The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

PVL: A Framework for Navigating the Precision-Variety Trade-Off in Automated Animation of Smiles

Nicholas Sohre,[1*] **Moses Adeagbo,**[1] **Nathaniel Helwig,**[2,3*]
Sofia Lyford-Pike,[4*] **Stephen J. Guy**[1*]
University of Minnesota
[1]Department of Computer Science [2]Department of Psychology [3]School of Statistics [4]Department of Otolaryngcology
*sohre@cs.umn.edu, helwig@umn.edu, lyfor009@umn.edu, sjguy@umn.edu

Abstract

Animating digital characters has an important role in computer assisted experiences, from video games to movies to interactive robotics. A critical challenge in the field is to generate animations which accurately reflect the state of the animated characters, without looking repetitive or unnatural. In this work, we investigate the problem of procedurally generating a diverse variety of facial animations that express a given semantic quality (e.g., very happy). To that end, we introduce a new learning heuristic called Precision Variety Learning (PVL) which actively identifies and exploits the fundamental trade-off between precision (how accurate positive labels are) and variety (how diverse the set of positive labels is). We both identify conditions where important theoretical properties can be guaranteed, and show good empirical performance in variety of conditions. Lastly, we apply our PVL heuristic to our motivating problem of generating smile animations, and perform several user studies to validate the ability of our method to produce a perceptually diverse variety of smiles for different target intensities.

Introduction

Virtual humans are increasingly a part of our games and other digital media. They appear in movies as animated actors, video games as interactive non-player characters, personal avatars in games, virtual reality and social media, and are even used to control human-like robots. A critical component of creating compelling interactions with digital characters is the animation of the human face. Humans use and expect faces to produce a variety of cues for nonverbal communication such as intonation and emotion. Understanding the full variety of movements that control and effect these cues is important both to fields that study real humans (e.g., medicine and psychology) as well as those which seek to create realistic virtual characters (e.g., games and movies).

Our goal in this work is to create algorithms that can automatically generate a variety realistic animations for virtual characters, a problem which is closely related to a field of AI known as Procedural Content Generation (PCG). PCG is especially relevant in the realm of games and interactive digital entertainment where is it important to present the user with engaging, dynamic experiences that respond to the user's actions in real time.

For procedurally animated virtual characters to meet their goal of emotionally engaging the users, there are two important qualities the procedural animations must maintain. Firstly, it is important that their expressions are as high quality and natural in appearance as possible. If the generated motion is halting, confusing, or otherwise unrealistic in its execution, the users will be distracted from the intended emotional content of the expression. Secondly, the procedural generation system must be able to create a variety of motions that is reflective of the full diversity real people have in showing the same basic expression. In fact, the importance of variety in character animations has been established through multiple users studies (McDonnell et al. 2008; O'Sullivan 2009) and has been highlighted as an important challenge in PCG (Preuss, Liapis, and Togelius 2014).

Unfortunately, these dual goals of generating high quality content and generating a diverse variety of content are often in direct conflict. Algorithms that focus too much on the quality of their content often do so by sacrificing the variety of their output. In this paper, we examine this trade-off in the context of procedural systems for creating mouth movements for virtual characters to form smiles of different intensity (e.g., slight, full, none), and propose new methods to produce a broad diversity of smiles that accurately display the target intensity level. Our work presents three main contributions:

- *Formalization and analysis of quality-variety trade-off*: We formally define the notions of quality and variety for a certain class of content generation models (constraint-based optimization formulations), and explore the theoretical basis of the inherent trade-offs between the two.
- *Precision Variety Learning heuristic (PVL)*: We introduce a framework for a constraint-based optimization formulations of PCG which allows a user to tune the level of precision needed for a specific application, and automatically maximize its variety of procedurally generated content for a given level of precision.
- *Variety-Enhanced, Data-driven Facial Animation System*: We apply our PVL generation approach to a non-parametric classifier trained on a recently published set of smile animation data (Helwig et al. 2017) in order to create a system capable of producing a large variety of smiles

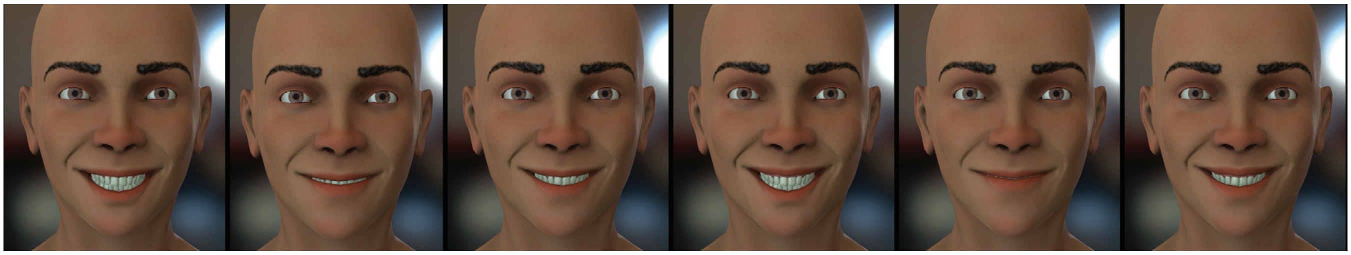


Figure 1: A variety of happy, smiling mouth shapes generated by our method, rendered in a high-quality real-time engine.

at a given level of smile intensity. We evaluate the quality and diversity of the resulting smiles through user studies.

While the results presented here focus on PCG smiles (e.g., see Figure 1), the approach is generic and can be directly applied both to other facial expressions (e.g, sad, angry), and to other forms of procedurally generated content.

Background

The animation of digital human-like faces has a rich history in the literature, from performance capture to modeling, to human perception of facial actions, and creating facial expressions for digital characters. Likewise, the study of PCG is a quickly growing field, covering everything from game maps and mechanics to textures and audio (Hendrikx et al. 2013). Below, we briefly highlight some closely related works.

Facial Animation

There is a rich literature surrounding the task of facial animation, an overview of which can be found in (Vinayagamoorthy et al. 2006). The most common technique is the use of a 3D spatial mesh that is then manipulated according to some model of facial movement. As with the models we employ here, many models of natural facial deformations are based on interpolative blendshapes (Zhang et al. 2016; Bouaziz, Wang, and Pauly 2013; Li et al. 2013; Xu et al. 2014). Blendshape-based models involve linearly interpolating the mesh between a set of exemplar configurations.

In many cases, the approach to animating these models utilize the capture of a facial performance by a human actor. Researchers have proposed various methods to accomplish this, from adaptive dimensionality reduction (Li et al. 2013), to neural networks (Costigan, Prasad, and McDonnell 2014) to local patch alignment (Zhang et al. 2016), and generating blendshape segmentation schemes (Joshi et al. 2005).

Generative methods for digital character facial expressions have also recently been explored. Some generate facial expressions from dialogue audio and text transcripts (Marsella et al. 2013). Physically-based models of the face can also be used to synthesize facial animation, such as speech (Sifakis et al. 2006).

Researchers have employed user studies to evaluate the effectiveness of digital character animation (Kokkinara and McDonnell 2015; McDonnell 2012; Liu et al. 2016), as well as to study the impact of variety (McDonnell et al. 2008; O'Sullivan 2009).

Machine Learning for Facial Analysis

Supervised learning is the most closely related area of machine learning to our work, surveyed in (Kotsiantis, Zaharakis, and Pintelas 2007). Others have developed specialized algorithms to recognize faces and facial actions (Pantic and Rothkrantz 2000; Franco and Treves 2001; Bartlett et al. 2005), as well as recognizing emotions (Michel and El Kaliouby 2003).

PCG as Machine Learning

There are many PCG techniques, and some synopses of the field are given in (Smith 2014; Hendrikx et al. 2013). Recent works have considered how to create engaging (Togelius et al. 2013), diverse (Liapis, Yannakakis, and Togelius 2015), and interactive (Yannakakis and Togelius 2011; Smith 2014) content. Machine learning techniques can be applied to PCG problems in different ways, as content evaluators or to generate content directly (Summerville et al. 2017; Togelius et al. 2011).

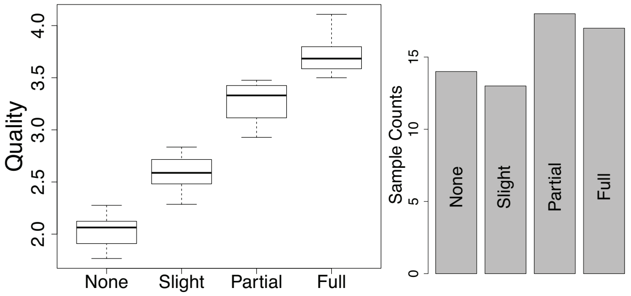
Diverse, High-Quality Content

Quality-Diversity algorithms have recently been identified as an important type of algorithm, with search-based approaches like evolutionary algorithms (Pugh et al. 2015) and Human-in-the-loop methods that combine user input with search to efficiently traverse search spaces (Mouret and Clune 2015) showing promise in this area. To the authors' knowledge, this work is the first to propose a machine-learning-based approach for this class of algorithms.

Problem Definition

As a motivating context for our problem formulation, consider the task of creating a 3D role-playing style game (RPG) where the player is immersed in an open world, free to explore and interact with many non-player characters (NPCs). To keep the NPCs engaging, their behaviors should be both appropriate to context (e.g., convey the right emotion), and appear natural and lifelike (i.e., not mechanically repetitious or robotic). To do this, we must be able to produce facial movements that exhibit the desired semantic meaning, while capturing the diversity of motion seen in real human faces, both within and across individuals. With these two goals as our primary focus, we can establish a formal definition of our problem.

We will represent facial animations as parameterized into a feature space $\mathcal{F}$, so that $x \in \mathcal{F}$ represents a complete facial



(a) Smile Quality Scores

(b) Happiness Bins

Figure 2: Training Data **(a)** A visual summary of the semantic classes. **(b)** Sample counts by class.

motion, and define $\mathcal{S}$ to be the set of semantic labels. Let us also define a function $D : X \subseteq \mathcal{F} \mapsto \mathcal{R}$ that operates on a set of faces to measure its diversity, and a function $Q_s : X \subseteq \mathcal{F} \mapsto \mathcal{R}$ as the quality of a set. Finally, let $C_s : \mathcal{F} \mapsto \{0, 1\}$ be a binary function that identifies whether or not a given animation exhibits a target semantic label s. Then, given some target $s \in \mathcal{S}$, our task is to find the set of faces exhibiting the desired semantic that maximizes the diversity and quality functions:

$$\underset{C_s}{\operatorname{argmax}} \big[Q_s(X), D(X) : \forall x \in X\big(C_s(x) = 1\big)\big]. \tag{1}$$

This equation represents a multi-objective optimization problem. To develop a solution for our domain of facial animations, we must establish quantifiable definitions of Q and D, identify an appropriate feature space for $\mathcal{F}$, and learn C. The remainder of this section describes our approach to each, followed by our proposed method for actually generating animations.

Measuring Quality & Diversity

We note that from here on we will assume X to be a finite set that is representative of C's continuous positive decision region in feature space. Then we define the Quality $Q(X)$ as the percentage of $x \in X$ that are true members of the target class:

$$Q_s(X) = \frac{|\{x \in X : [C_s^*(x) = 1]\}|}{|\{X\}|}. \tag{2}$$

Where C_s^* is the true semantic label function. In the context of equation 1, this is equivalent to the precision of the classifier D, which is how we will measure Q.

To measure the diversity of a set X, we will take its cardinality. This approach is consistent with existing measures of diversity for finite sets of candidate samples proposed in PCG (Preuss, Liapis, and Togelius 2014). Formally,

$$D(X) = |\{X\}|. \tag{3}$$

An important property for D is that adding members to X can never decrease the diversity measure overall (other reasonable diversity metrics, such as the variance of the set, do not satisfy this property).

Feature Space (F)

Along with their study, Helwig et al. proposed a generalizable, low-dimensional feature space to be used to represent smile animations. We refer to this feature space as *facial space*, and adopt it for $\mathcal{F}$. This feature space is composed of distances between key points surrounding the mouth as identified by medical professionals including: angle, extent, and dental show. Angle is computed as the angle between the bottom lip and mouth corner, extent is the width of the smile, and dental show is the separation between the upper and lower lips.

Classifier (C)

By definition, the task for C is one of classification. To do this, we will construct binary classifiers from annotated data via supervised learning that maps samples to a membership prediction given a target class in $\mathcal{S}$. Our formulation allows for any binary classifier, though different classifiers will have different theoretical properties and performance. Here, we consider several well established classifiers:

- *Nearest Neighbor models (KNNs)*: We employ a variant of KNN known as Restricted Neighborhood Search. The prediction for a sample is positive if a sufficient number of nearby neighbors (called witnesses) within some distance r are positive. The prediction for a query sample $q \in \mathcal{F}$ is positive if and only if

  $$\frac{\sum_{x \in \mathcal{W}_q} \pi(x)}{|\mathcal{W}_q|} \geq t \wedge |\mathcal{W}_q| \geq k, \tag{4}$$

  where $\mathcal{W}_q$ is the set of witnesses for q, t is the minimum proportion of witnesses that must be positive, k is the minimum number of witnesses to make a prediction, and $\pi(x)$ takes the value 1 if x is a positive training sample and 0 otherwise. For our classifier, we choose $k = 6$ based off the density of our training data, $r = 0.4$ based on the distribution of inter-point distances, and $t = 0.3$ via tuning.

- *Support Vector Machines (SVMs)*: these classifiers use quadratic programming to find a linear separator between positive and negative samples that maximizes the margin between them. A key property of SVMs is their use of kernels, which transform training data into higher dimensional spaces (where linear separators are more likely to be found) before measuring distances via an inner product. In this way, learning can take place in a high dimensional space while computation stays in a low dimensional space. Here, we employ the *kernlab* SVM package (Karatzoglou et al. 2004) for the R programming language, using the ”vanilla” kernel.

- *Random Forests (RFs)*: these classifiers take many random subsets of the training data and build decision trees on each. For prediction, a majority vote is taken of the random decision trees on the query sample, combating the tendency of decision trees to over-fit. Here, we employ the *randomForest* package for the R programming language (Liaw and Wiener 2002) with 1000 trees.

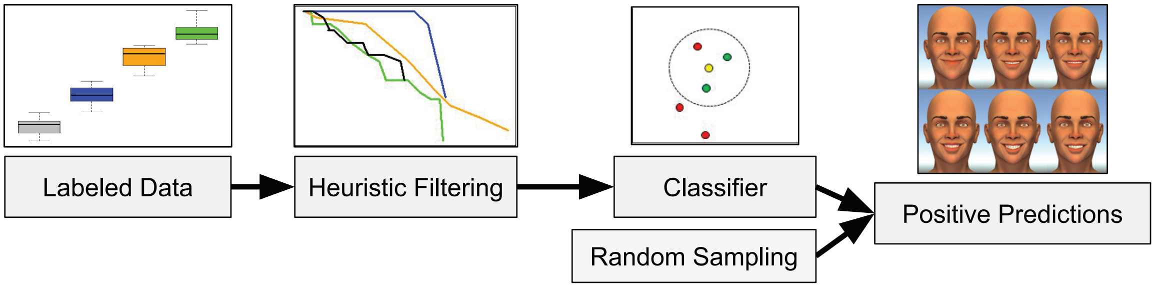


Figure 3: A graphical overview of our approach.

Semantic Classes ($\mathcal{S}$)

We will choose $\mathcal{S}$ to be a set of discrete classes, which we derive from training data that is then used to learn C. Discrete classes are motivated in part by the scenario of RPG-style video games; here, characters typically need to display one of a small set of emotions depending on the players behavior. Additionally, classification is a natural formulation for this problem (as opposed to regression) in that it allows a single semantic class to contain a variety of feature space points.

Approach & Implementation

An overview of how we utilize $\mathcal{S}$, C, and $\mathcal{F}$ to generate facial animations can be seen in Figure 3. Once training data has been labeled for a target class, we apply our learning heuristic as a pre-processing step, which we discuss in detail in the next section. A binary classifier is then trained on the labeled data to predict target class membership. We then use rejection sampling to generate new animations, uniformly sampling $\mathcal{F}$ and passing them through C keeping only those that yield positive predictions. To render these new samples as a human facial expression, they must be transferred onto a digital facial model. For this we use a 3D mesh with interpolative blendshapes as defined by an artist. We employ an iterative local optimization technique to solve for blendshape weights given a facial space target. Some examples (transferred onto the 3D model by our method and rendered by professional software) are shown in Figure 1 This transfer method can be applied to any facial animation system that is locally controllable in the feature space.

Dataset. In order to learn a model of happiness C and derive a set of semantic classes for $\mathcal{S}$, we will turn to a dataset of annotated facial movements from (Helwig et al. 2017). This dataset consists of results from a large-scale user study at a state-wide fair. Participants were shown expressions from a sweep of anatomically plausible mouth movements on a tablet device, and asked to assign a quality score for how well the face portrayed a smile. Over 900 subjects participated in the survey, providing over 10,000 responses in total. The stimuli contained mostly smile-like faces, but also had some negatively angled mouths, which served as controls. We aggregate the responses to produce a dataset composed of 63 facial expressions annotated with their mean perceived smile quality.

Smile Intensities. We derive $\mathcal{S}$ by defining ranges of quality scores from the dataset as four discrete classes of smiles: *None, Slight, Partial, and Full*. A summary view of the resulting classes are shown in Figure 2. An ANOVA test shows high statistical significance with 4 classes, with $[F(3,60) = 863.5, p < 0.001]$. A post-hoc analysis also confirms statistical significance between all pairs of classes. Figure 2b shows similar class sizes.

Experimental Methodology. We compute a (noisy) estimate of precision on our real-world face data via a hold-one-out cross validation loop, computing the mean precision over the folds using held out samples as test data. We also compute a variety estimate within each fold, taking the mean over the folds. Variety estimates are computed on a set of 1000 uniformly sampled points in facial space within the bounding volume of the training data. These samples are passed to the classifier, and the variety is reported as the proportion that are predicted positive. We then validate our results with a follow-up user study.

Maximizing Variety, Maintaining Precision

Our approach makes use of a binary classifier to identify faces that match the targeted semantic class. While traditional binary classifiers seek to maximize predictive accuracy, maximizing this alone fails to highlight the important trade-off between the precision of the classifier and the diversity of faces that will be generated. Because our model only generates positively classified faces, false positives will be discarded, unseen by any user. As a result, for many animation contexts maximizing precision is the most important goal; the quality of the faces generated by our method is unaffected by false negatives.

However, we would like to support the generation of a large variety of positively classified faces (e.g, many faces that look happy in different ways). In every classification task there is a fundamental trade-off between precision and variety; maximizing one comes at the cost of the other. Consider the positive decision region of the feature space on which our definition of variety depends: as this region grows larger, the classifier has an increased risk of producing false positives due to encroaching on regions that contain true negatives. Below, we explore this trade-off within the context of our facial generation system, and then present a learning heuristic method that exposes this trade-off to allow us to maximize

Algorithm 1: PVL Prediction

```
Input   : $sample, trainData, pClass, m$
Output : $prediction$
$pos \leftarrow$ getPositiveSamples($trainData, pClass$);
$neg \leftarrow$ getNegativeSamples($trainData, pClass$);
$pos \leftarrow$ sortByDistanceToNearest($pos, neg$);
$pos \leftarrow$ getfirstNSamples($positive, m$);
$trainData \leftarrow$ union($positive, negative$);
$prediction \leftarrow$ getPrediction($trainData, sample$);
return($prediction$);
```

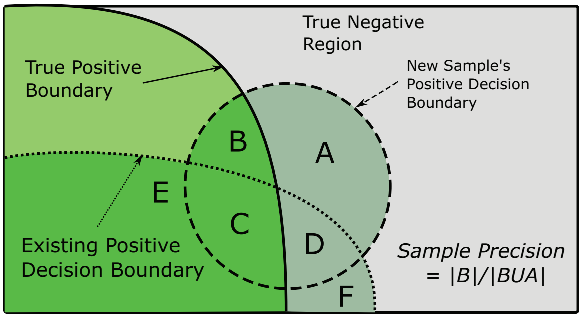

True Negative Region
True Positive Boundary
New Sample's Positive Decision Boundary
B
A
E
C
D
F
Existing Positive Decision Boundary
Sample Precision = |B|/|BUA|

Figure 4: Sample Precision Conceptual regions when adding a positive sample into the training set are depicted and labeled. We define sample precision as the ratio of area **B** to area **A**. The precision of the existing classifier is the ratio $|\mathbf{C} \cup \mathbf{E}|/|\mathbf{C} \cup \mathbf{E} \cup \mathbf{D} \cup \mathbf{F}|$, and the precision of the resulting classifier is $|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E}|/|\mathbf{B} \cup \mathbf{C} \cup \mathbf{E} \cup \mathbf{A} \cup \mathbf{C} \cup \mathbf{F}|$

variety in positively classified faces while retaining as much precision as possible.

Precision Variety Learning

The key insight which enables our approach is that high precision can be ensured by carefully selecting which positive samples are allowed in to the training set. For example, choosing to only include positive training samples that are far away from negative samples can increase the precision of the model at the cost of false negatives, which is a favorable trade given our goals. However, including too few positive training samples results in very little variety, which is an equally important objective. Varying the positive samples allowed into the training set exposes this trade-off for tuning between precision and variety.

To that end, we introduce a parameter m that controls what samples are used in the training set for a binary classifier (e.g, KNN). The training set is constructed by a heuristic ordering of the positive training set by *sample precision*. To define sample precision, we look at the subregion of the positive decision region that is added by a sample given an existing classifier. Sample precision is taken to be the proportion of this new region that overlaps the true positive region of the feature space. Figure 4 illustrates the regions involved and how they are used. Importantly, sample precision considers only the *additional* positive decision area supported by the new training sample. When samples are arranged such that sample precision is decreasing, we say they are in *precision-optimal* order. The first m positive training samples (i.e., with the m highest sample precisions), together with all of negative training samples are provided as input the the binary classifier. For all m, all negative training samples are included as they do not increase the risk of generating a false positive. We call the resulting approach *Precision-Variety Learning* (PVL), and the algorithm is presented in Algorithm 1.

Unfortunately, a positive training sample's sample precision cannot be computed directly as it depends on the ordering of the points added to the classifier before it. We therefore propose an order-independent estimation of sample precision as the distance of a given positive training sample to its nearest negative neighbors. Intuitively, this heuristic captures the fact that false positives (which reduce precision) are likely to lie near negative training samples. This assumption is explored further in the following section.

The key feature of m is the way in which it captures and exposes the trade-off between precision and variety. This is our solution to the multi-objective optimization problem posed in equation 1. Like the pareto-fronts used in many solutions to multi-objective problems, m exposes a precision-variety front that can be exploited to gain as much variety as possible for a desired level of precision. While we do not claim that m generates a pareto-optimal front, we can identify conditions that guarantee the monotonicity of the front, which m is designed to produce. A critical property of pareto fronts, monotonicity insures that any loss of one objective does not allow the loss of the other (e.g, giving up precision will either maintain or increase variety). This also enables a directed search for optimizing m given a desired precision or variety.

In the case of a neighbor based classifier such as KNN with a precision-optimal ordering of positive training samples, the resulting trade-off front is provably monotonic in m under some supporting assumptions. By monotonicity we mean for increasing m, variety does not decrease and precision does not increase, and vice versa for decreasing m. To prove this, it is sufficient to show that as m increases, we have non-increasing precision and non-decreasing variety. Our formal arguments for each are as follows.

Proof of Monotonicity

When used with a neighbor-based classifier (such as KNN), there are several key theoretical properties which are maintained by using the PVL approach, which we demonstrate below. The first is that, under certain conditions of the underlying data, the precision of the classifier decreases monotonically as m increases. We also show, regardless of the quality of data, both that specificity (the rate of true negatives) decreases monotonically and variety increases monotonically. Taken together, this means as m increases our predictions will have more inaccuracies (both in terms of admitting false positives and rejecting true negatives), but will increase variety; this serves as the theoretical bases for our claim that PVL is navigating a trade-off between the quality of procedurally generated content and its variety.

Definition 1: Quality of Approximation. We define the *quality of approximation* of our heuristic for a given dataset as the degree to which our distance-based ordering maintains a precision-optimal ordering. The quality of approximation will be high when two conditions hold: 1) the data has a clear positive decision boundary (i.e., samples are more homogeneous the further they are from the boundary) and 2) the boundary has limited curvature. Because our heuristic ordering first adds points that are far away from negative samples, the existence of a clear decision boundary ensures initial points will contribute new positive classification area with higher precision than later points which are closer to the boundary. Assuming limited curvature allows us to safely approximate the distance to the decision boundary as the distance to the single nearest negative sample.

Theorem 1: Decreasing Precision as m ***increases.*** Let P_m be the precision of the classifier for arbitrary m and P_{m+1} be the precision of the classifier after including the $(m+1)$th positive training sample. Further let P_s be the sample precision of the $(m+1)$th sample. Given their respective false positive (FP) and true positive (TP) counts we can compute the precision of new classifier with $m+1$ samples as:

$$P_{m+1} = \frac{TP_m + TP_s}{TP_m + TP_s + FP_m + FP_s}. \tag{5}$$

We therefore need to show that $P_m \geq P_{m+1}$, that is:

$$\frac{TP_m}{TP_m + FP_m} \geq \frac{TP_m + TP_s}{TP_m + TP_s + FP_m + FP_s}, \tag{6}$$

which (by cross multiplication) is equivalent to the condition

$$TP_m * FP_s \geq TP_s * FP_m. \tag{7}$$

When the quality-of-approximation (*Definition 1*) hold perfectly, we have $P_m \geq P_s$, which implies

$$\frac{TP_m}{TP_m + FP_m} \geq \frac{TP_s}{TP_s + FP_s}$$
$$\iff TP_m * FP_s \geq TP_s * FPm, \tag{8}$$

satisfying the requirement of equation 7.

Theorem 2: Decreasing Specificity as m ***increases.*** As with precision, maximizing specificity (true negative rate), is important for a classifier that is to be used in the generation of procedural content. We note that specificity and precision can be jointly optimized via the elimination of false positives. Formally, specificity is defined as

$$TN/(TN + FP), \tag{9}$$

where TN represents the true negatives and FP the false positives of a classifier. To show we have decreasing specificity over m, it suffices to observe that increasing m only adds positive training samples to the classifier. As a result, the negative decision region of a neighbor-based classifier cannot increase, and the positive decision region cannot decrease. Thus, false positives are increasing and true negatives decreasing, constraining specificity to decrease. Notably, this property is independent of the order in which the positive samples are added.

Theorem 3: Increasing Variety as m ***increases.*** The supporting argument for increasing variety over m is already established in *Theorem 2*; since adding positive samples constrains the positive decision region to increase, by definition the variety of the classifier will also increase. This property is also independent of the positive samples' order of inclusion.

Results & Analysis

Behavior of m**.** To observe the impact of m on classification, we estimate precision and variety over different values of m on a synthetic dataset with a circular ground truth decision boundary. This allows precision to be computed with arbitrary accuracy by sufficiently sampling the feature space and testing them on the classifier. Similarly, variety can be estimated by sampling in the feature space and measuring the positive classification rate. Figure 5 shows our results using the KNN classifier: for small m, the precision of the model remains high, but results in a classifier that produces little variety when sampled. Conversely, for large m, a larger variety of points can be generated, at the cost of precision. Thus, m allows us to tune the precision/variety trade-off in the learning process. This curve exhibits the expected monotonicity for this type of classifier.

The curve produced by varying m resembles the ROC curves used to indicate the performance of binary classifiers. Just as ROC curves report the interplay between two conflicting goals of interest (true positive rate and false positive rate), our PVL curves report the performance of a binary classifier in terms of two other conflicting goals relevant to the task at hand.

Comparing Classifiers As our PVL heuristic supports multiple classification techniques, we compare several algorithms in terms of their precision and specificity over m. Specificity-variety curves for the different classifiers on our real-world data with four classes are shown in Figure 6. As in Figure 5, each curve exhibits increasing m from left to right, with the exception of the Partial and Slight classes for

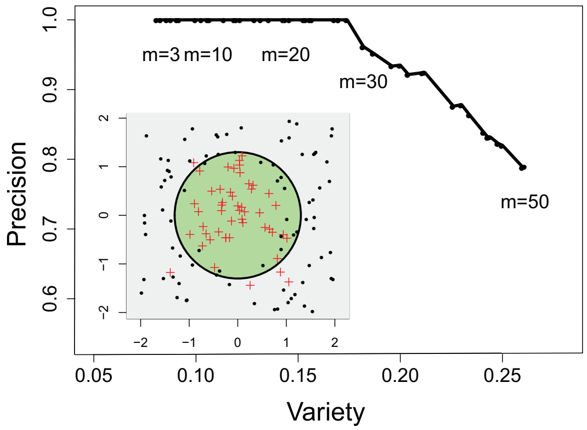


Figure 5: Precision-Variety Trade-off curve over m for synthetic circular boundary data.

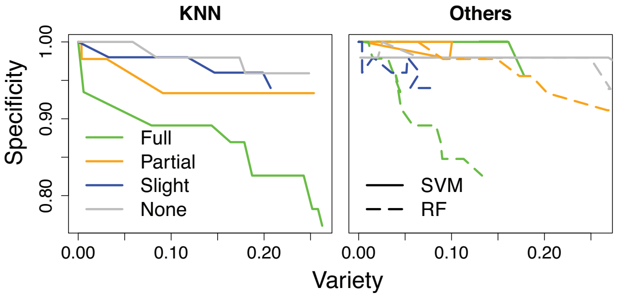


Figure 6: Specificity Curve Comparison Specificity curves over m for each semantic class using different supervised learning methods.

SVM. The KNN classifier curves exhibit the monotonicity guaranteed for specificity over increasing m.

We also compute curves for Precision, as depicted in Figure 7. The limited number of positive samples in our face data cause the uncertainty in estimating precision to be prohibitively large for four classes. To accommodate for this, we construct two classes from the data, and suppress values of m that produce less than 10 positive predictions. In the case of KNN, our method shows a strong monotonic trend, demonstrating its effectiveness on real-world data where our data condition assumptions (see *Definition 1*) do not hold perfectly.

Notably, the SVM and RF algorithms differ from KNN in both the specificity and precision variety curves; the general behavior is similar, but can be erratic for some classes (such as Slight and Partial). While our theoretical guarantees concerning monotonicity do not extend to RFs and SVMs, in practice the curves tend towards monotonicity; SVMs preserve monotonicity when conditions are favorable (and suffer more erratic behavior when conditions are poor), and RFs robustly exhibit a general if not local monotonic trend. Theoretical similarities between RFs and neighbor based methods have been noted (Lin and Jeon 2006), which likely contribute to this phenomenon.

Analysis of Faces. Our method is capable of producing a variety of mouth shapes with a targeted smile intensity. Taking advantage of its theoretical properties, we use our KNN based set of classifiers to train C and render the resulting facial animations. Figure 1 demonstrates some examples where the *Full* smile class was targeted, with $m = 8$. This m value provides a large gain in variety without a large loss of precision, resulting in faces that differ in appearance, but are all happy. Figure 9 shows some examples of training our PVL model to produce faces from other semantic categories. The middle and bottom rows show faces generated from classes Slight and None respectively.

We note that all of our semantic classes exhibited a large amount of variety, though it varies with the range of m (which is bound by the sample counts in Figure 2). While there is generally more variety achievable for a given level of precision in the *None* class (as a smile is just one of many kinds of facial expressions), there is still significant variety present for *Full* smiles, including those that have no dental show (as in the top center of Figure 9). This highlights the fact that no single feature was responsible for the semantic meaning of the expression.

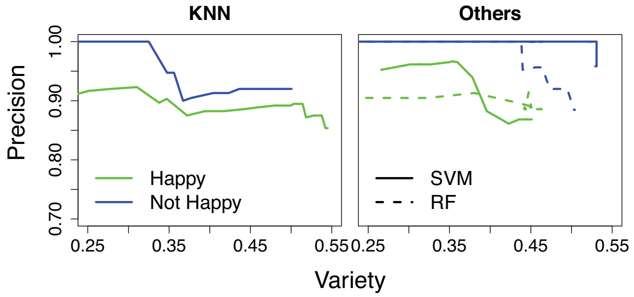


Figure 7: PVL Curve Comparison PVL curves over m for a two-class split of our face data using different supervised learning methods.

Validation Study. To validate our method's effectiveness in creating a variety of faces within a targeted happiness level, we performed a two-part user study with 19 participants (11 women and 8 men with average age 28.3). We created a second facial model of a different race to test the extendability of our approach to other models. Subjects assessed side-by-side pairs of generated animations in terms of their similarity and happiness (see Figure 8).

The first section of the study analyzed the PVL learning approach. Here, pairs of faces were shown from sets generated using the same m value (i.e., either two faces from the m=3 set or two faces from the m=8 set) with *Full* as the target class . Some examples of the faces shown can be seen in the top row of Figure 9. Participants were asked to indicate on a Likert scale how similar the two facial expressions appeared. Our hypothesis was that pairs from sets with lower m would be more similar (have less variety) than sets from higher m. A Wilcoxon signed rank test confirms ($\mathbf{Z} = 4.78$, $\mathbf{p} < 0.0001$, $\mathbf{r} = 0.367$) that comparisons between two expressions from

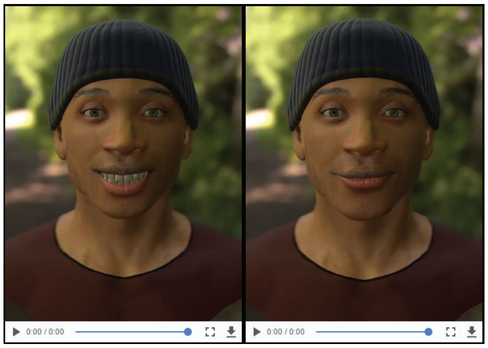


Figure 8: An example question from our follow-up user study, using a different virtual character.

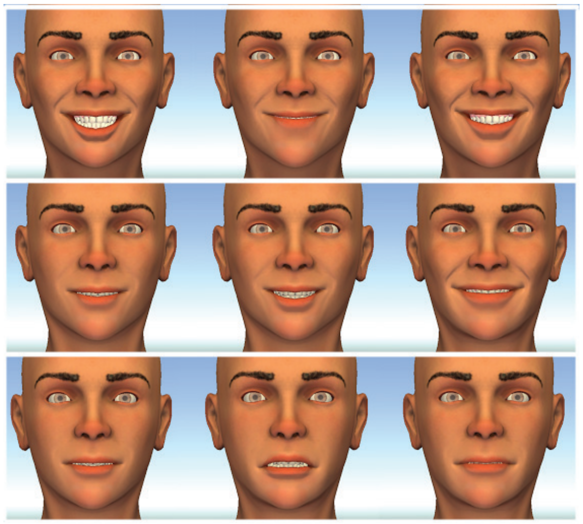


Figure 9: Example set of generated faces with High (top row) Medium (middle row) and None (bottom row) targeted happiness levels.

$m = 3$ were perceived as more similar than those generated with m = 8, confirming our hypothesis and validating our PVL learning approach.

The second section of the study aimed at validating the predictive accuracy of our PVL classification approach. Here we used a two-alternative forced choice (2AFC) design where participants were asked to indicate which of two smiles appeared happier. In every pair of smiles, one came from a set of smiles with a predicted *Partial* category smile and one with a predicted as *Full*. Both sets of smiles were generated with m=8, so as to have a variety of different smiles. A two-sided binomial test showed that smiles predicted as *Full* were most likely to be rated happier ($\mathbf{p} < 0.001$, P(success)$= 0.68$, $RR = 1.26$), validating our ability to generate faces with different smile intensities.

Conclusions

In this work, we have proposed and implemented a system for the generation of a variety of smiles for use in digital characters. We formulated our problem as a multi-objective optimization task, seeking both high quality and diverse animations. Our approach to generate new animations utilized a dataset of annotated facial expressions as training samples for a binary classifier to predict whether or not a new facial expression would be perceived as having a targeted semantic class. To solve our multi-objective problem, we introduced *Precision-Variety Learning*, which allows a balance between precision and variety of a classifier to be directed by manipulating the training data set, providing theoretical guarantees under certain conditions. The classifier was then used to generate a variety of faces with targeted smile intensities novel to the existing data.

Limitations. Some limitations of our method motivate further study. The dataset we used is limited in terms of its coverage of plausible facial positions. Data covering a larger range could enable the study of a more diverse set of emotions. Another limitation is the fact that the blendshapes used for animation have a limited extent, thereby necessitating constrained optimization. This could be relaxed by allowing blendshapes to be extrapolated past their original bounds. We also note that increasing the coverage and density of the annotated faces could allow for more granular categories or regression classifiers to be trained. This could support more fine-tuned control over the target emotions or the generation of mixtures of emotions. While empirically our PVL approach performs very well, the theoretical properties are dependent on some data assumptions that may not hold in real-world settings. Further analysis may identify guarantees that hold when these assumptions are relaxed.

Future Work. In the future we intend to explore extensions to our work both in the areas of facial animation and the uses and properties of PVL. One such avenue is the generation of other emotions and mixtures of emotions by incorporating additional datasets and facial features. Building data-driven models that capture how perceived emotional intent relates to facial movement has implications beyond making compelling digital characters. As computational techniques allow us to permute facial positions in a way that human actors cannot, another exciting area of future work is to investigate faces with large asymmetry or other issues which may arise from facial trauma or nervous system damage. This can allow our work to inform areas of medicine such as facial reconstructive surgery, emotional recognition therapy, and psychologists looking to quantitatively study how intervention can help patients express emotional intent. We also plan to explore the application of PVL to different domains. Additionally, we will investigate further the theoretical properties of our approach, such as alternate heuristic orderings for m, conditions that influence existence of precision-optimal orderings, and what guarantees can be made for different classifiers and different data conditions.

Acknowledgements

This work has been supported in part by the National Science Foundation through grants #CHS-1526693, #CNS-1544887, and #IIS-1748541.

References

Bartlett, M. S.; Littlewort, G.; Frank, M.; Lainscsek, C.; Fasel, I.; and Movellan, J. 2005. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, 568–573. IEEE.

Bouaziz, S.; Wang, Y.; and Pauly, M. 2013. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)* 32(4):40.

Costigan, T.; Prasad, M.; and McDonnell, R. 2014. Facial retargeting using neural networks. In *Proceedings of the Seventh International Conference on Motion in Games*, MIG '14, 31–38. New York, NY, USA: ACM.

Franco, L., and Treves, A. 2001. A neural network facial expression recognition system using unsupervised local processing. In *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, 628–632. IEEE.

Helwig, N. E.; Sohre, N. E.; Ruprecht, M. R.; Guy, S. J.; and Lyford-Pike, S. 2017. Dynamic properties of successful smiles. *PloS one* 12(6):e0179708.

Hendrikx, M.; Meijer, S.; Van Der Velden, J.; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9(1):1.

Joshi, P.; Tien, W. C.; Desbrun, M.; and Pighin, F. 2005. Learning controls for blend shape based realistic facial animation. In *ACM SIGGRAPH 2005 Courses*, 8. ACM.

Karatzoglou, A.; Smola, A.; Hornik, K.; and Zeileis, A. 2004. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software* 11(9):1–20.

Kokkinara, E., and McDonnell, R. 2015. Animation realism affects perceived character appeal of a self-virtual face. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15, 221–226. New York, NY, USA: ACM.

Kotsiantis, S. B.; Zaharakis, I.; and Pintelas, P. 2007. Supervised machine learning: A review of classification techniques.

Li, H.; Yu, J.; Ye, Y.; and Bregler, C. 2013. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.* 32(4):42–1.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2015. Constrained novelty search: A study on game content generation. *Evolutionary computation* 23(1):101–129.

Liaw, A., and Wiener, M. 2002. Classification and regression by randomforest. *R News* 2(3):18–22.

Lin, Y., and Jeon, Y. 2006. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association* 101(474):578–590.

Liu, K.; Tolins, J.; Tree, J. E. F.; Neff, M.; and Walker, M. A. 2016. Two techniques for assessing virtual agent personality. *IEEE Transactions on Affective Computing* 7(1):94–105.

Marsella, S.; Xu, Y.; Lhommet, M.; Feng, A.; Scherer, S.; and Shapiro, A. 2013. Virtual character performance from speech. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 25–35. ACM.

McDonnell, R.; Larkin, M.; Dobbyn, S.; Collins, S.; and O'Sullivan, C. 2008. Clone attack! perception of crowd variety. In *ACM Transactions on Graphics (TOG)*, volume 27, 26. ACM.

McDonnell, R. 2012. *Appealing Virtual Humans*. Berlin, Heidelberg: Springer Berlin Heidelberg. 102–111.

Michel, P., and El Kaliouby, R. 2003. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th international conference on Multimodal interfaces*, 258–264. ACM.

Mouret, J.-B., and Clune, J. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

O'Sullivan, C. 2009. Variety is the spice of (virtual) life. In *International Workshop on Motion in Games*, 84–93. Springer.

Pantic, M., and Rothkrantz, L. J. 2000. An expert system for recognition of facial actions and their intensity. In *AAAI/IAAI*, 1026–1033.

Preuss, M.; Liapis, A.; and Togelius, J. 2014. Searching for good and diverse game levels. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, 1–8. IEEE.

Pugh, J. K.; Soros, L. B.; Szerlip, P. A.; and Stanley, K. O. 2015. Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 967–974. ACM.

Sifakis, E.; Selle, A.; Robinson-Mosher, A.; and Fedkiw, R. 2006. Simulating speech with a physics-based facial muscle model. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 261–270. Eurographics Association.

Smith, G. 2014. The future of procedural content generation in games. In *Proceedings of the Experimental AI in Games Workshop*.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2017. Procedural content generation via machine learning (pcgml). *arXiv preprint arXiv:1702.00539*.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):172–186.

Togelius, J.; Champandard, A. J.; Lanzi, P. L.; Mateas, M.; Paiva, A.; Preuss, M.; and Stanley, K. O. 2013. Procedural content generation: Goals, challenges and actionable steps. In *Dagstuhl Follow-Ups*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Vinayagamoorthy, V.; Gillies, M.; Steed, A.; Tanguy, E.; Pan, X.; Loscos, C.; and Slater, M. 2006. Building expression into virtual characters.

Xu, F.; Chai, J.; Liu, Y.; and Tong, X. 2014. Controllable high-fidelity facial performance transfer. *ACM Transactions on Graphics (TOG)* 33(4):42.

Yannakakis, G. N., and Togelius, J. 2011. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing* 2(3):147–161.

Zhang, J.; Yu, J.; You, J.; Tao, D.; Li, N.; and Cheng, J. 2016. Data-driven facial animation via semi-supervised local patch alignment. *Pattern Recognition* 57:1–20.