9 and Supplementary Table 4). Additionally, we can observe that using background augmentation decreases the error. However, adding the backgrounds to the synthetic data had a much more significant effect on model performance than adding backgrounds to the ground truth data. Background augmentation only adds randomization and variance to the background of the ground truth data, whereas it provides synthetic data a background where none existed before, thus greatly reducing the domain gap, and not just the variance in data.

We did not explore the effect of other augmentations such as flips, rotations, warping, motion, hue, color as they are much more commonly used and applied.

Scale-Aware Loss function Adding point and part-specific scaling improves performance by 12.3% over ignoring the scale in the loss function (see Supplementary Figure 10 and Table 4). The majority of the benefit is absorbed by the seams and eyebrows (which result in an average MAE of 2.95 and 2.67 when the model is not trained with scaling parameters, respectively, versus 2.45 and 2.15 respectively when the model is trained with scaling parameters), while the eyes do not appear to benefit. This could be due to greater invariance of the eye shape to pose and facial expression. The seams also benefit from point scaling because we can overweight their impact on the loss function.

5. Artist Workflow and Tool Interaction

Our Nuke node (Figure 6) is interactive on three levels: (1) The input to the node is a rough crop created by the artist. (2) The node performs inference on this crop and provides intermediate inferred output in the form of landmark points, which the artist can use as-is or adjust. (3) When the artist is satisfied with the positions of the key-points, these are then used to generate the output of the tool–Bézier splines of the shapes, which the artist can also adjust. See Supplementary Section 4 for a detailed discussion of the workflow.

In preliminary production testing, 26% of shots rotoscoped using our tool were able to be used with no adjustment, and another 47% were able to be used with minor adjustments. This represents a significant time savings for the studio, as artists are able to rotoscope almost 73% of their shots with no manual work and some spline adjustment.

A more detailed discussion of the production aspects of this work can be found in [27].

6. Future work

Broader Applications: We believe that our method can be applied to many more fine rotoscoping problems such as facial aging and de-aging, removing or adding scars, adding stains and injuries to a face, augmentations for aliens and deformed characters and other sequential fine rotoscoping tasks. When applied on human faces, these techniques may require more sophisticated methods to bridge the domain gap beyond our addition of synthetic data.

Contour-Aware Loss Function: We hope to better utilize the labeled ground truth data to make our loss function contour-aware. We can mine the rotoscoped shapes for tangent data pertaining to each labeled point, for example, we could store two additional data points that represent the tangent to the curve at the landmark point and then calculate the loss function using the distance to these additional segments instead of the distance to the individual point.

Confidence Scoring Improvements: We currently output a confidence score associated with whether the model infers a face or not. Ultimately, we want the model to output a confidence score more closely associated with how likely the landmark points and inferred shapes are going to save an artist time. We also hope to explore incorporating feedback in the form of artist clicks into the confidence score.

Challenging Shots: The most challenging shots have occlusions such as characters facing sideways or characters wearing a hat.

Animated Transforms: We would like the tool to feature animation of points and shapes. We also hope to give the user some control over how many "keyframes"are output on a shot, to eliminate frames that do not perform well and to interpolate between the frames that do.

Improved Synthetic Data: We hypothesize that the synthetic data may add additional benefit if assets such as pre-production rendered animation could also be used.

Incorporation of Optical Flow: We also plan to use optical flow to improve temporal coherence for the inferred shapes across sequential frames. Additionally, we can score the correlation between the shapes inferred on the current frame and the prior frame and predict excessive deviation.

References

[1] Aseem Agarwala, Aaron Hertzmann, David H. Salesin, and Steven M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, 23(3):584–591, Aug. 2004.

[2] Hannes Appell, Sebastian H. Schmidt, and Nicolas Palme. Enhancing organic visual effects while simplifying rotoscoping techniques. In *ACM SIGGRAPH ASIA 2009 Sketches*, SIGGRAPH ASIA '09, New York, NY, USA, 2009. Association for Computing Machinery.

[3] Luis Bermudez, Nadine L. Dabby, Yingxi Adelle Lin, Sara Hilmarsdottir, Narayan Sundararajan, and Swarnendu Kar. Dataset for "a learning-based approach to parametric rotoscoping of multi-shape systems". `https://github.com/swkar/Rotomation`, 2020.

[4] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. volume 107, pages 177–190, 04 2014.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[6] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, June 2010.

[7] X. Dong, S. Yu, X. Weng, S. Wei, Y. Yang, and Y. Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 360–368, June 2018.

[8] Evan Goldberg. Medial axis techniques for stereoscopic extraction. In *SIGGRAPH 2009: Talks*, SIGGRAPH '09, New York, NY, USA, 2009. Association for Computing Machinery.

[9] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[11] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Stylizing video by example. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019.

[12] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. volume 1, pages 321–331, 01 1988.

[13] K. Khan, M. Mauro, and R. Leonardi. Multi-class semantic segmentation of faces. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 827–831, Sep. 2015.

[14] Jaehwan Kim, Jae Hean Kim, Sang Hyun Joo, Byoung Tae Choi, and Il Kwon Jeong. Volume matting: Object tracking based matting tool. In *ACM SIGGRAPH ASIA 2010 Posters*, SA '10, New York, NY, USA, 2010. Association for Computing Machinery.

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

[16] Martin Koestinger, Paul Wohlhart, Peter M Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 2144–2151. IEEE, 2011.

[17] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 109–117, Red Hook, NY, USA, 2011. Curran Associates Inc.

[18] Vineet Kushwaha, Maneet Singh, Richa Singh, Mayank Vatsa, Nalini Ratha, and Rama Chellappa. Disguised faces in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2018.

[19] Wenbin Li, Fabio Viola, Jonathan Starck, Gabriel J. Brostow, and Neill D. F. Campbell. Roto++: Accelerating professional rotoscoping using shape manifolds. *ACM Trans. Graph.*, 35(4), July 2016.

[20] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5257–5266, 2019.

[21] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*, IJCAI'81, pages 674–679, 1981.

[22] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3691–3700, July 2017.

[23] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, and et al. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6), Dec. 2018.

[24] Ondrej Miksik, Juan-Manuel Pérez-Rúa, Philip HS Torr, and Patrick Pérez. Roam: a rich object appearance model with application to rotoscoping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4691–4699, 2017.

[25] Kai Ruhl, Martin Eisemann, and Marcus Magnor. Cost volume-based interactive depth editing in stereo post-processing. In *Proceedings of the 10th European Conference on Visual Media Production*, CVMP '13, New York, NY, USA, 2013. Association for Computing Machinery.

[26] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J. Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] Jeff Stringer, Narayan Sundararajan, James Pina, Swarnendu Kar, Nadine L. Dabby, Adelle Lin, Luis Bermudez, and Sara Hilmarsdottir. Rotomation: Ai powered rotoscoping at laika. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery.

[28] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[29] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *CoRR*, abs/1904.04514, 2019.

[30] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, June 2013.

[31] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.*, 37(4), July 2018.

[32] Jue Wang, Bo Thiesson, Yingqing Xu, and Michael Cohen. Image and video segmentation by anisotropic kernel mean shift. volume 3022, pages 238–249, 05 2004.

[33] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. *ACM Trans. Graph.*, 23(3):574–583, Aug. 2004.

[34] Kirk A. Woolford and Carlos Guedes. Particulate matters: Generating particle flows from human movement. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, page 691–696, New York, NY, USA, 2007. Association for Computing Machinery.

[35] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 532–539, June 2013.

[36] Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: Landmark detection and geometric style in portraits. *ACM Trans. Graph.*, 38(4), July 2019.

[37] Kaan Yücer, Alec Jacobson, Alexander Hornung, and Olga Sorkine. Transfusive image manipulation. *ACM Transactions on Graphics (TOG)*, 31(6):1–9, 2012.

[38] H. Zhang, Q. Li, Z. Sun, and Y. Liu. Combining data-driven and model-driven methods for robust facial landmark detection. *IEEE Transactions on Information Forensics and Security*, 13(10):2409–2422, Oct 2018.

[39] Shizhan Zhu, Cheng Li, Chen-Change Loy, and Xiaoou Tang. Unconstrained face alignment via cascaded compositional learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.