Title
A Framework for Learning Photographic Composition Preferences from Gameplay Data

Author
Escoffery, Dustin

Publication Date
2012

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

SANTA CRUZ

A FRAMEWORK FOR LEARNING PHOTOGRAPHIC COMPOSITION PREFERENCES FROM GAMEPLAY DATA

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Dustin Escoffery

March 2012

The Thesis of Dustin Escoffery
is approved:

Professor Arnav Jhala, Chair

Professor James Davis

Professor Alex Pang

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures **iv**

Abstract **v**

Acknowledgments **vi**

1 Introduction **1**

2 Related Work **5**

3 The Panorama Framework **10**
3.1 Architecture 11
3.2 Panorama Game 13
3.3 Implementation 17
3.4 Image Corpus 23
3.5 Evaluation 26

4 Preference Learning **34**
4.1 Image Synthesis 35
4.2 Composition Features 36
4.3 Crowdsourcing Preferences 42
4.4 Machine Learning 44
4.5 Results and Discussion 50

5 Conclusion **52**
5.1 Future Work 53

References **54**

List of Figures

3.1 The Panorama framework 11
3.2 Panorama game interface (left) and picture-taking feedback (right) 13
3.3 Camera control: pan, zoom, and tilt 15
3.4 A typical game level 17
3.5 Wireframe rendering of a windmill object 19
3.6 Hidden bounding boxes for measuring composition 21
3.7 Images produced in gameplay 25
3.8 Good balance (left), rule of thirds (center), and spacing (right) . . 26
3.9 Example photo with annotations 32

4.1 Quadtree decomposition of image space 39
4.2 Web form for collecting pairwise image preferences 42
4.3 A multilayer perceptron for preference classification 45

Abstract

A FRAMEWORK FOR LEARNING
PHOTOGRAPHIC COMPOSITION PREFERENCES
FROM GAMEPLAY DATA

by

DUSTIN ESCOFFERY

The automatic evaluation of images in computational cinema and photography is a challenging problem. There are neither comprehensive rules nor adequate training data to develop an expert system. This thesis presents the design and implementation of a computer game for synthesizing image data, and an experimental framework for learning photographic composition preferences from online ratings.

The first topic addressed is the development of Panorama, a computer game for photographic composition research. Panorama generates images through gameplay by simulating the task of photography in a virtual environment. The synthesized photographs are automatically annotated from their underlying representation, and contribute to a corpus of images with well-defined visual features. Both the game and data are publicly available to support research in image analysis.

The second topic is a photograph learning experiment using data from the Panorama corpus. In this section, machine learning is used for predicting user preferences of images, based on computed visual features. Image preference ratings are acquired by crowdsourcing to consider subjectivity across many individuals. Using this unique data collection process, it is shown how the framework can be applied to reason about image quality with respect to photographic composition.

Acknowledgments

I would like to thank the following individuals, without whom this scholarship would not be possible:

- Arnav Jhala—for supporting my graduate studies at UC Santa Cruz and guiding my research in games and computational media
- Reid Swanson—for collecting supplemental data and doing preference learning in this domain as affirmation of my framework
- Wai Son Wong—for enduring long summer days with me in the lab to develop the game presented in this thesis
- My Parents—for emphasizing the importance of my education and giving me their love and support in all of my endeavors

Chapter 1

Introduction

Do computer games have merit beyond entertainment? This thesis considers the application of gaming technology to interdisciplinary problem solving. The research explores gameplay as an interface for data collection, with the objective of building machine learning models to reason about complex phenomena. In particular, this thesis presents a framework that addresses the challenging problem of learning to evaluate photographic composition.

This work is motivated by image sharing websites, social networking services, and modern smartphones with built-in high-resolution cameras, which have contributed to bringing digital photography to the forefront of popular culture. A significant interest in photo sharing has emerged, creating a powerful expressive medium. However, the digital images frequently associated with these technologies are not typically informed by artistic principles. The average photo-taker is not an

expert photographer, and does not possess the requisite skills to maximize the use of the medium. But, every person has a natural sense of quality, and can identify good and bad images. So is it possible to train an expert system to reinforce the photographic principles informed by our fundamental preferences?

Historically, games have been a popular medium for teaching real-world skills. Early examples like *Reader Rabbit* and *Math Rabbit* used computer gameplay to aid children in understanding literacy and mathematics. Educational games, like *Mario Teaches Typing*, were specifically designed to combine entertainment with skill development. Over the decades, the application of gameplay to other disciplines has evolved to where, for example, the United States military now employs multiplayer simulation games to provide soldiers with a virtual learning environment [15]. In academia, researchers actively study how gaming technology can best be used to deliver rich educational experiences [6].

Still, a rather unexplored application of gaming technology is to examine artistic concepts like visual aesthetics. There is potential in using games to gain insight about subjective disciplines, because the rules are not well defined, and can be improved from interactive player feedback. Photography and image composition in particular is an ideal subject, because the rules are strongly influenced by people's beliefs. Humans are naturally well-trained at judging visual quality, given their exposure to years of artistic tradition. However, each individual has their own subjective preferences, and their expertise is not

easily explained. So, a logical approach is to use machine learning on human-provided preferences in an effort to discover how certain image characteristics influence one's perception of quality.

The technical challenge for modeling photographic judgment is in finding appropriate data to train an expert system. Manually collecting photographs with controlled content is difficult [17], and requires computer vision algorithms to determine features. The work described in this thesis takes an alternative approach, which explores a novel data generation method for image synthesis and automatic annotation.

Panorama is a game for creating a corpus of images with well-defined visual features. The gameplay is such that players take photographs of an artificial environment testing their composition skills. The players are scored in real-time based on heuristics for *balance*, *rule of thirds*, *spacing*, and custom features derived from the underlying properties of each image. Scoring teaches players to be better photographers as the levels progress, while at the same time, providing information on what kind of photos people take in practice. Most importantly, the images produced are automatically annotated for visual features given the game's internal representation. As a result, data is generated efficiently while the game is played.

Images generated through gameplay are semantically similar, such that the subject is always an abstract black-and-white landscape. An experiment for judging relative quality is therefore more meaningful, because users will not

have subjective bias toward the contents of the picture. One such experiment is described, in which online users rate Panorama images for relative quality.

A principle part of this thesis is demonstrating how machine learning can be applied to the image data produced in gameplay. In this system, images are automatically annotated for select compositional features, and rated by users on a crowdsourcing platform. The data are used in a machine learning experiment to predict how users prefer one image over another based on composition. Two crowdsourcing platforms and multiple features are considered for one set of images. The results indicate that the whole Panorama framework for combining gameplay data with the opinion of crowds can be used to incrementally learn to model image preferences with respect to photographic composition.

This thesis begins by surveying related work pertaining to visual composition and preference learning (Chapter 2). Then, the Panorama framework is described, including the design, implementation, and application of the Panorama game (Chapter 3). Next, a photograph learning experiment is discussed that uses Panorama images and human ratings to train a model for estimating quality (Chapter 4). Finally, results are analyzed and ideas for future work are proposed (Chapter 5).

Chapter 2

Related Work

The research presented in this thesis is related to work in computer graphics, computer vision, and machine learning. The graphics and vision communities are interested in image composition, and the machine learning community is interested in modeling preferences from data. As such, there are notable academic sources from which this thesis draws value.

Photography professionals use the term *photographic composition* to describe the layout of visual elements in an image [7]. Generally it is believed that photographic composition contributes to aesthetic quality. Computer scientists working in the fields of computer graphics and computer vision are interested in photographic composition for applications in rendering and image processing.

Bares [3] describes an interactive camera system that follows photographic composition rules. He provides a tool for artists to act as the "virtual

photographer" by specifying high-level image goals, which are satisfied by a camera controller. Users indicate composition objectives for *balance*, *placement*, and *emphasis*, and a constraint solver optimizes the criteria to provide framing suggestions. Some of the rules used in this thesis (Section 3.5) were inspired by Bares's composition objectives.

Abdullah *et al.* [1] also study photographic composition in the context of camera control. They distinguish between "basic" rules like size and position, and "advanced" rules that represent high-level aesthetic features. They develop a camera system that uses particle swarm optimization for advanced compositional rules like *rule of thirds*, *diagonal dominance*, *visual balance*, and *depth of field*. These features are automatically calculated from the virtual scene using custom rating functions, as is done in this thesis. However, they choose to evaluate images with a "rendering approach", instead of a bounding box representation. Their technique is reportedly more accurate but less efficient than using geometric approximation.

Chang and Chen [5] employ photographic composition in a search problem, and describe a method for finding good compositions in panoramic images. In their work, quality is not a function of predefined heuristics, but is measured statistically from a database of expert photographs. Images are analyzed for structure and visual salience in a computer vision step in order to find similar candidates. They use stochastic search to find images most similar to expert examples. This thesis

uses gameplay as an interactive alternative for finding good compositions within large scenes.

Liu *et al.* [13] discuss a technique for cropping images to computationally improve composition. They combine several of the aforementioned techniques for a practical demonstration. They do a preprocessing vision step similar to Chang and Chen [5] to detect regions of visual salience. The regions are scored for rules similar to Abdullah *et al.* [1], and likewise use particle swarm optimization to suggest crop parameters. Their system illustrates a useful application of composition evaluation: to computationally improve aesthetics.

Banerjee *et al.* [2] describe an application of photographic composition in digital photography. They describe an algorithm that applies composition rules to automatically align the subject and shift the focus in a digital camera. Their system can easily be improved as new composition heuristics are discovered.

Lok *et al.* [14] discuss how composition can inform the layout of user interfaces. However, unlike photographic composition, they are more concerned with concepts like *visual balance*, and less about photography principles like *rule of thirds*.

Su *et al.* [22] propose a framework for providing composition recommendations, using image analysis and machine learning. They construct a personal preference model in an offline learning step, in which users specify preferred photographs of scenic landscapes. Instead of composition rules, they employ a "bottom-up" approach for extracting thousands of features, and selecting relevant ones

by boosting. A limitation of their system is finding the appropriate image dataset. The framework presented in this thesis circumvents this problem by generating its own dataset.

Ke *et al.* [10] suggest a set of high-level image features to use for machine learning classification. Their features were inspired by the opinion of professional and amateur photographers in addition to photography literature. They provide mathematical formulations to calculate the features from real photographs. Using Naive Bayes, they show that high-level features can successfully classify professional photographs.

Moorthy *et al.* [16] describe an experiment for estimating aesthetic appeal of videos. They collect user ratings for video appeal, and train support vector machines to classify videos as "good" or "bad". They hierarchically construct video features from individual frames. This thesis presents a similar experiment for modeling aesthetics of synthetic images, but evaluates image pairs instead of rating individually.

Yeh *et al.* [25] propose an online ranking system for personal photo collections based on aesthetic rules. In their system, features are computed for composition, color and intensity. Users provide their own images, adjust weights for the importance of different features, and can search for similar photographs. This thesis demonstrates how qualitative models can be built directly from image preferences.

Yannakakis [23] suggests a protocol for preference learning that is sensitive to personal affect. In his system, pairs of alternative choices are presented and evaluated by questionnaire. His model uses a "four-alternative forced choice" (4AFC) rating system for generalization across different subjects. This thesis adopts his strategy of pairwise comparisons, by collecting image preference data in 4AFC format, to construct the most general model of quality.

Chapter 3

The Panorama Framework

The Panorama framework is designed to support photographic composition research. It is motivated by the challenging problem of computationally evaluating image quality. The objective is to simulate photography in a virtual setting, so image data can be efficiently generated. At the heart of the framework is a game that serves as the interface for user interaction and image generation. To the player, the game helps to interactively learn the represented photography rules. To the researcher, the game provides image data from a controlled environment, averting the need to physically collect domain-specific photographs. Fundamentally, the framework provides a novel image-preference dataset that has applications for machine learning.

This chapter begins with a high-level description of the architecture, then describes the design and implementation of the Panorama game, and concludes with how image data are represented and annotated in an image corpus.

3.1 Architecture

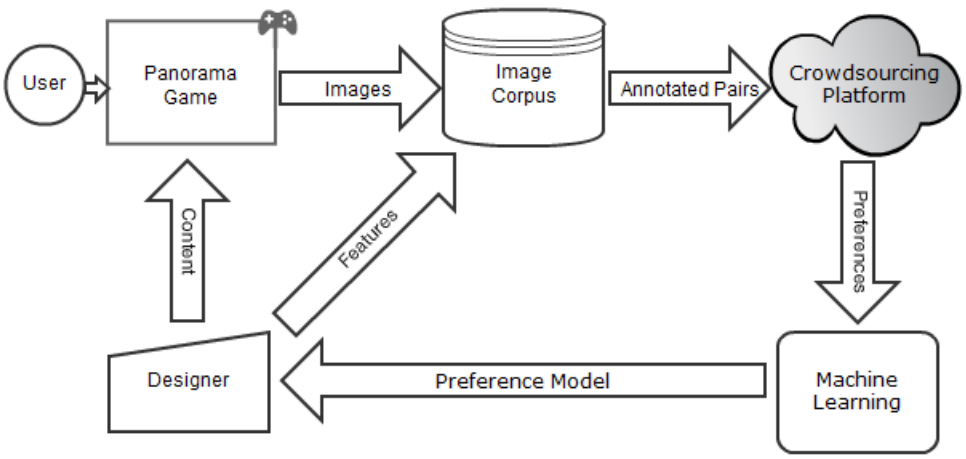

Figure 3.1: The Panorama framework

The Panorama framework is comprised of several interconnected components, as illustrated by Figure 3.1. The system begins with the user who interacts with the Panorama game (Section 3.2). Gameplay produces synthetic images, which are automatically annotated for composition features and contribute to the image corpus (Section 3.4). Image pairs are selected from the corpus and evaluated for relative preference, provided by external users through a crowdsourcing platform (Section 4.3). The pairwise preference data are saved in the corpus, and are used to

train machine learning algorithms (Section 4.4). An experiment designer uses the machine-learned preference model to inform adjustments to the game and selection of features (Section 4.2). Consequently, the next time a user plays the game, there will be new content that explores different dimensions of photograph learning.

The Panorama game is a major component of the framework, as it determines the characteristics of the artificial images being studied. This thesis describes the particular implementation of the game component, and reports on the effectiveness of using machine learning to predict human preferences of the photographs produced. The described experiment occurs in two distinct phases. First, one group of users plays the Panorama game to generate images, then another external group of users provides their ratings. Since the preference data are crowdsourced to anonymous workers, this model examines the general opinion of quality. Section 4.5 provides experimental results on how well preference can be predicted based on certain composition features.

The primary contribution of this framework is the construction of the image corpus. The Panorama game facilitates production of images with a common visual theme, and affords automatic annotation of features given the game's internal representation. The content of the images is influenced by the game designer such that new features are introduced in a controlled manner. As a result, the image corpus constantly grows in size and variety as users play new levels.

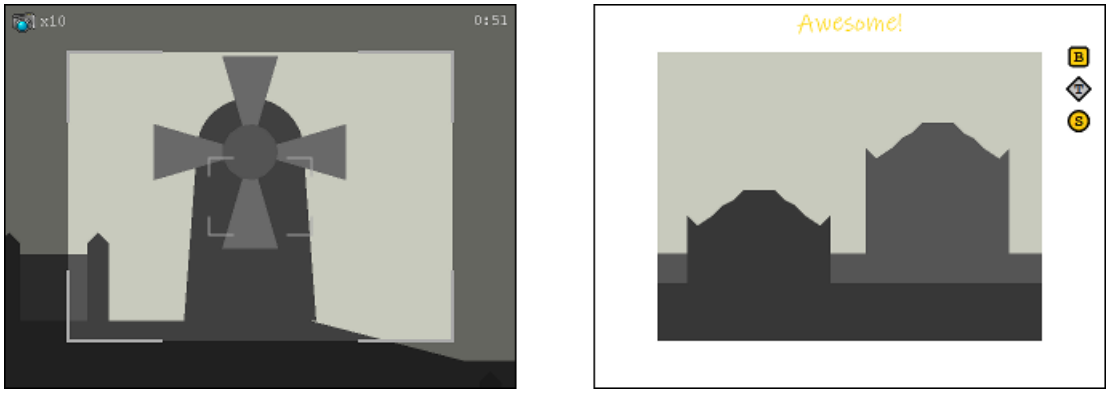

Figure 3.2: Panorama game interface (left) and picture-taking feedback (right)

The data of the Panorama corpus are made publicly available[1] for future researchers interested in preference learning and domain-specific image analysis.

3.2 Panorama Game

The Panorama game, shown in Figure 3.2, provides an interactive environment where a player (the photographer) takes pictures of a virtual landscape and is given feedback on select composition criteria. The goal is to gain insight on how to frame pictures effectively as a player progresses through levels of increasing complexity. This design facilitates the generation of customized image data, for the purpose of learning photographic preferences as done in Chapter 4. Synthesizing images in this manner benefits researchers for the ability to easily measure features of the

[1]The Panorama project, including the game, source code, image corpus, and preference data, is available on the UCSC Games department website (http://games.soe.ucsc.edu/project/panorama).

image's underlying representation. For this reason, every image is automatically annotated for its visual contents and saved in an external corpus (Section 3.4).

Panorama is inspired by a variety of genres, from music games like *Rock Band*—where aspiring musicians are scored for matching notes while playing familiar instruments—to traditional games like *GNU Backgammon*—where players are tutored on backgammon moves using computer analysis. Like *Rock Band*, Panorama fulfills the player's fantasy of performing a real activity by eliciting complementary skills, while also, like *GNU Backgammon*, it evaluates the player's performance using computer-learned heuristics. Furthermore, rules are improved by collecting data.

Mechanics

The mechanics of Panorama are similar to games like *Afrika* and *Pokémon Snap*, where the player assumes the role of a photographer and has control of the camera's view. However, in Panorama the objective is not to photograph specific subjects, but to "frame" them artistically. The player has three primary controls that are used to frame a shot, which have the following effects (Figure 3.3):

- Pan—move horizontally or vertically within the scene
- Zoom—move closer or farther from the subject
- Tilt—rotate right (clockwise) or left (counter-clockwise)

Figure 3.3: Camera control: pan, zoom, and tilt

When the player has finished framing the shot, they may snap a picture, which is the game's core mechanic. Once a picture is taken, it is saved in the Panorama corpus (Section 3.4), and scored for the chosen composition criteria (Section 3.5). The player gets immediate feedback about their composition scores to influence their strategy for future actions. The player may view their personal collection of images and scores in the game's main menu.

Each of the player's camera controls are bound to a fixed range. Panning is confined to the boundary of the scene. Zooming may enlarge the scene to no more than four times, and shrink it to no less than one-fourth of the original size, to limit how much of the scene can be viewed at a time. Tilting is restricted to angles less than 45 degrees from the initial flat inclination, to prevent sideways or upside-down images in the dataset, inconsistent with landscape photography. Controls are further limited in special "scrolling" levels, where the camera parameters are influenced by a fixed animation path defined by the level. These control mechanics, combined with level design and game interface, create challenge for the player as described in the following section.

Gameplay

The player's objective in Panorama is to take high-quality pictures of abstract two-dimensional landscapes, where quality is evaluated by compositional rules, defined in Section 3.5. In the game, the player views scenery through an on-screen viewfinder (Figure 3.2), which helps to align the intended shot. The scene is displayed in first-person perspective, but only the part of the screen outlined by the viewfinder is captured in each photograph. Players explore different landscapes in the game, which are presented in the form of levels (Figure 3.4). On each level, the player is given a fixed number of pictures, and completes the level once the quantity has been exhausted. The levels in Panorama are designed to introduce new visual features incrementally, and to progress the challenge of photographing more complex scenery. This is a decision to both engage the player and control the types of images being produced, allowing the researcher to collect data with specific properties.

While there is no cumulative reward for completing a level, the player is rewarded with colored "badges" after each picture is taken. One colored badge is given for each composition criterion scored in the game, and comes in four varieties. A gold badge indicates the highest achievement, then silver, then bronze, and finally red. These discrete achievements help the player to maximize their objective, while hiding the specific calculation that might be exposed by numeric scores. Image scoring is further generalized by an overall rating that is given as

Figure 3.4: A typical game level

narrative feedback. This feedback is presented graphically when each picture is taken (Figure 3.2), along with badges for three specific criteria: balance (B), rule of thirds (T) and spacing (S). These composition rules and how they are computed are discussed in Section 3.5. Note that no additional feedback is given to suggest how to improve along the scored dimensions. This was intentionally designed to encourage learning by experimentation, which provides more diverse images.

3.3 Implementation

Panorama is implemented for Microsoft Windows and Xbox 360 platforms. It uses the XNA Framework and is designed for minimal hardware specifications. Panorama emphasizes simplicity and efficient rendering techniques to run on the largest range of modern systems. This section describes three important concepts for creating the geometry in the game: the low-level mathematical structure, the representation of objects, and the high-level construction of a scene.

Hypercurve

The fundamental building block for much of the content in Panorama is a mathematical object referred to as a *hypercurve*. This structure represents an n-dimensional parametric equation of one parameter. It is primarily useful for numerical interpolation, both for scripting camera paths and animations as a function of time, and for parametrically expressing primitive objects as vector graphics.

A hypercurve is represented as a vector of cubic Hermite splines. These splines are constructed from a set of key frames, and are interpolated over a continuous interval. Tangents are given at the control points to influence how the function behaves between key frames. Specifically, linear tangents are used to represent polygons. Finite difference tangents are used to create smooth camera motion in non-uniform intervals.

Hypercurves are useful for representing transformations of any real-valued quantity in $\mathbb{R}^n$, such as position in Euclidean space. This is good for animating camera parameters in two dimensions, but does not apply to three-dimensional camera rotations. Extending Panorama to 3D would require adjusting the hypercurve to allow for interpolating orientations, as described by Shoemake [21]. Kim *et al.* [11] give the appropriate formulation of Hermite splines in rotational space, using quaternions for key frames and tangent angular velocities.

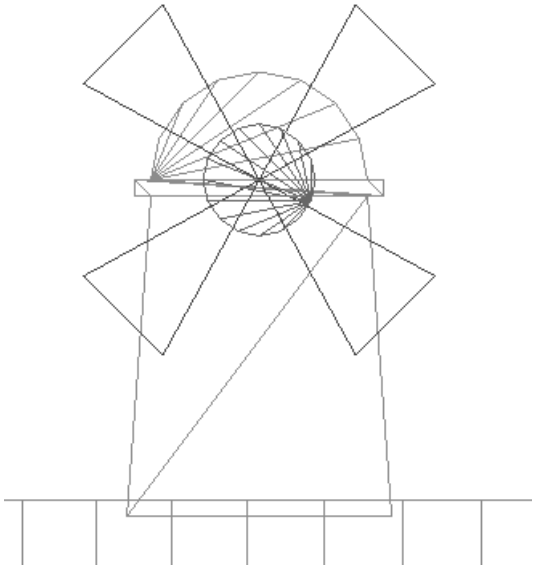

Figure 3.5: Wireframe rendering of a windmill object

Objects

The graphical objects in Panorama are created entirely from triangles. Rendering from geometric primitives facilitates procedural generation of content, looks good under transformation, and is computationally efficient.

The objects—buildings, trees, fences, and windmills—are composed of polygons. Each object has different aspects of random variation: buildings—height, width, roof pattern; trees—height, number of layers; fences—width, number of posts; and windmills—height, width, blade size. These variations do not have a specific aesthetic meaning. Functionally, only height and width matter for measuring composition. Ultimately, polygonal objects are triangulated by ear clipping using Eberly's implementation [20]. The sub-optimal $O(n^2)$ algorithm is

sufficient because objects typically have fewer than 100 vertices, and only need to be triangulated once. Figure 3.5 shows a windmill object with three polygonal layers, rendered in wireframe for illustration.

Curved shapes are mathematically modeled using two-dimensional hypercurves. The hypercurve is sampled to build piecewise linear polygons of varying resolution. This is important in providing level of detail. For example, the vector representation of a circle is a hypercurve approximating the parametric equation for x and y: $(r\cos t, r\sin t)$. If zoomed out, it might be sufficient to sample the curve in eight equal intervals to generate an octagon; but when zoomed in, the same circle can be rendered as a 64-sided polygon.

Hypercurves are not sufficient for representing surfaces in three-dimensional environments. An analogous mathematical structure for parametric equations of two parameters is required. NURBS [18] are commonly used to model such parametric surfaces in three dimensions. However, to be consistent with the design of the hypercurve, the precise counterpart, *hypersurface*, would use bicubic Hermite interpolation in which the surface passes through the control points. The contours of such a surface would be determined by tangents with respect to each parameter and a second-order "twist" with respect to both.

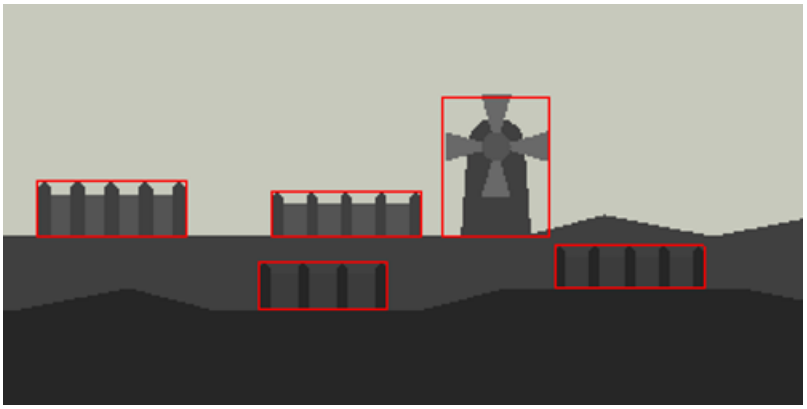

Figure 3.6: Hidden bounding boxes for measuring composition

Scenes

The scenes in Panorama are comprised of overlapping layers, and an off-white backdrop representing the sky. Each layer is a different shade of gray, to be easily distinguishable. Scenes vary in size (how wide they are), number of layers, and contents. The content of a layer includes a fixed ground surface, and a set of objects added on top. Which objects, their location, and stylistic variation may be constant or stochastically determined at run time. For example, the first level is always the same single building, but the third level is a composite of trees and buildings with pseudo-random positioning. This is done to keep the game "interesting". If all scenes were fixed, then the optimal choices for framing could be solved. Furthermore, procedural content generation can be used to customize levels to the player's personal preference, as proposed for future work (Section 5.1).

Each object in the scene has a hidden bounding box representing the position and size (Figure 3.6). These bounding boxes provide the main representation for images in the Panorama framework. The only information logged and scored in the annotation process (Section 3.5) is the layout of the bounding boxes. Because the imagery is abstract, it is believed that layout must significantly contribute to determining quality. Thus bounding boxes are minimal information for photographic composition research, which is not concerned with semantics. Functionally, bounding boxes are used in the game for collision detection, to determine which objects appear in the photograph. For tilted photographs, the angle of inclination is required, and intersection is tested using the separating axis theorem.

All scenes in Panorama are stationary, and once constructed, the geometry is static. This has the advantage of buffering the data in graphics memory. Transformation of the geometry and minor animations are handled by shaders to avoid continuously streaming vertex data to the GPU. The player's control of the camera viewpoint is strictly a manipulation of a global transformation matrix. Each layer has its own world matrix that is dependent on the global transformation. All rotations and scales (tilts and zooms) are propagated identically to the descendant layers. Translations (pans) are dampened across layers to produce a parallax scrolling effect. Moving the layers at different speeds creates an illusion of depth, and gives the player extra control for aligning objects.

To determine what objects appear in an image, the local transformation is applied to the bounding boxes in each layer, and tested for collision with the viewport before the data are exported. The following section describes the image corpus component of the framework, which hosts all of the data created by the Panorama game.

3.4 Image Corpus

The Panorama corpus is a database of artificial photographs created by the Panorama game. It is explicitly designed for machine learning research in image composition and evaluation. The corpus provides two main advantages to the research community.

First, the images are *domain-specific*. The corpus hosts a large collection of images that are generated from the same procedure. As such they share similar characteristics; they are abstract, grayscale landscapes with limited contents, constrained by gameplay mechanics. Analyzing images with such commonalities is beneficial for measuring preferences, because they share a similar context. In contrast, real photographs, which are not taken under the same conditions, are difficult to compare because they have varied subject matter. Manually finding sufficient examples from a single source is cumbersome, and thus the novel domain is the first contribution of the Panorama corpus.

Second, the images are automatically *annotated* for compositional features. To reason about image preferences, it is necessary to know the contents. For real photographs, the images must be either manually annotated, or processed by computer vision algorithms. This preprocessing step can be labor-intensive and imprecise. But for synthetic images, annotation can be provided directly by the generating system. Indeed, the Panorama corpus is constructed from gameplay image data in which a mathematical, bounding-box representation for denoting composition is provided for each image. This representation facilitates calculation of custom features for machine learning (Section 4.2). Furthermore, the conditions of the game under which the image was taken is also logged.

Three kinds of data are represented in the Panorama corpus:

- A rendered bitmap of each gameplay image. Bitmaps are stored in PNG format, with dimensions 640×480 pixels.
- A list of bounding boxes for objects in each image. Bounding boxes are represented as two axis-aligned points for the top-left and bottom-right extents of the object. Coordinates are measured with respect to the center of the image, so values range from $x \in [-320, 320)$ and $y \in [-240, 240)$.
- A summary of contextual information for each image. This includes what game level the image was taken in and what camera parameters were used (location, zoom, and tilt).

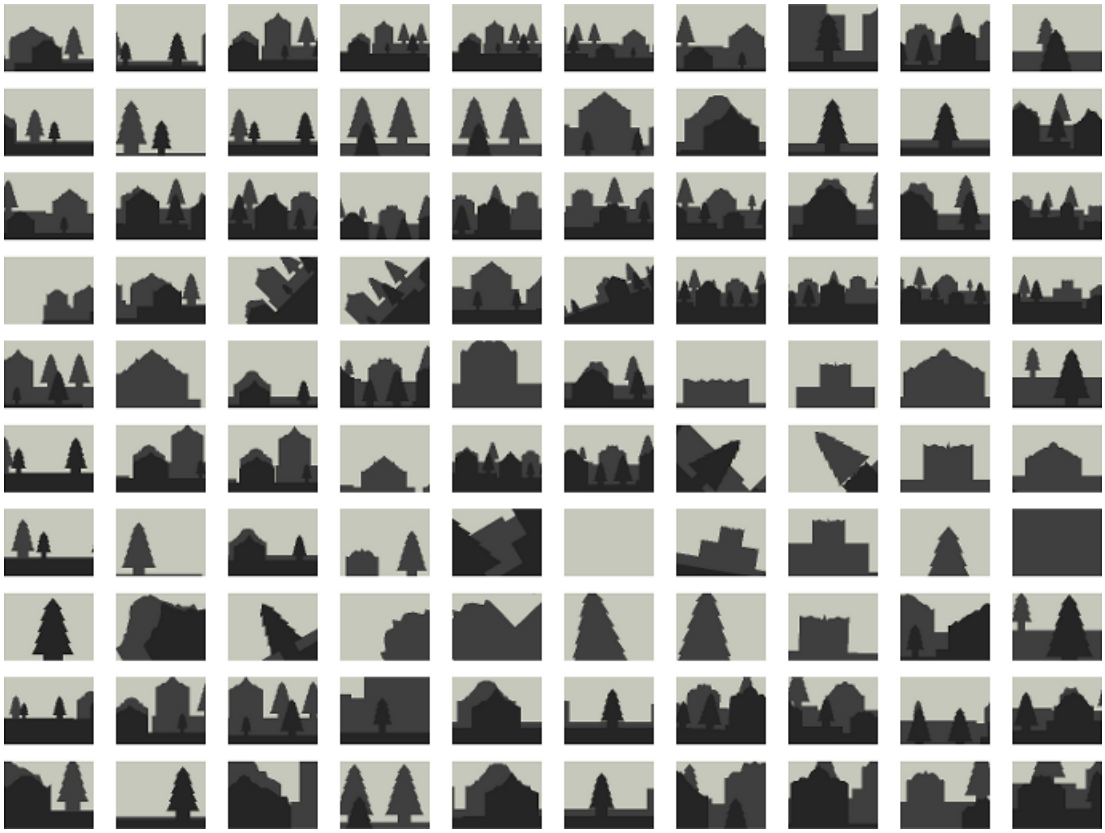

Figure 3.7: Images produced in gameplay

3.5 Evaluation

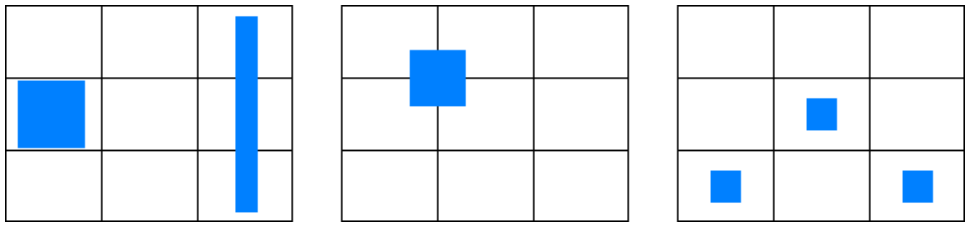

Figure 3.8: Good balance (left), rule of thirds (center), and spacing (right)

The Panorama game implements three photographic composition criteria for evaluating pictures during gameplay—*balance*, *rule of thirds*, and *spacing* (Figure 3.8). These criteria are inspired by photography literature, and are designed to help the player improve along different dimensions of framing. Each composition criterion is automatically scored from a bounding-box representation of the photo. The three scores are then combined to give the player an overall rating of their photograph.

Representation

The benefit to using a virtual environment is that information is accessible from the underlying model of the scene. Because images are generated synthetically, the contents are explicitly defined and can be reasoned about. Photographic composition research in particular is interested in the layout of elements.

To address layout mathematically, it is useful to simplify the image representation to a set of bounding boxes indicating the size and location of the visual elements.

As mentioned in Section 3.4, a vector of rectangles annotates images produced in the Panorama game. These rectangles are used for both evaluating the composition of pictures during gameplay, and also computing features for machine learning (Section 4.2). Each rectangle indicates the corners of a bounding box for an object that appears in the picture. This representation is used to calculate the compositional scores defined in the following sections. The area, a_i, and center, $\mathbf{c}_i$, of each object $i \in [1, n]$ is inferred from its bounding box, obj_i. All points are measured in pixels with respect to the origin at the center of the frame.

Balance

Balance is a measure of where the objects appear in the frame. For multiple objects, an average position somewhere in-between is considered. Good balance occurs when the weighted average position of all objects tends toward the center of the frame. To compute this, Panorama finds a center of mass using object area for weights. The balance score, B, is the length of the center of mass, $\mathbf{M}$, normalized by one half of the diagonal length of the frame.

$$\mathbf{M} = \frac{\sum a_i \mathbf{c}_i}{\sum a_i} \tag{3.1}$$

$$B = 1 - \frac{2 \times |\mathbf{M}|}{\sqrt{\text{width}^2 + \text{height}^2}} \tag{3.2}$$

Rule of Thirds

Rule of thirds is a measure for how well the objects are aligned to invisible lines called *thirds lines*. The thirds lines partition the frame into a 3×3 grid of equal sizes. Photographers say that positioning the target on these lines makes for better composition [7]. Panorama evaluates rule of thirds by computing the distance between the centers of each object to each thirds line.

The function δ is used to find the distance of a point, $\mathbf{p}$, to the nearest thirds line along an axis, a, normalized by one-third of the frame size, $\mathbf{F}$. The rule of thirds score, T, is the average distance of each element's centroid to the two nearest thirds lines.

$$\delta(\mathbf{p}, a) = 3 \min_{t:\pm 1/6} |\frac{\mathbf{p}_a}{\mathbf{F}_a} - t| \tag{3.3}$$

$$T = 1 - \frac{1}{2n} \sum \delta(\mathbf{c}_i, x) + \delta(\mathbf{c}_i, y) \tag{3.4}$$

Spacing

Spacing is a measure of the area between and around the objects in a picture. Good spacing depends on the context of the photograph. For this kind of landscape photography, spacing score is determined by the placement of the horizon (ratio of sky to ground), in addition to the amount of area occupied by objects. Panorama considers at two kinds of spacing: horizontal and vertical.

Horizontal spacing is the ratio of empty space between left and right halves of the frame. Empty space is computed by subtracting the combined area of bounding boxes from each region. Horizontal spacing, H, is good when the ratio is uniform. Let L and R be the left and right regions when vertically dividing the frame in half:

$$\text{left} = \frac{1}{2} \times \text{width} \times \text{height} - \text{area}(L \cap \bigcup_i \text{obj}_i) \tag{3.5}$$

$$\text{right} = \frac{1}{2} \times \text{width} \times \text{height} - \text{area}(R \cap \bigcup_i \text{obj}_i) \tag{3.6}$$

$$H = \frac{\min(\text{left, right})}{\max(\text{left, right})} \tag{3.7}$$

Vertical spacing is a ratio between sky and ground. The optimal position of the horizon is considered to be on the bottom thirds line. Therefore, vertical spacing, V, is good when the area above the horizon is twice the area below:

$$\text{sky} = \frac{\text{height}}{2} + \min_i(\text{obj}_i.\text{bottom}) \tag{3.8}$$

$$\text{ground} = 2 \times (\frac{\text{height}}{2} - \min_i(\text{obj}_i.\text{bottom})) \tag{3.9}$$

$$V = \frac{\min(\text{sky, ground})}{\max(\text{sky, ground})} \tag{3.10}$$

The total spacing score, S, is the average of horizontal spacing, H, and vertical spacing, V.

$$S = \frac{H + V}{2} \tag{3.11}$$

Score

When a player takes a picture in Panorama, they get feedback on the three composition criteria, and are given an overall score. For *balance*, *rule of thirds*, and *spacing*, feedback comes in the form of colored badges: gold (awesome), silver (great), bronze (good), and red (bad). Similarly, the overall quality is rated from "bad" to "awesome" and appears in large writing when the picture is taken.

The four scoring classes are enumerated by integer values from 0 to 3. These discrete labels are computed from the real-valued composition measurements, by taking a linear threshold with respect to four consecutive, non-overlapping intervals.

$$\text{rate}(x, a, b, c) = \begin{cases} 3 & \text{if } x \geq a \\ 2 & \text{if } x \in [b, a) \\ 1 & \text{if } x \in [c, b) \\ 0 & \text{if } x < c \end{cases} \tag{3.12}$$

For each composition criterion, the thresholds are $a = 0.9, b = 0.7, \text{and } c = 0.5$. The total score is a function of the individual scores, where the classes have different weights: $\mathbf{w} = [0, 2, 3, 4]$. The sum of weighted scores is determined by the thresholds: $a = 9, b = 6, \text{and } c = 3$.

$$\text{total} = \sum_{i \in \{B,T,S\}} \mathbf{w}[\text{rate}(x_i, 0.9, 0.7, 0.5)] \tag{3.13}$$

$$\text{score} = \text{rate}(\text{total}, 9, 6, 3) \tag{3.14}$$

Hence, the overall score is a linear combination of the three individual scores. In the future, scoring could potentially be handled by an artificial neural network where the weights and thresholds are learned from player feedback.

Evaluation Example

This example shows the mathematical steps of how to score an image (shown in Figure 3.9).

1. Compute B.

$$\begin{aligned}
\mathbf{M} &= \frac{25{,}200 \times (-175, -90) + 106{,}000 \times (97.5, 15)}{25{,}200 + 106{,}000} \\
\mathbf{M} &\approx (45, -5) \\
B &= 1 - \frac{2 \times |(45, -5)|}{\sqrt{640^2 + 480^2}} \\
B &\approx 0.89
\end{aligned}$$

2. Compute T.

$$\begin{aligned}
\delta(\mathbf{c}_1 : (-175, -90), x) &= 3 \times \min(|\frac{-175}{640} \pm \frac{1}{6}|) \approx 0.32 \\
\delta(\mathbf{c}_1 : (-175, -90), y) &= 3 \times \min(|\frac{-90}{480} \pm \frac{1}{6}|) \approx 0.06 \\
\delta(\mathbf{c}_2 : (97.5, 15), x) &= 3 \times \min(|\frac{97.5}{640} \pm \frac{1}{6}|) \approx 0.04 \\
\delta(\mathbf{c}_2 : (97.5, 15), y) &= 3 \times \min(|\frac{15}{480} \pm \frac{1}{6}|) \approx 0.41 \\
T &= 1 - \frac{1}{4}(0.32 + 0.06 + 0.04 + 0.41) \\
T &\approx 0.79
\end{aligned}$$

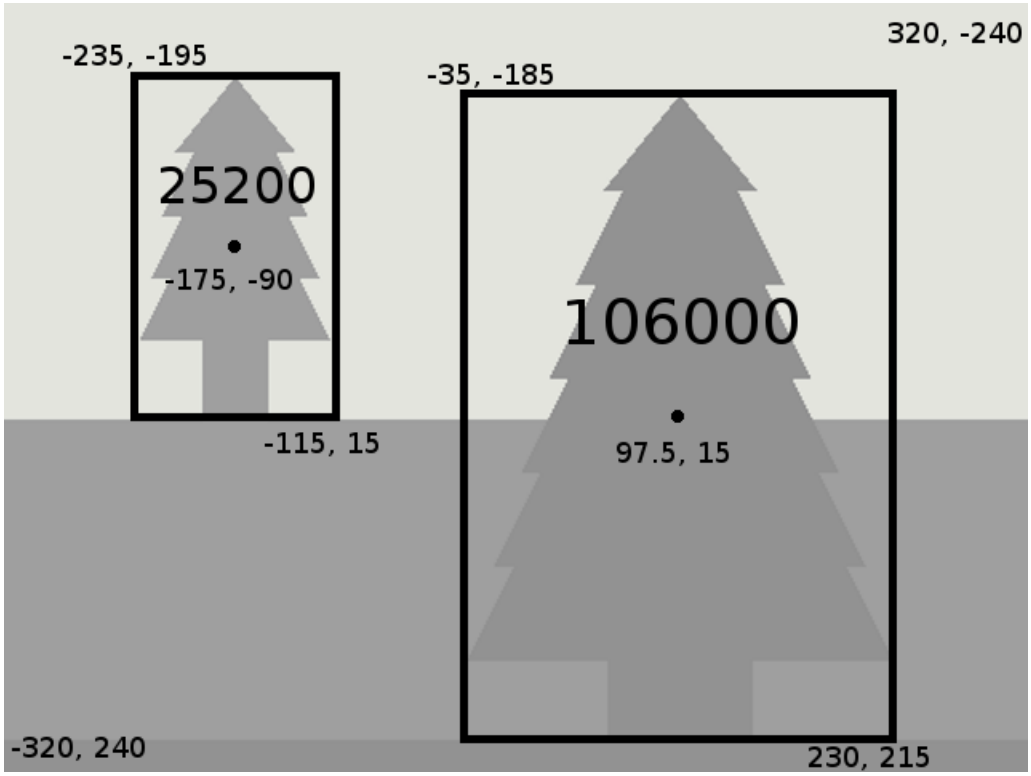

Figure 3.9: Example photo with annotations

3. Compute S.

$$\begin{aligned}
\text{left} &= \frac{1}{2} \times 640 \times 480 - 25{,}200 - \\
&\quad (0 - (-35)) \times (215 - (-185)) = 114{,}400 \\
\text{right} &= \frac{1}{2} \times 640 \times 480 - (230 - 0) \times (215 - (-185)) = 61{,}600 \\
H &= \frac{61{,}600}{114{,}400} \approx 0.54 \\
\text{sky} &= \frac{480}{2} + \min(15, 215) = 255 \\
\text{ground} &= 2 \times (\frac{480}{2} - \min(15, 215)) = 450 \\
V &= \frac{255}{450} \approx 0.57 \\
S &= \frac{0.54 + 0.57}{2} \approx 0.56
\end{aligned}$$

4. Compute ratings $b, t,$ and s.

$$\begin{aligned}
b &= \text{rate}(B : 0.89, 0.9, 0.7, 0.5) = 2 \text{ (great)} \\
t &= \text{rate}(T : 0.79, 0.9, 0.7, 0.5) = 2 \text{ (great)} \\
s &= \text{rate}(S : 0.56, 0.9, 0.7, 0.5) = 1 \text{ (good)}
\end{aligned}$$

5. Compute final rating.

$$\begin{aligned}
\mathbf{w} &= [0, 2, 3, 4] \\
\text{total} &= \mathbf{w}[b : 2] + \mathbf{w}[t : 2] + \mathbf{w}[s : 1] = 3 + 3 + 2 = 8 \\
\text{score} &= \text{rate}(\text{total} : 8, 9, 6, 3) = 2 \text{ (great)}
\end{aligned}$$

Chapter 4

Preference Learning

The Panorama framework provides a platform for synthesizing annotated photographs. These photographs are useful for image preference learning because they are abstract and their properties are well-defined. Given enough example images, it is possible to reason about how composition contributes to image quality. In an experiment to do so, many people are asked to judge the quality of images from the Panorama corpus. The chosen images are then automatically annotated for select compositional features. These composition-preference examples are used to train machine learning models, which estimate human judgment of arbitrary images based on the represented features.

This chapter begins by describing the studied images and how they were acquired from Panorama gameplay. Then, the visual features used for evaluation are described. Next, the crowdsourcing procedure for collecting preference ratings

is explained. Lastly, analysis is provided for two machine learning experiments, in which regression and classification using support vector machines are compared in different training configurations.

4.1 Image Synthesis

The first step of the preference learning experiment is collecting the appropriate images to be studied, accomplished by playing the Panorama game. In an informal user study, five individuals were invited to play the game. No profiling was done, however they were all students in the computer science department, aged 20–25, of whom one was female and four were male.

The participants were briefly instructed on how to play the game, including the controls, photo-taking objective, and scoring feedback described in Section 3.2. In independent sessions, they each played the game for a maximum of two minutes, then a thirty-second break, and another two minutes of play. The two gameplay intervals consisted of one specific level of the Panorama game (shown in Figure 3.4). The second interval was a repeat of the level, to see if a player noticeably improved or adapted their strategy from experience.

In total, five players played the Panorama game twice. Each time, they took ten virtual photographs to complete the level. So, the whole user study produced a dataset of one-hundred images, shown in Figure 3.7. These images are now available in the Panorama corpus and can be studied in future experiments.

The following sections describe how the images are annotated for composition features, rated for relative quality, and analyzed by machine learning.

4.2 Composition Features

The second step of the preference learning experiment involves identifying visual features that might contribute to quality. Photography literature provides some guidance in this regard by suggesting general rules of thumb [7]. Features such as *rule of thirds* provide a good starting point, because the human raters are likely to be cognizant of these principles. However, the rules provided by photography experts are not comprehensive, so new features are considered and validated by experimentation.

The purpose of feature selection is to consider all things that might contribute to deciding quality. For real photographs, the complexity is overwhelming. However, images from the Panorama corpus simplify the problem, because many considerations like color, exposure, and depth of field do not apply. This experiment considers how composition contributes to image preference, and is exclusively concerned with the layout of elements. Thus, potential features are those which can be calculated from the bounding box representation of the images, which is given in the dataset.

In this experiment, seven primitive features are defined, listed in Table 4.1. They are computed in a similar manner to the Panorama game scoring criteria,

Feature		Description
Symmetry	S	Distance from the image's center of mass to four axes: $x = 0$, $y = 0$, $x = y$, and $x = -y$
Thirds	T	Average distance of each object's centroid to the nearest thirds line on both axes
Spacing	Q	Binary indication that a region of space is not empty, as partitioned by a quadtree of depth four
Objects	N	Number of objects
Crops	C	Number of objects cropped by the edge of the frame
Occlusions	O	Number of objects that intersect another object
Size Ratio	R	Area of the smallest object, divided by the largest object

Table 4.1: Composition features used for preference learning

described in Section 3.5. These particular features were thought to be the most salient, and chosen for ease of implementation and conceptual understanding. Logically, many other features exist from other sources, such as camera parameters (zoom, tilt) or image analysis (brightness, contrast, leading lines). However, such features require additional data or preprocessing of the rendered bitmap, which is outside the scope of this experiment.

In addition to the seven modeled features, there are countless ways to process an input image to construct arbitrary features. However, most would be unintelligible, and feature extraction is not the goal of this experiment. Instead, it is shown that preferences can be learned fairly well with just these seven decompositions. As demonstrated in Section 4.4, the machine learning algorithms can naturally reason about higher-dimensional patterns, by using a transformation of the feature space using kernel methods.

The features used in this experiment represent simple concepts, and are named as follows: *symmetry*, *rule of thirds*, *spacing*, *objects*, *crops*, *occlusions*, and *size ratio*. Descriptions and mathematical formulations of these features are given in the following subsections.

Symmetry

The *symmetry* feature, S, is a generalization of the balance score calculated by the Panorama game (Section 3.5). Instead of measuring balance with respect to the center, symmetry considers the distance from the image's center of mass to four axes: $x = 0$, $y = 0$, $x = y$, and $x = -y$. Thus symmetry, S, is a four-dimensional vector, calculated as follows:

Recall that center of mass, $\mathbf{M}$, is defined as the sum of objects' centers, weighted by area (Equation 3.1). Given frame width, w, and height, h,

$$s_{x=0} = \frac{2\,|\mathbf{M}.x|}{w} \tag{4.1}$$

$$s_{y=0} = \frac{2\,|\mathbf{M}.y|}{h} \tag{4.2}$$

$$s_{x=y} = \frac{2\,|\mathbf{M}.x - \mathbf{M}.y|}{\sqrt{2(w^2+h^2)}} \tag{4.3}$$

$$s_{x=-y} = \frac{2\,|\mathbf{M}.x + \mathbf{M}.y|}{\sqrt{2(w^2+h^2)}} \tag{4.4}$$

$$S = (s_{x=0}, s_{y=0}, s_{x=y}, s_{x=-y}) \tag{4.5}$$

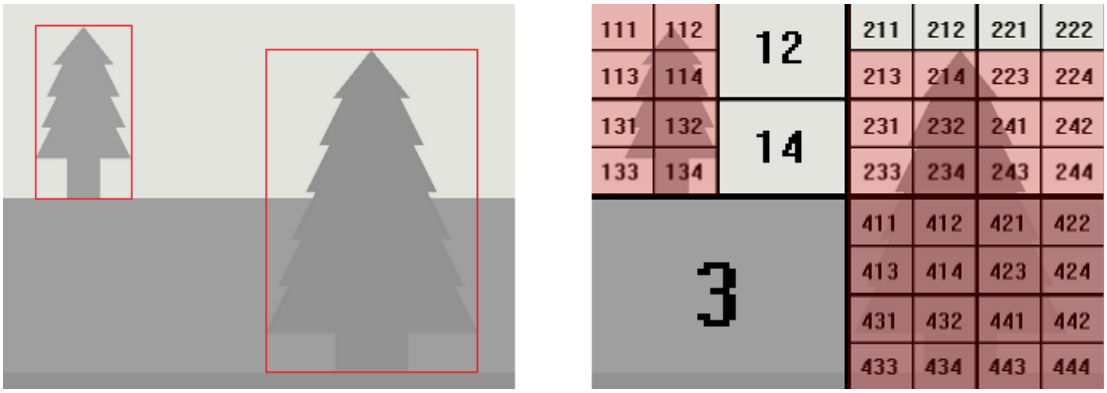

Figure 4.1: Quadtree decomposition of image space

Rule of Thirds

The *rule of thirds* feature, T, measures how well visual elements are aligned to the *thirds lines*, which partition the frame into a uniform 3×3 grid. This feature is identical to the one scored by the Panorama game, and the equation is given in Section 3.5. Rule of thirds, T, is a single scalar value, ranging from 0 (bad alignment) to 1 (good alignment).

Spacing

The *spacing* feature, Q, measures the "emptiness" of the frame. It is represented as a string of bits indicating whether certain regions of the frame are empty. The regions are determined by a quadtree decomposition of depth four (Figure 4.1). In the quadtree, the first region is the entire frame, then the four quadrants are evaluated recursively. The feature takes the value 0 if a region is

empty, and 1 otherwise. This is an efficient way of calculating spacing, as an empty frame is only evaluated once. The process terminates after three subdivisions, and thus the complete tree can be represented in $\frac{4^4-1}{3} = 85$ bits. Mathematically, it is defined as follows:

Let the function q_0 indicate whether some element, e, in the set of all elements, E, intersects a given rectangle, $\mathbf{r}$. The function q_{n+1} recursively divides rectangle $\mathbf{r}$ into four quadrants: top-left, $\mathbf{r}_{tl}$; top-right, $\mathbf{r}_{tr}$; bottom-left, $\mathbf{r}_{bl}$; and bottom-right, $\mathbf{r}_{br}$. The spacing feature, Q, is the string of bits generated from the quadtree function, q_3, for frame rectangle, $\mathbf{F}$.

$$q_0(\mathbf{r}) = \begin{cases} 1 & \text{if } \exists e \in E : e \cap \mathbf{r} \neq \emptyset. \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

$$q_{n+1}(\mathbf{r}) = (q_0(\mathbf{r}), q_n(\mathbf{r}_{tl}), q_n(\mathbf{r}_{tr}), q_n(\mathbf{r}_{bl}), q_n(\mathbf{r}_{br})) \tag{4.7}$$

$$Q = q_3(\mathbf{F}) \tag{4.8}$$

Objects

The *objects* feature, N, is the number of elements in set E, normalized by $\frac{1}{10}$.

$$N = \min(1, \frac{|E|}{10}) \tag{4.9}$$

Crops

The *crops* feature, C, is the number of elements in set E that are not fully within frame rectangle $\mathbf{F}$, normalized by $\frac{1}{10}$.

$$C = \min(1, \frac{|\{e : e \cap \mathbf{F} \neq e\}|}{10}) \tag{4.10}$$

Occlusions

The *occlusions* feature, O, is the number of unique intersections between two elements in E, normalized by $\frac{1}{10}$.

$$O = \min(1, \frac{|\{e_1 \cap e_2 : e_1 \neq e_2 \wedge e_1 \cap e_2 \neq \emptyset\}|}{10}) \tag{4.11}$$

Size Ratio

The *size ratio* feature, R, is the ratio of minimum to maximum area, a_i, of all elements $i \in [1, n]$.

$$R = \frac{\min_i a_i}{\max_i a_i} \tag{4.12}$$

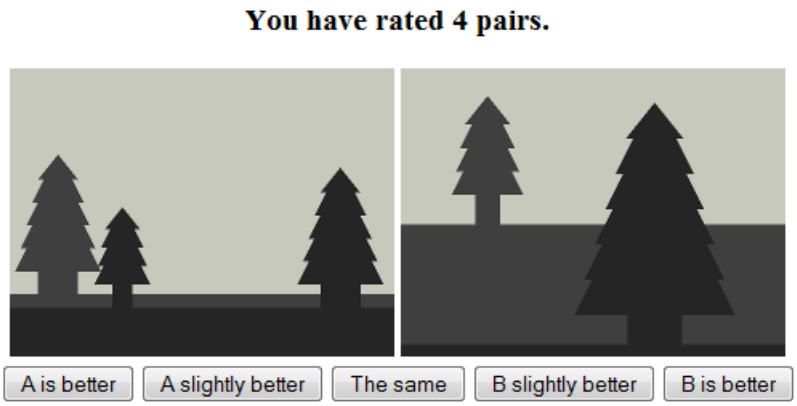

Figure 4.2: Web form for collecting pairwise image preferences

4.3 Crowdsourcing Preferences

The third step of the preference learning experiment is collecting preference ratings for the images to establish ground truth. Because quality is subjective, enough people have to be sampled to get a sense of the general opinion. Furthermore, a lot of redundant information must be collected to account for varying beliefs and noise in polling. Crowdsourcing provides such a mechanism for sampling diverse populations.

This experiment collects data on two online platforms to acquire data from different sources and on different scales. First, 1000 ratings were collected from nine acquaintances on Facebook, via the "app" seen in Figure 4.2. Second, the process was crowdsourced to Amazon Mechanical Turk, where 24,000 ratings were collected from 36 anonymous workers. In both cases, none of the participants were

assumed to have specific knowledge of photography, and thus the ratings have a natural range of rating expertise.

In the two trials, relative preference was measured by presenting image pairs. It is believed that comparing alternative choices provides an accurate model of affect [23]. In addition, pairwise preferences are useful for ranking when inconsistencies are present [19]. So, participants were given two images and queried for preference. The images were drawn randomly from the 100 image sample (Section 4.1), so there were $\binom{100}{2} = 4,950$ possible pairs.

In the Facebook questionnaire, users were given a five-point Likert scale [12] on which to indicate pairwise preference. Given two images A and B, the options were: "A is better", "A is slightly better", "They are the same", "B is slightly better", and "B is better". This format allows the user to specify relationship with an indication of magnitude.

The Amazon Mechanical Turk questionnaire also measured pairwise preferences, but employed a different scale. Given two images A and B, the options were: "A is better", "B is better", "Both are good", and "Both are bad". This method is called four-alternative forced choice (4AFC) and has distinct advantages [23]. First, it eliminates the passive response "They are the same", which might be tempting for apathetic raters, and thus forces users to make a choice. Second, it introduces two new categories for equivalence, "Both good" and "Both bad", which can be used to determine substantive labels. That is, in the Likert system,

only relationship is learned, but no notion of which images are good (if any) is captured. The 4AFC format allows the user to specify both relative and absolute quality.

4.4 Machine Learning

The final step of the preference learning experiment is using the data to train machine learning models for estimating preference. This was carried out in two phases. First was a "pilot study" using the 1000 preference ratings acquired from Facebook, in which a multilayer perceptron was trained for classifying responses. Then a more rigorous classification experiment was performed using the 24,000 preference ratings from Amazon Mechanical Turk, in which support vector machines (SVMs) were trained on a broader range of features.

Pilot Study

The pilot study used machine learning to predict image preference responses given in an online questionnaire (Figure 4.2). The experiment was treated as a multiclass classification problem, where target labels were the five possible ratings for a given pair of images. The objective was to estimate preference based on calculated features of each pair of compared images.

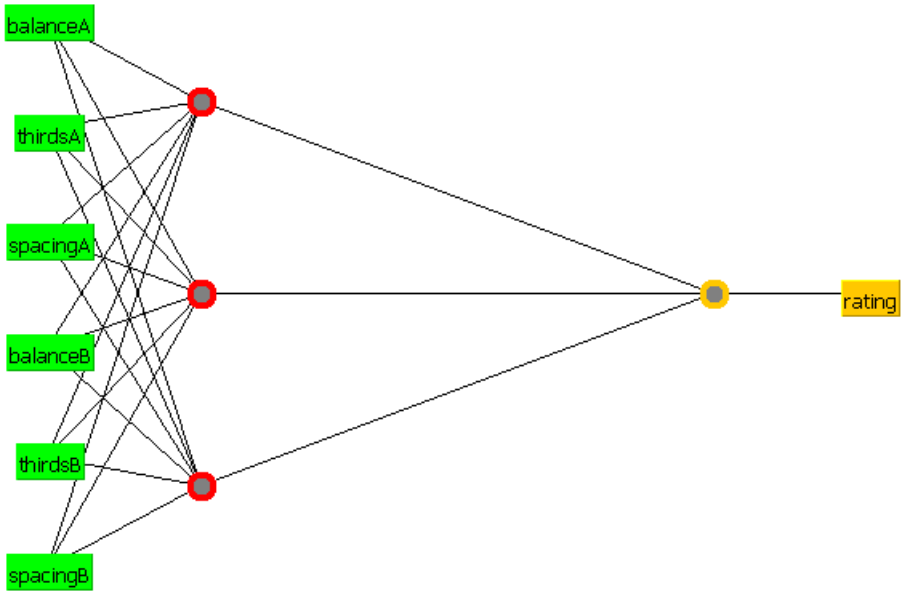

Figure 4.3: A multilayer perceptron for preference classification

This first experiment did not use the full list of features described in Section 4.2. Instead, it used the three scores from the Panorama game (Section 3.5): *balance*, *rule of thirds*, and *spacing*. This was designed to test the significance of the in-game scoring criteria. Since each image has three scores, the combined feature vector for image pairs had six dimensions.

A multilayer perceptron was constructed for classification using the Weka machine learning software [8]. The model was built using the default settings, and is shown in Figure 4.3. It consists of six input nodes, three hidden sigmoid nodes, and one linear-threshold output node. The network was trained using gradient descent, with a learning rate of 0.3 and momentum of 0.2. Under 10-fold cross validation, this model has a mean squared error of 1.3205.

Next, a linear regression model was trained in Weka, to analyze the effect of the chosen features. Mapping the response classes to integers from 0 to 4 produced the following model:

$$\begin{aligned} \textit{rating} \quad = \quad & -0.8519 \times \textit{balanceA} + 0.5473 \times \textit{spacingA} + 0.8529 \times \textit{balanceB} \\ & - 0.7313 \times \textit{thirdsB} - 0.6045 \times \textit{spacingB} + 2.5011 \end{aligned}$$

As expected, the coefficients for features of A are opposite to those for features of B, confirming that users indeed contrasted features in making decisions. However, the correlation toward final rating is low (0.2198), indicating that the represented features are insufficient. Like the perceptron classifier, the error for regression was quite high (1.1488 MSE), suggesting that the choice of features and data needed to be reexamined.

Lessons Learned

Overall, the initial pilot experiment did not produce strong results. The multilayer perceptron and linear regression models trained on 1000 preference ratings did not perform well using the three composition features from the Panorama game. This is likely because the machine learning techniques and data used were too minimal for the concept being modeled. Accuracy was low because of underfitting, given that so few image features (3 each) and relatively few training instances (100 images, 1000 preferences) were considered.

The primary lesson learned from the pilot experiment was that a more robust model, more features, and more training data would be necessary to tackle

the complexity of estimating image preferences. The model used for the pilot experiment didn't have enough resolution to explain how the input features contributed to the responses. So, a larger feature space is required, as well as more data to support the dimensionality increase. Thus the next experiment was devised with the following revisions:

1. An extensive dataset of 24,000 preferences collected from Mechanical Turk
2. High-dimensional feature space, including 95 features per image, and features for difference between images
3. Support vector machines with radial basis functions (Gaussian kernels) used for high-order classification

The following experiment uses the image features described in Section 4.2. While conceptually there are only seven, each image actually has 94 features, because *symmetry* and *spacing* are multidimensional (spacing is a string of 85 bits). As before, features of both images are considered for learning pairwise preferences. Additionally, the following experiment includes "delta" features for the difference of both images' features. Thus the complete feature vector has $94 \times |\{A, B, \Delta\}| = 282$ dimensions, of which there are $\binom{100}{2} = 4,950$ unique instances. The dataset includes 24,000 preferences, guaranteeing that there are redundant comparisons from different individuals.

Classification Experiment

This section describes a machine learning experiment to classify image preferences using support vector machines. Recall that the image set was synthesized from Panorama gameplay (Section 4.1), annotated for compositional features (Section 4.2), and rated for pairwise preferences by Amazon Mechanical Turk workers (Section 4.3). This section demonstrates the application of the data collection pipeline.

The objective of the experiment is to predict pairwise preferences along the 4AFC scale [23] for 24,000 ratings obtained by crowdsourcing. It is treated as a multiclass classification problem where the target classes are the four answers to the questionnaire, and the features are computed from each pair of input images. Three different support vector machines are trained using the LIBSVM software package [4]. These SVMs test different hypotheses of the preference data, and their accuracies are compared using cross-validation.

First, a four-class support vector machine is trained to predict the 4AFC preferences. The support vector machine uses a Gaussian kernel where the hyperparameters are selected by grid-search [9]. The SVM is trained on 65% of the data (15,797 instances), with hyperparameters $C = 2^5$ and $\gamma = 2^{-7}$. The feature vector consists of the features calculated for both images (Section 4.2) and the difference of those features. After training, the SVM is used to predict on

16% of the data (representing an 80-20 split) to achieve an accuracy of 57.0271% (2,252 of 3,949).

Next, a support vector machine is trained for binary classification using the first two classes of 4AFC. There were approximately 18,000 responses that answered one of these choices. The SVM is trained on 65% of the data (11,431 instances), with hyperparameters $C = 2^5$ and $\gamma = 2^{-7}$. After training, the SVM is used to predict on 16% of the data to achieve an accuracy of 76.1724% (2,209 of 2,900).

Lastly, a support vector machine is trained for binary classification, using a special subset of the data. The data for this experiment is chosen by a method similar to leave-one-out cross validation. One image is chosen at random, and all pairs using that image are removed from the training set. The remaining pairs using the image constitute the test set. This creates approximately a 90-10 split. The SVM is trained on 72% of the data (12,980 instances), with hyperparameters $C = 2^5$ and $\gamma = 2^{-7}$. After training, the SVM is used to predict on 8% of the data to achieve an accuracy of 65.8031% (889 of 1,351).

4.5 Results and Discussion

The three support vector machine classifiers developed in the preference learning experiment provide evidence for the successes and failures of the experimental design. The reported results provide a typical example of what can be expected from similar experiments using this framework.

Overall, the classification accuracy is quite good, though it is difficult to analyze why it is the case. The first SVM predicts multiclass preferences with 57.0271%, which is a substantial improvement over random classification of four classes (worst case 25% accuracy). Things to consider for improving accuracy would be better-designed features, or a different machine learning model.

The second SVM achieves the highest accuracy, 76.1724%, by predicting binary responses for the first two classes. This indicates that learning is dependent on the number of possible responses represented by the questionnaire. The improvement over the first SVM suggests that it is difficult to predict when users will specify the ambivalent responses in the 4AFC method.

The third SVM tests for overfitting by using cross-validation of preferences for certain input images. It performs binary prediction with 65.8031%, which is a more conservative estimate for how well the model performs. This number is most alarming, as it only slightly outperforms random prediction (50% accuracy). There are two likely reasons for this poor performance: insufficient image data and noise.

The two-step data collection process of generating images and collecting pairwise preferences has a notable limitation: increased data requirements. The amount of preference examples to adequately cover a set of images increases combinatorially for larger image sets. In such experiments, $\binom{n}{2}$ pairwise preferences are necessary for complete coverage of n images. While complete coverage is not strictly necessary, it is apparent that measuring preferences in this way does not scale well.

Furthermore, it is difficult to predict overall preference because there is significant noise in the training data. Because preferences are subjective, there will be inconsistencies in the data from users that provide conflicting responses for the same pair of images. For future work it will be useful to build personalized preference models, or use this dataset for collaborative filtering.

Chapter 5

Conclusion

This thesis has presented a framework for generating images from a computer game to study photographic composition. It has been shown how the images from gameplay may be paired with crowdsourced preference ratings to estimate quality. A machine learning experiment has been described to emphasize the benefit of the framework toward learning photographic composition preferences.

In summary, this research sought to learn how high-level photographic composition features contribute to perceived image quality. To achieve this, the Panorama framework was developed to collect tailored image data and show that it is possible to learn using crowdsourced ratings. A machine learning experiment was devised to provide evidence that high-level preferences can be predicted from compositional features computed from the dataset.

The primary contribution of this work is a new system for synthesizing images with well-defined visual features. The resulting image corpus is available to researchers and includes annotations for the contents of each virtual photograph. Furthermore, the system can be extended to account for new dimensions of image analysis, as is proposed for future work.

5.1 Future Work

The framework presented in this thesis is designed to be extensible. In the future, the Panorama game can be adapted to study different kinds of images by adding new graphical objects and scenes, introducing new features like color, lighting, texture, and depth of field, or even extending to realistic 3D environments to best approximate real photographs. Furthermore, the data of the Panorama corpus could represent semantic information by providing object labels and weights for different subjects.

New preference learning experiments can also be devised for the existing data. One direction of future research is to study subjectivity across groups of individuals using collaborative filtering. A potential application of such research is for personalization of photographic composition. Eventually, a personal model of image affect could be used for procedural content generation, or to drive cinematic camera planning systems [24].

References

[1] Rafid Abdullah, Marc Christie, Guy Schofield, Christophe Lino, and Patrick Olivier. Advanced composition in virtual camera control. In *Proceedings of the 11th international conference on Smart graphics*, SG '11, pages 13–24, Berlin, Heidelberg, 2011. Springer-Verlag.

[2] Serene Banerjee and Brian L. Evans. In-camera automation of photographic composition rules. *IEEE Transactions on Image Processing*, 16(7):1807–1820, 2007.

[3] William Bares. A photographic composition assistant for intelligent virtual 3d camera systems. In *Smart Graphics*, volume 4073 of *Lecture Notes in Computer Science*, pages 172–183. Springer Berlin / Heidelberg, 2006.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, May 2011.

[5] Yuan-Yang Chang and Hwann-Tzong Chen. Finding good composition in panoramic scenes. In *IEEE 12th International Conference on Computer Vision*, ICCV '09, pages 2225–2231, 2009.

[6] Simon Egenfeldt-Nielsen. *Beyond Edutainment Exploring the Educational Potential of Computer Games*. PhD thesis, IT-University of Copenhagen, 2005.

[7] Tom Grill and Mark Scanlon. *Photographic Composition*. Amphoto Books, New York, NY, USA, 1990.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[9] Chih-wei Hsu, Chih-chung Chang, and Chih-jen Lin. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2003.

[10] Yan Ke, Xiaoou Tang, and Feng Jing. The design of high-level features for photo quality assessment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR '06*, pages 419–426, 2006.

[11] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 369–376, New York, NY, USA, 1995. ACM.

[12] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):1–55, 1932.

[13] Ligang Liu, Renjie Chen, Lior Wolf, and Daniel Cohen-Or. Optimizing photo composition. *Computer Graphics Forum*, 29(2):469–478, 2010.

[14] Simon Lok, Steven Feiner, and Gary Ngai. Evaluation of visual balance for automated layout. In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI '04, pages 101–108, New York, NY, USA, 2004. ACM.

[15] Ryan McAlinden, Andrew S. Gordon, H. Chad Lane, and David Pynadath. Urbansim: A game-based simulation for counterinsurgency and stability-focused operations. In *Workshop on Intelligent Educational Games at the 14th International Conference on Artificial Intelligence in Education*, AIED '09, pages 41–50, Brighton, UK, 2009.

[16] Anush K. Moorthy, Pere Obrador, and Nuria Oliver. Towards computational models of the visual aesthetic appeal of consumer videos. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV '10, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag.

[17] Joan I. Nassauer. Framing the landscape in photographic simulation. *Journal of environmental management*, 17(1):1–16, 1983.

[18] Les Piegl and Wayne Tiller. *The NURBS book (2nd ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[19] Thomas L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, 1977.

[20] Philip J. Schneider and David Eberly. *Geometric Tools for Computer Graphics*. Elsevier Science Inc., New York, NY, USA, 2002.

[21] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '85, pages 245–254, New York, NY, USA, 1985. ACM.

[22] Hsiao-Hang Su, Tse-Wei Chen, Chieh-Chi Kao, Winston H. Hsu, and Shao-Yi Chien Chien. Preference-aware view recommendation system for scenic photos based on bag of aesthetics-preserving features. *IEEE Transactions on Multimedia*, PP(99), 2012.

[23] Georgios N. Yannakakis. Preference learning for affective modeling. In *3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, ACII '09, pages 1–6, 2009.

[24] Georgios N. Yannakakis, Héctor P. Martínez, and Arnav Jhala. Towards affective camera control in games. *User Modeling and User-Adapted Interaction*, 20:313–340, 2010.

[25] Che-Hua Yeh, Yuan-Chen Ho, Brian A. Barsky, and Ming Ouhyoung. Personalized photograph ranking and selection system. In *Proceedings of the international conference on Multimedia*, MM '10, pages 211–220, New York, NY, USA, 2010. ACM.