Improving Selfie Aesthetics with Interactive Guidance based on Empirical Models

by

Qifan Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We introduce RealSelfie, a smartphone camera application providing interactive guidance to help people take better self-portrait photos (commonly called "selfies"). The application uses empirical models to estimate aesthetic quality built from data gathered by 2,700 Amazon Mechanical Turk (AMT) aesthetic quality assessments of synthetic photographs. The synthetic photographs are generated from 3D models of realistic human models by manipulating a virtual camera and virtual lighting to precisely explore the space of three photographic principle parameters: face size, face position, and light direction. The RealSelfie application calculates the current value for each parameter using computer vision techniques and then compares those values with each model's aesthetic estimates to display directional hints overlaid on the live camera preview. As part of this system, we contribute an algorithm to estimate lighting direction using the pattern of light and shade near the nose. We conduct a study to evaluate the RealSelfie application with 20 participants in a controlled environment to eliminate background and lighting confounds. AMT ratings of the photos show that RealSelfie provides a 26% increase in aesthetics over providing no guidance.

Acknowledgements

I would like to thank all the little people who made this possible. First, thank my awsome supervisor Daniel Vogel who gave me the most help. Also, thank Edith Law, Michael Terry, and Charles Clarke for sharing their Amazon accounts with me. Without them, I would not be able to finish the study. Special thanks is given to Edith Law and Edward Lank for being my readers. Thank my friends Jingjie Zheng, Peiyuan Sun, and Yunjia Sun for providing valuable suggestions. Without the help from all these people, it is impossible to achieve this. I am really grateful to work with these people!

Dedication

This is dedicated to the ones I love, my parents Aiguo Li and Fengli Zhang, and my girlfriend Jingwen Gao!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The rapid development of mobile devices has revolutionized people's daily life especially in sharing multimedia content. As Kelsey puts it, "we are moving away from photography as a way of recording to storing the past, and instead turning photography into a social medium in its own right" [15]. With powerful phone cameras, smartphones have largely replaced point-and-shoot cameras because of portability, usability, and network connectivity. The enormous market potential has drawn phone manufacturers' attention, and it can be foreseen that in the near future the enhancement on phone cameras will be maintained.

Meanwhile, photo sharing platforms, such as Facebook, Twitter, Instagram, and Flickr are prevalent which also boosts the increase in photographs. According to statistics quoted by Flickr, an average of 6.5 million photograph are uploaded daily by its users [3] while Instagram has over 150 million active monthly users who collectively generate 1.2 billion likes per day [1]. Self portraits, commonly known as "selfies", make up a large percentage of the photos shared on these platforms. According to a poll commissioned by Samsung in the U.K. 2013, selfies account for 30 percent of pictures taken by people aged 18 to 24 [11].

Bakhshi et al. argue that faces are shown to be a powerful visual tool used in human non-verbal communication [1]. They also show that photos with faces are 38% more likely to receive likes and 32% more likely to receive comments. However, due to the lack of photography skills, many of these self-portrait photographs are unsatisfying images.

This issue has drawn researchers' attention. A common focus is the development of algorithms and techniques to measure visual aesthetic quality offline, *after* the photograph

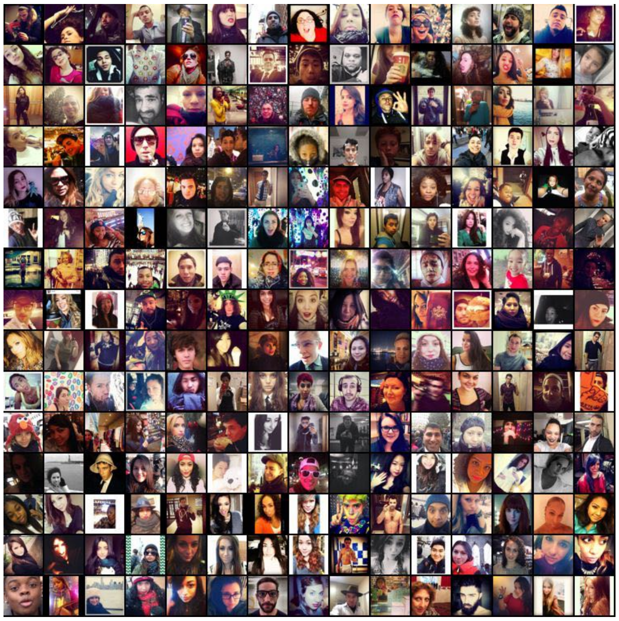


Figure 1.1: Grid of Selfies from SelfieCity 2014 [28].

has been taken. One general approach is to model aesthetic measurement as a machine learning problem with an unconstrained number of features [8, 13, 9, 39]. Specific features are extracted from a large number of images and analysed to compute a measure of aesthetics. Other researchers have proposed methods to assess the aesthetics of photographs based on the most important principles of photography mentioned in literature or followed by the professional photographers [26, 27, 34]. However, these investigations are often done using a large amount of photographs acquired from an online database with images spanning many classes, such as landscapes, animals, and portraits. This dramatically increases the variability of the images and may not provide a good general solution since different classes of photos likely have different aesthetic criteria.

More targeted approaches focusing on single classes of images exist [27, 16, 30, 34]. But these are still aesthetic measurements performed after the photo is taken. Also, when machine learning is used, suggestions for how to improve the photo are difficult to make.

Most similar to our work is a system and evaluation by Xu et al. [40]. They created a system that provided interactive feedback while a portrait photograph is being taken. However, they require advanced hardware like a three-camera array, they encode principles like the “rule-of-thirds” and best face size without validation, and they do not include the important factor of lighting. Not only do we build a system without any special hardware, but we investigate photographic principles first rather than accept them as correct, and we include the additional factor of lighting. We build empirically derived models of all three photographic principles to provide a perceptual understanding of the their relationship to aesthetics. Different than verifying whether real-time guidance can significantly improve the aesthetics of photos as Xu et al. did, we explore what features can be used while providing real-time guidance to improve the aesthetics. We believe our methodology can also be extended and generalized more easily for further study.

1.2 Contributions

This thesis investigates the effects of three features (*face size*, *face position*, and *lighting direction*) on the aesthetics of a specific subclass - single person self-portrait photographs like selfies. We use realistic male and female three-dimensional mesh models to generate synthetic images exploring the range of values for these three features and recruited 2,700 workers including duplicates on Amazon Mechanical Turk (AMT) to assess the aesthetics across those varying values. Based on the evaluations, we built an empirical model for each feature using lookup schemes and interpolation. Based on the empirical aesthetic models, we implemented a system called *RealSelfie* which provides realtime interactive feedback on these three features to assist people in taking high-aesthetic self-portraits through single tapping. RealSelfie uses computer vision techniques to detect the face size, face position, and we developed a simple but effective algorithm to estimate 3D direction of the dominant lighting in 2D photographs. Finally, we validated the usability of *RealSelfie* and show our app combined with the underlying empirical models produce self-portrait photos with 26% higher aesthetic scores than an unaided camera application.

In sum, we contribute the following:

- A systematic quantitative assessment of three compositional features of selfie aesthetics (*face position*, *face size*, and *lighting direction*) conducted on AMT. We generate three sets of synthetic self-portrait photographs using 3D modelling software and six realistic 3D human models spanning different ethnicities (three female and three male).

- Empirically-derived models of three aesthetic features (*face position*, *face size*, and *lighting direction*) that estimate the aesthetic score and “direction for improvement” of self-portrait selfie photographs.

- A simple but effective computer vision algorithm to estimate the dominant lighting direction in a single person self-portrait photo.

- A smartphone camera application called RealSelfie to assist novice photographers to take self-portraits of high aesthetic quality by providing real-time guidance using our lighting direction detection method and empirically derived aesthetic models.

- The results of a controlled lab experiment and AMT experiment validating the usability of RealSelfie and its capability to increase the aesthetic quality of self-portrait photographs by 26%.

1.3 Organization

The thesis is organized as follows:

- Chapter 2 describes previous work: photography principles, computing aesthetic measures in photographs, and applications and techniques to improve smartphone photography.

- Chapter 3 describes the systematic quantitative assessment of selfie compositional aesthetics: the generation of synthetic photographs and the AMT experiment design and results.

- Chapter 4 describes the empirical models for each of the three compositional features.

- Chapter 5 describes the implementation of RealSelfie: the methodology to calculate the *face size*, *face position*, and *lighting direction* (using the algorithm we developed); and the user interface of the application.

- Chapter 6 describes the controlled lab experiment to evaluated RealSelfie usability and the AMT experiment to show how our models and RealSelfie increase aesthetic quality of selfies.

- Chapter 7 summarizes our results and contributions, and considers future work.

Chapter 2

Related Work

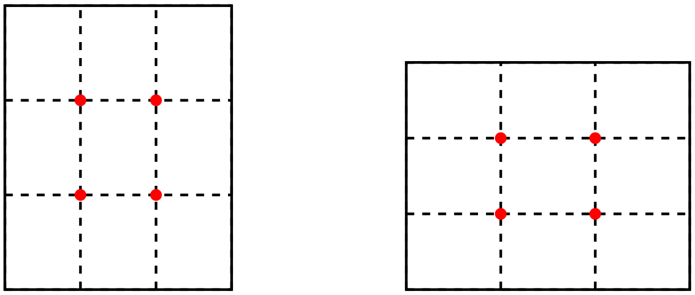
We review common photography principles, computational approaches to analyse and measure image aesthetics, and previous efforts related to improving aesthetics while a photograph is taken.

2.1 Photography Principles

Child [6] emphasizes that composition is important to attract and keep the viewer's attention and it complements the communication between the viewer and the photograph. One of the most common rules formulated over the centuries to help artists create harmonious images is *rule-of-thirds*. These grid lines and their intersecting points for "power-points" used to place significant elements within the image (illustrated in Figure 2.1). The significance of light is also discussed by Child – "...without light there is no photography... light creates texture, shape, and perspective..." [6]

Hurter also emphasizes the importance of both composition and *light* [12]. He further explains the application of rule-of-thirds for different portraits: head-and-shoulders, three-quarter, full-length, and vertical portraits. Light is the dominant factor to represent a three-dimensional reality in a two-dimensional space. "Just as a sculptor models clay to create the illusion of depth, so light models the shape of the face to give it depth and form." – Hurther [12].

The photographic principles including composition and lighting are further identified from a computational perspective using machine learning technique. Based upon these principles, Datta et al. [8] and Dhar et al. [9] built classifiers and applied linear regression



(a) Vertical

(b) Horizontal

Figure 2.1: The rule-of-thirds principle says that the subject of interest should lie at "power-points" — the intersection (red dots) of gridlines that divide an image into a three by three grid.

to infer the aesthetic quality of pictures across all classes. For the class of photos with faces, the importance of rule-of-thirds and *face position* on the aesthetic quality is also recognized by modelling the aesthetic quality based on the principles [41, 18].

2.2 Computing Aesthetic Measures in Photographs

2.2.1 General Image Aesthetics

Most of the present work investigated the influence of a large number of features on the visual aesthetics of photographs from all classes and modelled it as a machine learning problem. Datta et al. [8] extracted in total 56 features based on 3 principles: rules of thumb in photography, common intuition, and observed trends in photography. They built automated classifiers using a Support Vector Machine (SVM) and classification trees, then applied linear regression on polynomial terms of the features to infer the aesthetic quality.

To investigate the features more systematically, Dhar et al. [9] analysed 26 features from 3 dimensions: image layout or configuration, objects features or scene types, and sky-illumination attributes. Then they explored the effect of each feature on the aesthetic quality of general images and developed a simple method which automatically selected high aesthetic quality images by building high-level image feature predictors and training classifiers.

Schifanella et al. [35] used a similar approach to design a computational aesthetic framework to surface beautiful, but unpopular images. They also studied three regressed compositional features: colour features, spatial arrangement features, and texture features. They trained category-specific models using Partial Least Squares Regression (PLSR) and combined all of them to predict the aesthetic quality of images.

2.2.2 Aesthetics of Portraiture

In these works, researchers show the complexity of evaluating the aesthetic quality of photographs of all classes due to the high dimensionality of features. To further simplify the problem of investigating the influences of various features, the space of photography can be narrowed to human portraiture. Work on the aesthetic assessment of portraiture has a similar approach.

Males et al. [27] presented a method similar to Datta et al. [8] which is built on eight features that are most important to professional photographers including sharpness, low depth of field, composition, contrast, lightness, clipping, blown-out highlights, hue count, and face size. In addition to SVM, they also used Real Adaboost to assess the aesthetic quality of head shots.

Redi et al. [34] also probed this problem from the perspective of photography. They designed a framework which automatically evaluates the beauty of digital portraits with a classifier built on 5 main photographic dimensions acquired from portrait photograph literature including compositional rules, scene semantics, portrait-specific features, basic quality metrics, and fuzzy properties. Differently, Mazza et al. [30] examined the high-level content features influencing context perception of portraitures such as the dress and the gender.

Khan and Vogel [16] proposed a strategy to quantify the aesthetic quality of photographic portraits of individuals by using only a small set of classification features including spatial composition, the composition of highlights and shadows. Compared to the previous work using a large set of features, they achieved better predictors of human aesthetic judgements using a much smaller set.

However, all these works use photograph datasets acquired from an online database. This means extracting the features was not well controlled which may reduce accuracy when predicting aesthetics quality. Also, some of the rules used in these work are proved even to be wrong [40, 3], or are investigated for only one specific dimension such as spatial composition. Obrador et al. [31] discussed the important composition rules in detail and pointed out that rule-of-thirds was the best one which was also used in machine learning approach [8]. Li and Chen [19] further investigated the influence of face proportion same as Luo et al. did [25].

Meanwhile, Mazza et al. [30] showed that the background is not an influential factor for perceived context of portrait pictures. Redi et al. [34] concluded that the aesthetic quality of a portrait is linked to its artistic value, and independent from its age, gender, and race. They performed a large-scale regression analysis using LASSO[37]. They trained the regression parameter vector on demographics and computed the Spearman correlation between the predicted aesthetic score and the original score, and they found that demographics is largely uncorrelated with photographic beauty.

Manovich's Selfie City project et al. [28] is a pseudo quantitative study of selfies harvested from social media sites. The selfies are investigated using a mix of theoretic, artistic, and quantitative methods. They revealed the demographics (age and gender) of people taking selfies, their poses and expressions (smile), and discuss selfies in the history of photography, the functions of images in social media, and methods and dataset. They do not examine aesthetics.

2.3 Techniques to Improve Smartphone Photography

2.3.1 Guiding Systems

While the features affecting visual aesthetic quality have been studied comprehensively, researchers have also developed interfaces based on composition rules to guide people to take better photographs.

Ma et al. [26] proposed a photography suggestion approach to assist people to take high quality images by drawing an enclosure where the human should be located in the scene. The suggested enclosures are satisfactory based on the feedback provided by both novice and professional photographers. They defined parameters based on rule-of-thirds and visual balance, and built a function with these parameters to predict the score of enclosure. The system found the candidate enclosures under the condition that the salient objects are not overlaid, and the enclosure with highest predicted score was selected. Finally, the enclosure was optimized depending on the statistical results (on size and aspect ratio) of the enclosures from professional portrait photos containing landscape scenes. However, their model was built on the professional photos which still involve other features' effect. The effect of the parameters on aesthetics cannot be guaranteed as others. Also, their system cannot provide real-time guidance but only finding the best enclosure within a landscape image.

Bhattachrya et al. [3] presented an interactive application that enables users to improve the visual aesthetics of their digital photographs using spatial re-composition. Based on the interactive selection of a foreground object, the system presents recommendations for where it can be moved to optimize the aesthetic quality of image. However, this is post-photograph guidance. Xu et al. [40] prescribed a fully-automated solution. They implement a photo-taking interface providing real-time feedback on where to position the subject-of-interest based on rule-of-thirds with a three-camera array. They focused on the effect of providing real-time guidance and proved that the aesthetic quality of photographs was significantly improved by providing guidance compared to only a static rule-of-thirds grid. They also found that subject proportions close to 1/3 resulted in higher aesthetic quality. A problem shared by these interfaces is that they can provide real-time feedback only on composition but not on another important feature, lighting effect.

2.3.2 Accessories

Meanwhile, accessories have been developed to help people take selfies with higher aesthetic quality. Attaching a smartphone to a long pole and remotely triggering the camera shutter is a way to get more people or more of a background in a selfie. This idea has resulted in commercial "selfie sticks" which have become very popular – over 100,000 were sold in December 2014 alone [36].

Special lighting smartphone hardware are also available. LuMee [22] is a LED light phone case to take brighten selfies. A similar but more powerful phone case, Ty-Lite [23], provides cool, warm, and brilliant light settings for various lighting effects and preferences of different users.

Some specific cameras have also been commercialized just for people to take better selfies such as Casio digital cameras (e.g. EX-TR60, EX-TR70, etc.) [7]. The aesthetic quality of portrait photos is improved by changing the skin tone, smoothness, and tint. However, these are improvement applied after the photo is taken and they sacrifice the trueness of the portrait.

2.4 Summary

An issue with previous work analysing aesthetic quality is that researchers compare actual photographs (taken in uncontrolled conditions like Xu et al., or gathered from public datasets like most others). This means there is no control for what feature led to a rating. It could be an aesthetic difference, but it could also be facial expression, different backgrounds, or even culturally influenced features like youthfulness, colour of hair, eyes, and skin [28]. In addition, previous efforts to improve aesthetic quality have relied on specialized hardware like light attachments.

Xu et al. [40] are the only example of real-time guidance system to improve composition, but there system required a three camera array mounted on a tablet and they focused on a limited number of compositional factors without verifying the relationship between photographic principles and aesthetic quality [40].

Our work addresses each of these issues. In the next chapters we describe how we created a controlled dataset of portrait photographs, how we used this dataset to investigate the actual relationship between photographic principles and aesthetic quality, and how we turned those results into quantitative models to provide real-time composition and lighting guidance in a camera application on an unmodified smartphone.

Chapter 3

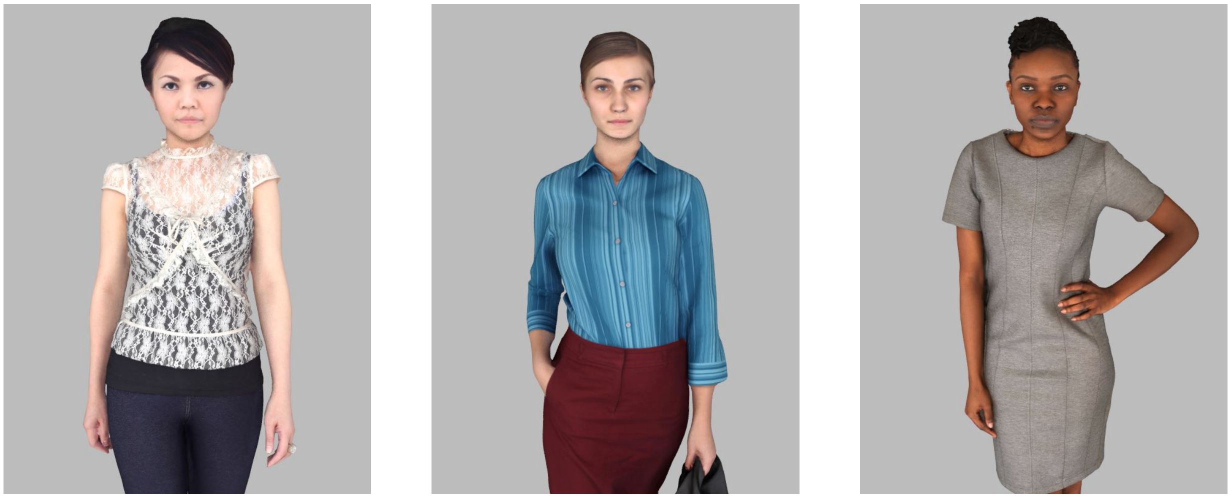
Aesthetic Ratings for Key Compositional Features

In this chapter, we describe how we created a synthetic selfie portrait dataset and an Amazon Mechanical Turk (AMT) experiment to gather aesthetic ratings of selfie portraits for key compositional features. First we generated synthetic selfie portraits using realistic 3D mesh models of men and women. By rendering the models in a 3D modelling package, we generate multiple selfies to explore the parameter space of three key features of composition: *face size*, *face position*, and *lighting direction*.

These sets of synthetic selfies are then used in the AMT experiment where they are rated for aesthetic quality. Since our selfies are highly controlled, our results from the experiment provide a measure of the effect of aesthetic quality in terms of only these compositional features. In the next chapter, we use these results to construct models to return an estimated aesthetic score given the current state of a compositional feature.

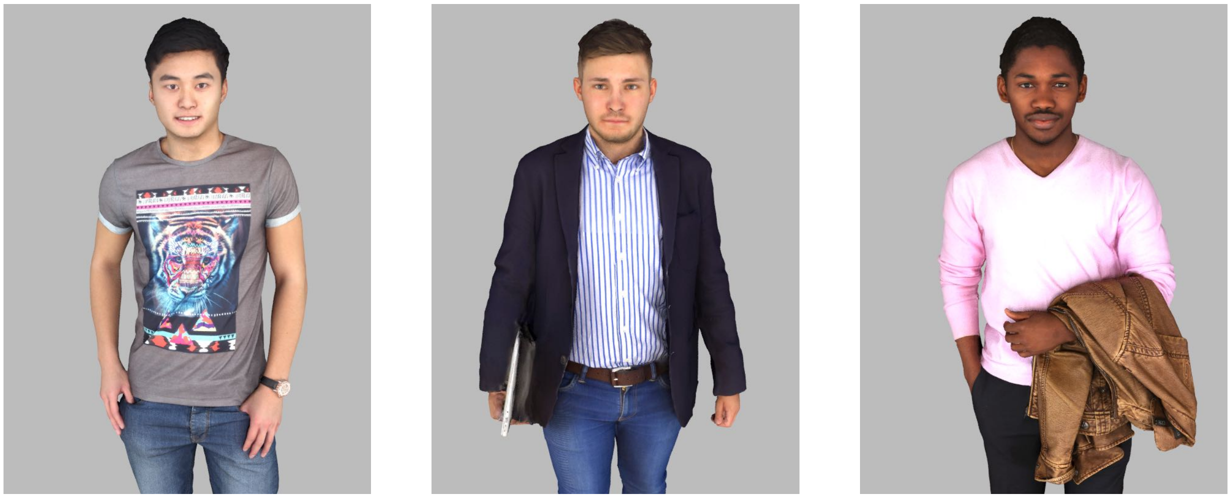
3.1 Synthetic Selfie Dataset

Each 3D human model is imported into Blender, an open-source 3D computer graphics package, where we precisely manipulate the position of the camera, the human model, and the lighting direction using a Python API and generate multiple synthetic selfie photographs. The properties of the camera including the field of view (FOV) and aspect ratio are manually set to be those of iPhone 6 camera for later application development. To imitate an outdoor scene, parallel lighting is used to simulate the sun.



(a) Asian Female (b) Caucasian Female (c) Black Female

Figure 3.1: Female Models of Different Skin-Colors



(a) Asian Male (b) Caucasian Male (c) Black Male

Figure 3.2: Male Models of Different Skin-Colors

Compared to the images collected from an online database or social media site, our photographs are highly controlled. By generating permutations of the three composition features on each of the six models, we remove confounds like background, gender, and smile when these photographs are rated for aesthetic quality.

Although we explain how all three sets of synthetic photographs before presenting the results, in fact the process was more interleaved. We generated the images for face size and ran the rating experiment for those sets of photographs first. The results of the face size ratings established which face sizes to use when generating photographs for face position. Likewise, we first ran an experiment for aesthetic ratings on the photographs for face position before selecting face positions when generating the set of lighting photographs.

3.1.1 Realistic 3D Human Models

For 3D models of humans, we use three females (Figure 3.1) and three males (Figure 3.2). These models were purchased from TurboSquid, an online 3D model marketplace (www.turbosquid.com). The models were chosen to cover Caucasian, Asian, and Black features with different skin pigments. Furthermore, the criteria to select models and poses was that they should be “average” looking, have a neutral facial expression, normal hair style, no glasses, and similar pose.

To maintain high unity, we imported the models to Blender to make minor modifications to the mesh so each faces directly to the camera without tilt. We also inserted invisible guides: transparent spheres at the eye and nose positions and a transparent plane to align the direction of the model’s face.

3.1.2 Face Size

We define *face size* as the ratio between the eye distance and the width of one cell in a rule-of-thirds grid using Equation 3.1.2 as shown in Figure 3.3. The reason to calculate the face size ratio in this way is to make the ratio relate to a conventional guideline that suggests a subject (like a face) should be approximately the size of one cell in a rule-of-thirds grid. This way a face size ratio of 1.0 matches that guideline.

To find the range of face sizes to test, we took real photographs of a closeup of the face without it becoming cropped and a face from the maximum distance possible with a selfie stick. For high accuracy, we developed a mini iOS application to detect the eye positions statically. It uses the same APIs as the ones for detecting face features in *Chapter 5*. We

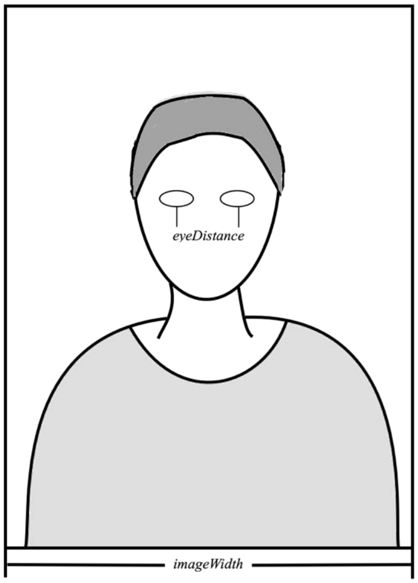


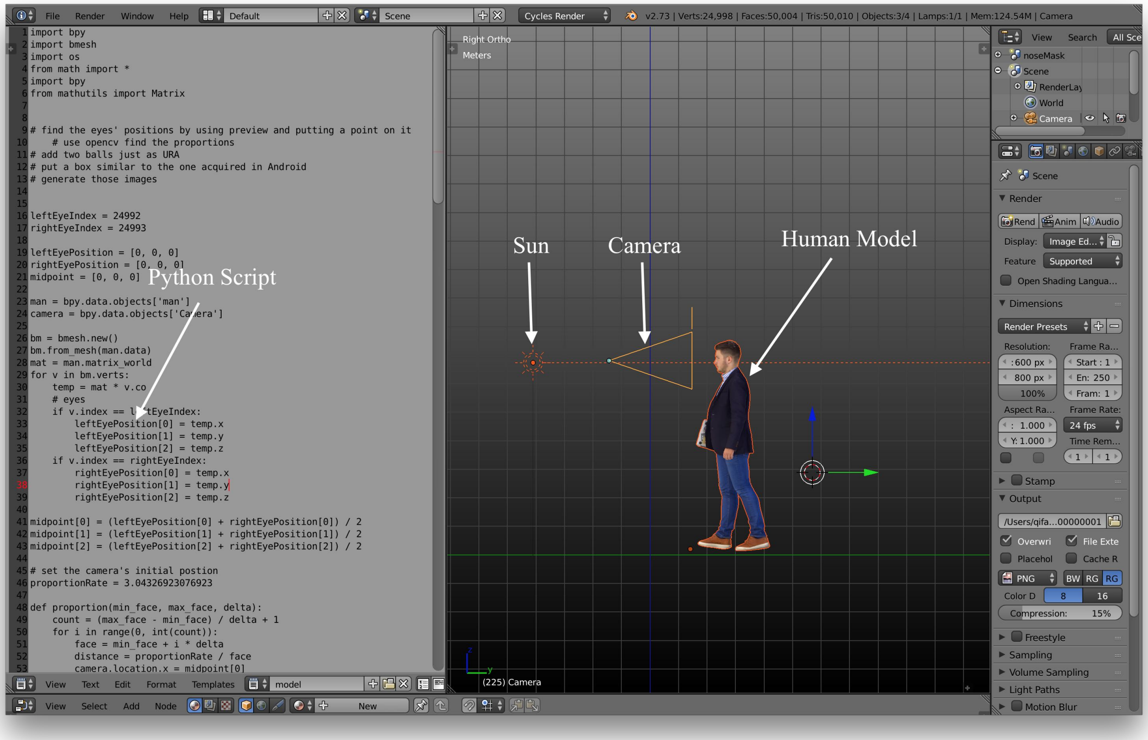
Figure 3.3: *Face size* is calculated based on *eyeDistance*: the eye distance and *imageWidth*: the width of image; both are in pixels.

imported those two images to the mini iOS application to perform static analysis of face size and calculated the face size ratio based on the same formula 3.1.2. We found that the minimal value is approximately 0.2 and the maximal value is approximately 2.0.

$$face\ size\ ratio\ = \frac{2 \times eyeDistance}{gridWidth} \tag{3.1}$$

where $gridWidth$ is defined as $imageWidth/3$.

To explore the influence of *face size* in isolation, we generate 19 images with face size ratio ranging from 0.2 to 2.0 changing by 0.1. The *face position* in all images is fixed by locating the centroid of the eyes in the centre of the image, and the lighting direction is also fixed to be straight on the face. The images with varying ratios are produced by manipulating the relative distance between the face and the camera through the Python API in Blender. For each model, this is achieved by 2 steps: *Calculate the "ratio factor"* and *Generation.*



```
import bpy
import bmesh
import os
from math import *
import bpy
from mathutils import Matrix


# find the eyes' positions by using preview and putting a point on it
    # use opencv find the proportions
# add two balls just as URA
# put a box similar to the one acquired in Android
# generate those images


leftEyeIndex = 24992
rightEyeIndex = 24993

leftEyePosition = [0, 0, 0]
rightEyePosition = [0, 0, 0]
midpoint = [0, 0, 0]

man = bpy.data.objects['man']
camera = bpy.data.objects['Camera']

bm = bmesh.new()
bm.from_mesh(man.data)
mat = man.matrix_world
for v in bm.verts:
    temp = mat * v.co
    # eyes
    if v.index == leftEyeIndex:
        leftEyePosition[0] = temp.x
        leftEyePosition[1] = temp.y
        leftEyePosition[2] = temp.z
    if v.index == rightEyeIndex:
        rightEyePosition[0] = temp.x
        rightEyePosition[1] = temp.y
        rightEyePosition[2] = temp.z

midpoint[0] = (leftEyePosition[0] + rightEyePosition[0]) / 2
midpoint[1] = (leftEyePosition[1] + rightEyePosition[1]) / 2
midpoint[2] = (leftEyePosition[2] + rightEyePosition[2]) / 2

# set the camera's initial postion
proportionRate = 3.04326923076923

def proportion(min_face, max_face, delta):
    count = (max_face - min_face) / delta + 1
    for i in range(0, int(count)):
        face = min_face + i * delta
        distance = proportionRate / face
        camera.location.x = midpoint[0]
```

Scene
Render
Rend Anim Audio
Display: Image Ed...
Feature Supported
□ Open Shading Langua...
Dimensions
Render Presets
Resolution: :600 px, 800 px, 100%
Frame Ra... Start : 1, En: 250, Fram: 1
Aspect Ra... : 1.000, Y: 1.000
Frame Rate: 24 fps
Time Rem... 1 1
□ Stamp
Output
/Users/qifa...00000001
☑ Overwri ☑ File Exte
□ Placehol □ Cache R
PNG BW RG RG
Color D 8 16
Compression: 15%
□ Freestyle
Sampling
Volume Sampling
Light Paths
□ Motion Blur

Figure 3.4: Setup in Blender: the window on the left provides the interface to run Python script to manipulate objects; the window on the right allows users to manipulate the settings of the scene; the middle one shows the scene (sideview) where the sun shines light straight on the model's face and the camera is at the same level as the eyes.

Settings for Other Features

To generate the same photographs, we first set the camera's properties in Blender to be same as the front-facing camera in iPhone 6: *Lens focal length* = 45mm, *FOV* = 54.2°, aspect ratio = 3:4. For all synthetic photographs, the background was set to be 18% grey to exclude the effect on aesthetics of the background.

We also tagged the vertices at the centres of the eyes, found the corresponding coordinates $(x,\ y,\ z)$ of both eyes, and used Equation 3.2 to calculate the centroid's position $(x, y, z)_{centroid}$ through the Python API. Based on the eye centroid's coordinates, we set the coordinates of the camera to be $(x, y, z-\delta)_{centroid}$ where δ is the distance. The camera film plane was rotated to remain parallel to the human model's face plane. The parallel light is generated by a Blender "Sun" light and rotated to be straight on the model's face shown as in Figure 3.4.

$$(x, y, z)_{centroid} = \frac{(x, y, z)_{left} + (x, y, z)_{right}}{2} \tag{3.2}$$

Calculate the Distance to Face Size Ratio

Before generating the synthetic images of *face size*, it is necessary to find the relation between d (the distance from the camera to the centroid of eyes) and the r (the face size ratio) for each model. The value of ratio factor for each model is slightly different because the feature detection algorithm in iOS gives a different result for a different model, likely based on the model's interocular distance and the shape of their head.

For each model, we used a ratio function (Function 3.3 where α is defined as the ratio factor) to represent the relation because the *face size ratio* decreases as the *distance* increases. The only thing left was to calculate α for each model.

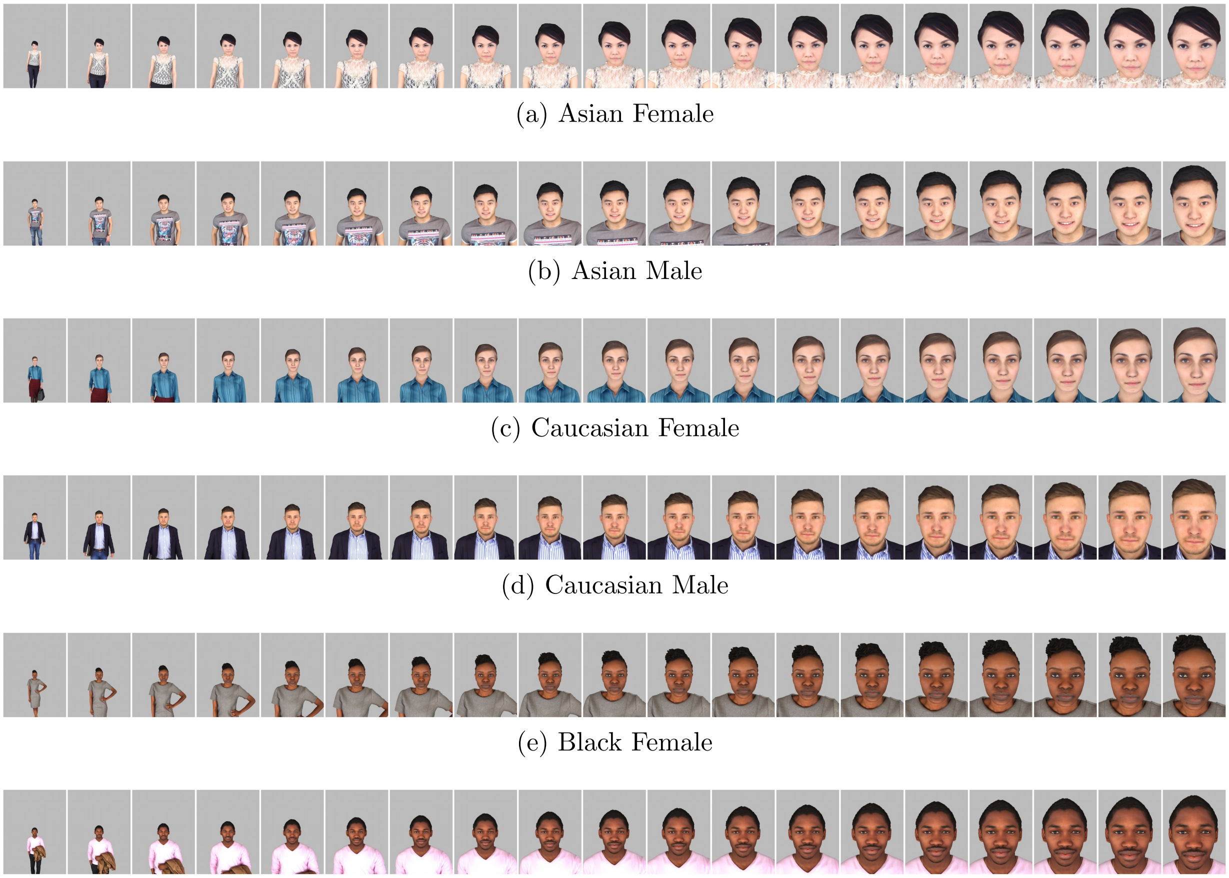
$$r = \frac{\alpha}{d} \tag{3.3}$$

For high accuracy and efficient calculation of α, we picked specific distances ($d_1 = 1.0, d_2 = 2.0, d_3 = 3.0$) and generated the corresponding image for each distance. Then, we imported the images to the previously-used mini iOS application which detected the eyes' positions and calculated the correspondinsg rs (r_1, r_2, r_3). Once the distances and ratios were passed into Equation 3.3, three ratio factors ($\alpha_1, \alpha_2, \alpha_3$) were calculated. Finally, the mean of the three ratio factors was assigned to be the α of the specific model (Equation 3.4). The same procedures were performed to calculate the mean α for each model.

$$\alpha = (\alpha_1 + \alpha_2 + \alpha_3)\ /\ 3 \tag{3.4}$$

Generate the Images

Based on the *ratio factor* α, we passed in the desired $face\ size\ ratio$ values, 0.2 (with a selfie stick) to 2.0 (close-up) in increments of 0.1, to get the corresponding camera distances. Then we positioned the camera at the corresponding positions and generated the images through Python API. The final set of 19 face size images for each model are shown in Figure 3.5.



(a) Asian Female

(b) Asian Male

(c) Caucasian Female

(d) Caucasian Male

(e) Black Female

(f) Black Male

Figure 3.5: Synthetic images of *face ssize* with r ranging from 0.2 to 2.0

3.1.3 Face Position

We define *face position* as the relative position of the centroid of the eyes within a normalized 12 × 12 grid (see Figure 3.6. Dividing the space into 12 cells follows from dividing the three grid cells in a rule-of-thirds grid by four. For this reason, *face position* values are multiples of $\frac{1}{12}$.

Settings for Other Features

In pilot tests, we found that face size affects face position aesthetics. This follows from the observation that faces of different sizes become cropped at different positions. Therefore, we generated four sets of *face position* images, each with fixed face size ratio. We chose *face size* ratios, 0.3, 0.5, 0.8, and 1.0 because they form key positions in the distribution of *face size* aesthetic scores. Recall we interleaved synthetic photo set generation with aesthetic rating experiments. The distribution of aesthetic scores is provided in Figure 3.15 below). The lighting direction is the same as *face size* images, straight on the face.

Calculate Camera Positions

Since the α (distance-to-face-size ratio) is known, we passed in the four desired ratios to Equation 3.3 for the corresponding distances. In the beginning, the camera is placed at $(x, y, z - \delta)_{centroid}$ so that the centroid is in the centre of the image (the red dot in Figure 3.6). Based on the distance, the FOV, and the aspect ratio of the camera, we calculated the width and height of the image in 3D space. Then, we segmented the image using rule-of-thirds by drawing 2 evenly separated lines both vertically and horizontally. To refine the space and keep the face from becoming cropped, we chose in total 81 (9 × 9) positions: all dots in Figure 3.6 where each rule-of-thirds cell is further subdivided into 4 sub cells. We calculated the grid width w_g and height h_g using Formula 3.5 with image width w_i and height h_i.

$$\begin{aligned} w_g &= w_i \ / \ 12 \\ h_g &= h_i \ / \ 12 \end{aligned} \tag{3.5}$$

Generate the Images

We procedurally translated the camera according to the grid's width and height. For example, to generate the image with the face at the blue spot in Figure 3.6, we need to

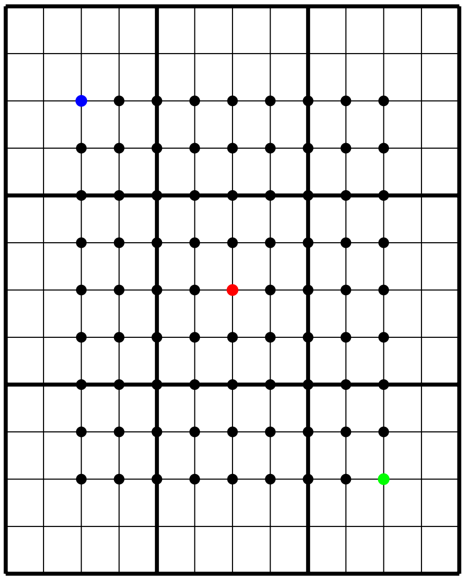


Figure 3.6: 12×12 discretized grid space to quantify face position

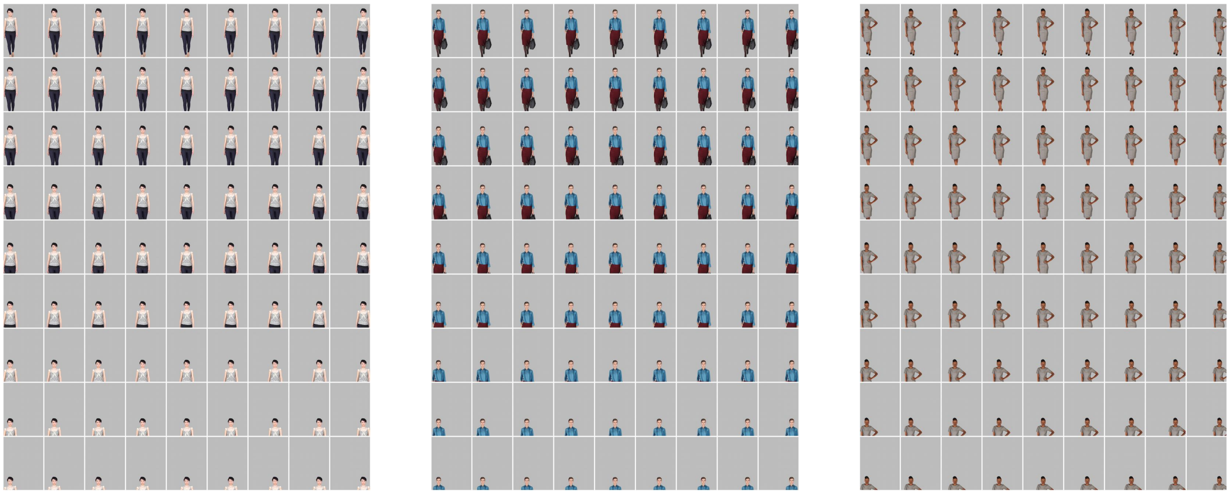
move the camera in the opposite direction to the green spot. We iterated through all 81 positions to generate all images of different face positions for each face size ratio.

To assure that the face is always oriented to the camera, we use the transparent plane on the model's face to set the model rotation so the film plane is parallel.

3.1.4 Light Direction

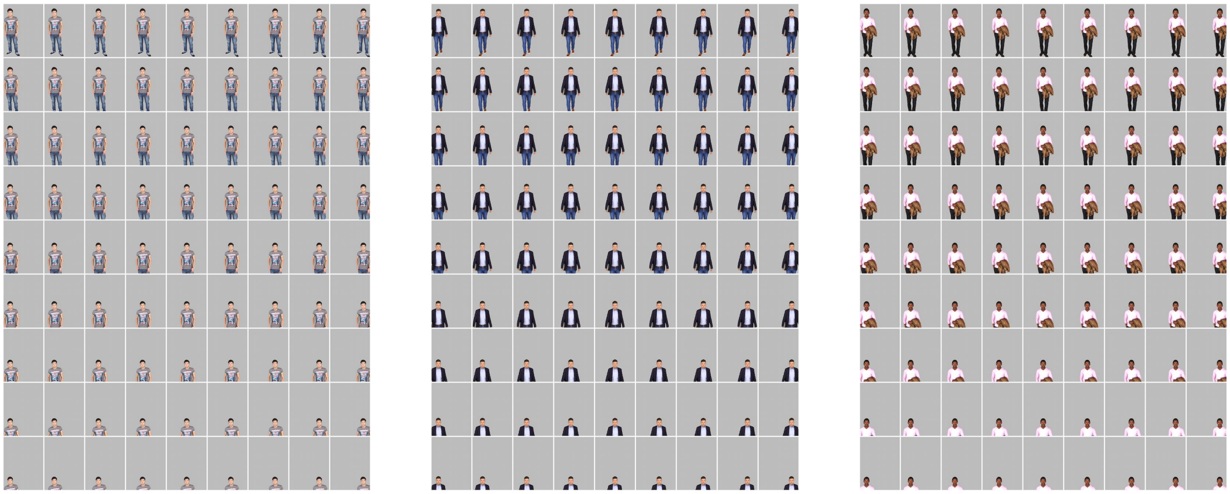
Our *lighting direction* composition feature captures the dominant direction of light on the face, such as the sun. We parametrize this direction as two angles: θ for the light elevation and ϕ for the light azimuth (see Figure 3.9). Assuming that the model's face is at the origin and facing in the direction of the positive x axis, point(r, θ, ϕ) represents the position of the light source.

Therefore, when $\theta = 0°$, the light is positioned directly above the face. When $\theta = 90°$ and $\phi = 0°$, the light is straight on the face. When $\theta > 0$ and $0 > \phi > -90°$, then the light is shining on the right side of the face and when $\theta > 0$ and $0 < \phi < 90°$ it is shining on the left side. When $\theta < 0$, $\phi > 90°$, or $\phi < -90°$, then the lighting is coming from behind the face.



(a) Asian Female (b) Caucasian Female (c) Black Female

Figure 3.7: Example Face Position Synthetic Selfie Sets for Female Models (for face size 0.3)



(a) Asian Male (b) Caucasian Male (c) Black Male

Figure 3.8: Example Face Position Synthetic Selfie Sets for Male Models (for face size 0.3)

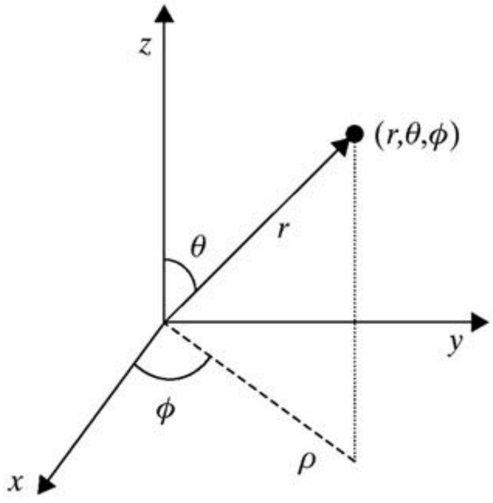


Figure 3.9: Spherical Coordinate System (diagram from [17]): θ and ϕ are used to parameterize the lighting direction

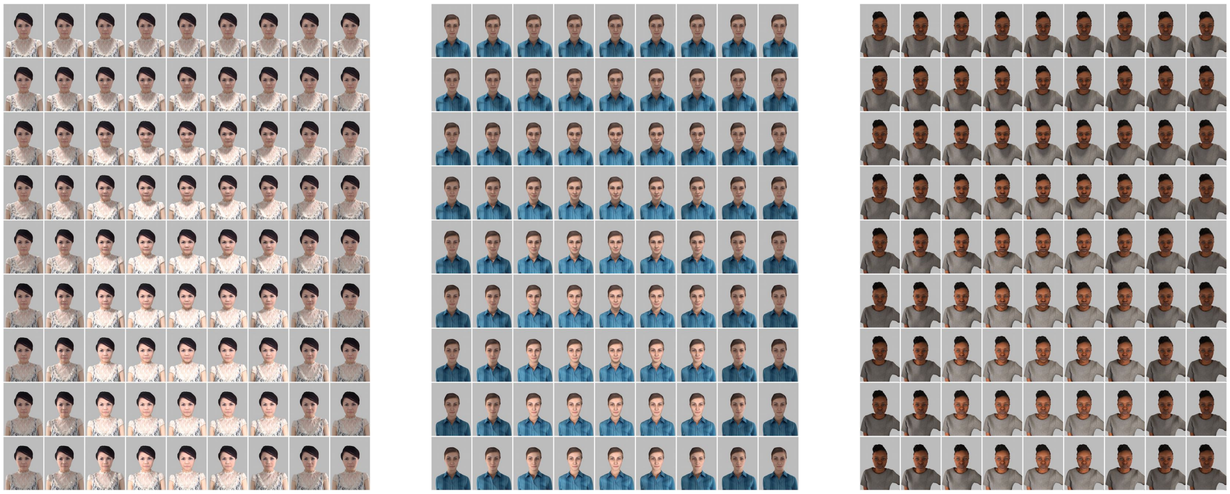
Settings for Other Features

After collecting assessment scores for *face position* (explained in the following section), we found one best position for each ace size ratio. For $r = 0.3$, the best position (x, y) is $(6/12, 2/12)$; For $r = 0.5$, the best position is $(6/12, 3/12)$; For $r = 0.8$, the best position is $(6/12, 4/12)$; For $r = 1.0$, the best position is $(6/12, 4/12)$. As before, the face is always facing the camera. For each of the best positions and corresponding sizes, we generate a set of 81 images with different *lighting directions*.

Generate the Images

We reduce the space of possible lighting directions to evaluate with the assumption that the light should primarily be landing on the face from straight on or above. For this reason, we set θ to range from directly in front (90°) to directly above (0°), and ϕ ranges from slightly behind the right (-120°) to slightly behind the left (120°).

We select a step size of 11.25°for θ, and 30°for ϕ. Hence, there are 9 values for both θ and ϕ, creating a total number of 81 combinations. After the camera is positioned to achieve the desired combination of face position and face size, our Python rotates the directional light around the face accordingly. These values are used to set the lighting direction in Blender through Python API. Example images are provided in (Figures 3.10 and 3.11).



(a) Asian Female

(b) Caucasian Female

(c) Black Female

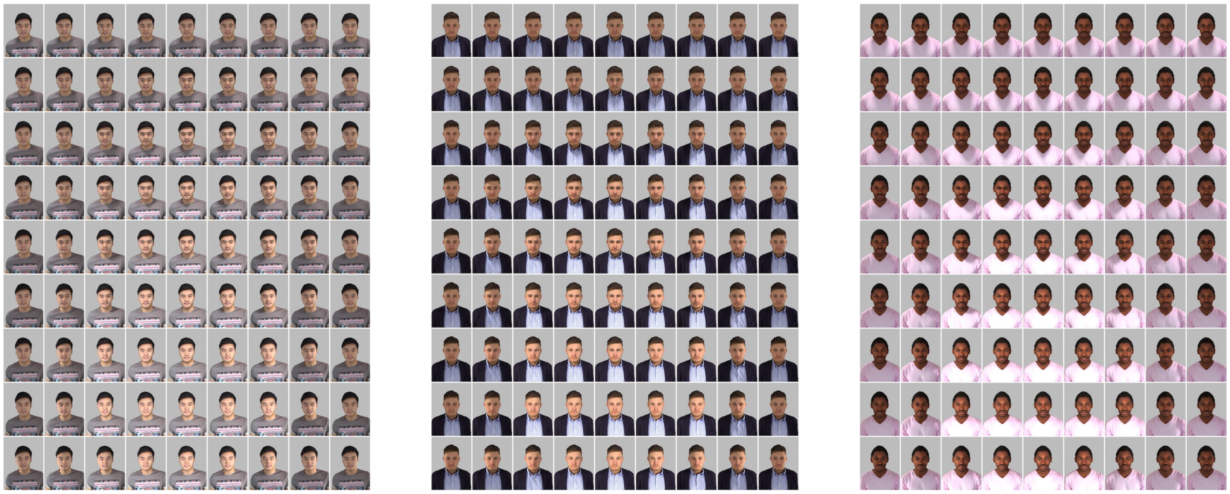
Figure 3.10: Light Direction Images of Female Models

3.2 Aesthetic Assessment Experiment

These sets of synthetic selfies are used in an crowdsourcing aesthetic assessment experiment run on Amazon Mechanical Turk (AMT). Over 2,700 workers pick the best and worst images for each model for each set of images generated for the three compositional features. These empirical results are used to create aesthetic score estimation models in the following chapter.

3.2.1 Participants

We recruited 2,700 AMT workers without any criteria for their location, experience, or age. Our objective is to get aesthetic ratings from “average people.” Workers were paid between $0.10 and $0.30 per task (called a HIT on AMT).



(a) Asian Male (b) Caucasian Male (c) Black Male

Figure 3.11: Light Direction Images of Male Models

3.2.2 Task and Implementation

The task requires a participant to view images in a set for certain model and pick N best and N worst. This design was used instead of a forced choice between two images because that would have required too many comparisons – to assess all pairs of face position photos with one model alone requires $3240 = (81 \times 80)\ /\ 2$ comparisons.

It is important that an interface for crowdsourcing tasks is clear, usable, and efficient to make it easier, and therefore more likely, for workers to complete the task correctly and honestly [33]. Our interface for workers to assess the aesthetic quality of generated photos consists of four components (see Figures 3.12, 3.13, 3.14):

- *Large image view* on the left shows the current image being viewed. Once a participant arrives on the task, they will see a randomly picked image from all images to avoid the bias caused by the initial image.
- *Thumbnails* arranged in a line or grid on right half lets participants see all images and forms a navigation method to select images to view and rate. The purple boundary shows the current image displayed on the left. Participants have two ways to quickly look through the images: one is dragging across multiple thumbnails so the large image shows the current thumbnail under the cursor; the other one is pressing the arrow keys to step through each thumbnail and large image one-by-one.

- *Three buttons* at the top-right enable participants to classify the current image into one of three categories: “Bad”, “Undecided”, “Good”. Participants can classify the current image by clicking any of the buttons. They can also classify the image by pressing shortcut keys: “1” for “Good”, “2” for “Undecided”, and “3” for “Bad”.

- *Submit* button on the bottom of the right half enables the participants to submit their evaluations once the requirements on the amount are satisfied.

To record what photos are viewed by each participant, the image which is displayed in the large image view more than 2 seconds will be automatically assigned to “Undecided.”

The entire task was developed as a website using AngularJS with all rating results logged to a MongoDB database. On AMT, we created a Human Intelligence Task(HIT) to introduce this task and provided the link to our website. Once participants accepted the HIT, they will be redirected to the website and do the task there. Once the requirements on the numbers of “Good” and “Bad”, they will get a unique “survey code” by clicking the *submit* button. They are asked to copy and paste to the HIT on AMT for differentiating each participant.

3.2.3 Design

Recall that we generated 1 set of images for face size, 4 sets of images for face position, and 4 sets of images for lighting direction, making 9 sets in total. Each task rates images in one set for one human model; with 6 human models there are $9 * 6 = 54$ task variations. For each task variation we recruited 50 workers, requiring 2,700 workers in total.

The task design for each composition feature are as follows:

- *Face Size* — In this assessment, each task has only one sub-task in which each participant is required to pick up at least 3 good and 3 bad among 19 images with face size ratio ranging from 0.2 to 2.0.

- *Face Position* — In this assessment, each task consists of 4 sub-tasks — one sub-task for one *face size ratio* (4 different face size ratios are picked from the previous assessment). In each sub-task, each participant needs to pick at least 8 good and 8 bad among 81 images — one image for one position.

- *Light Direction* — Similar to the assessment for *face position*, each task has 4 sub-tasks — one sub-task for a pair of fixed *size* and *position* which are chosen based on the results from the previous assessment. In each sub-task, each participant is also asked to pick at least 8 good and 8 bad among 81 images — one image for one lighting direction.

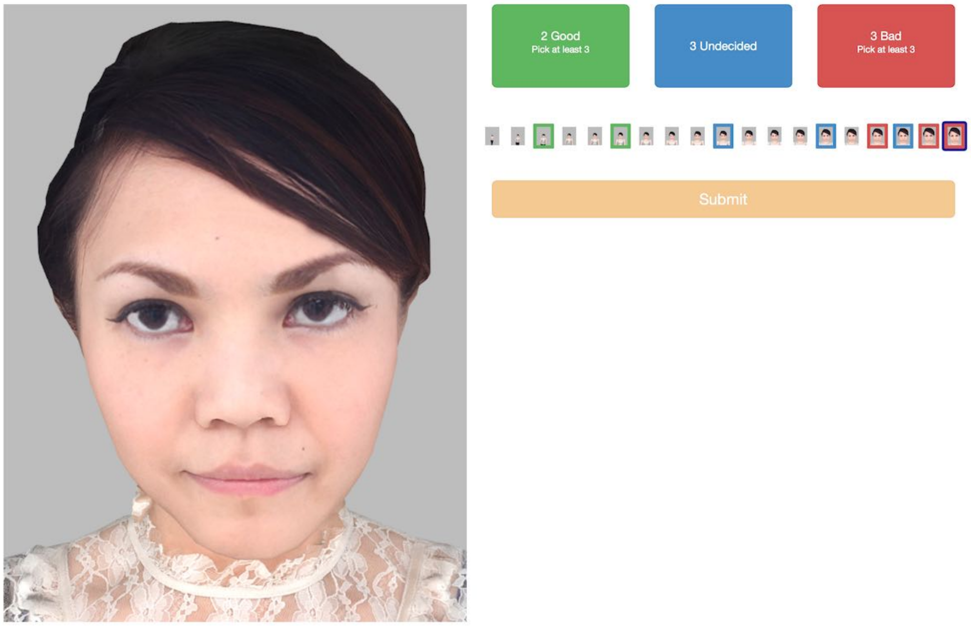


Figure 3.12: Assessment task user interface for face size.

3.2.4 Results

We disregarded those task submissions that took dramatically less time ($< 1min$ for *face size*, $< 2mins$ for both *face position* and *lighting direction*) than the average task time. This filters out likely bogus ratings from poor quality workers.

For each task, we recorded what images have been classified as "Good", "Undecided", and "Bad", as well as the number of images that were "Unviewed" (the participant never looked at them). To calculate the score of each image, we summed all ratings using the following tally: +1 for each "Good", -1 for each "Bad", and 0 otherwise. After summing this tally for each image, we conditioned the score based on the number of times it was actually assessed using:

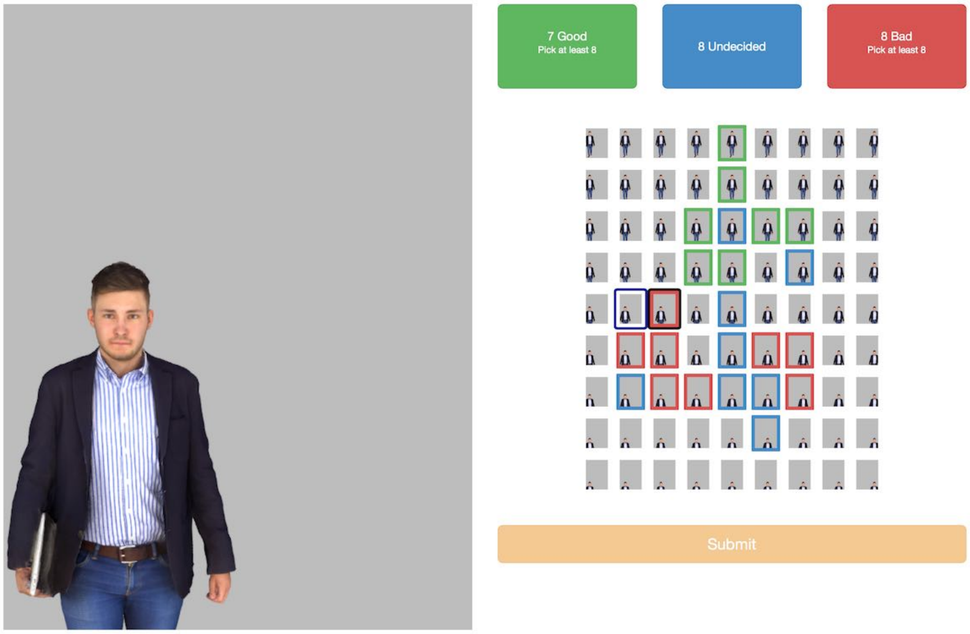


Figure 3.13: Assessment task user interface for face position.

$$true\ average = \frac{score}{number\ of\ times\ being\ viewed} \tag{3.6}$$

where *number of times viewed* is the number of ratings that were "Good", "Undecided", or "Bad" for each image. Finally, we calculated the standard error of the mean (SEM), and the percentage of each image that was viewed to examine the actual sample size and stability of the score.

Face Size

Figure 3.15 illustrates the results. The standard error of the mean (SEM) is very stable ranging from 0.04 to 0.05 indicating high consensus for the scores. In addition, all images were viewed by more than 74.27% of the workers.

The highest score is 0.33 when the face size ratio is 0.8 other near maxima at face size ratio of 0.5 (score 0.32) and face size ratio of 0.9 (score 0.32). In between these peaks the score dips to 0.22 at face size ratio of 0.6. This suggests people prefer faces to be 50% of a rule-of-thirds-grid cell or 80% to 90% of a rule-of-third grid cell. Note that faces very far

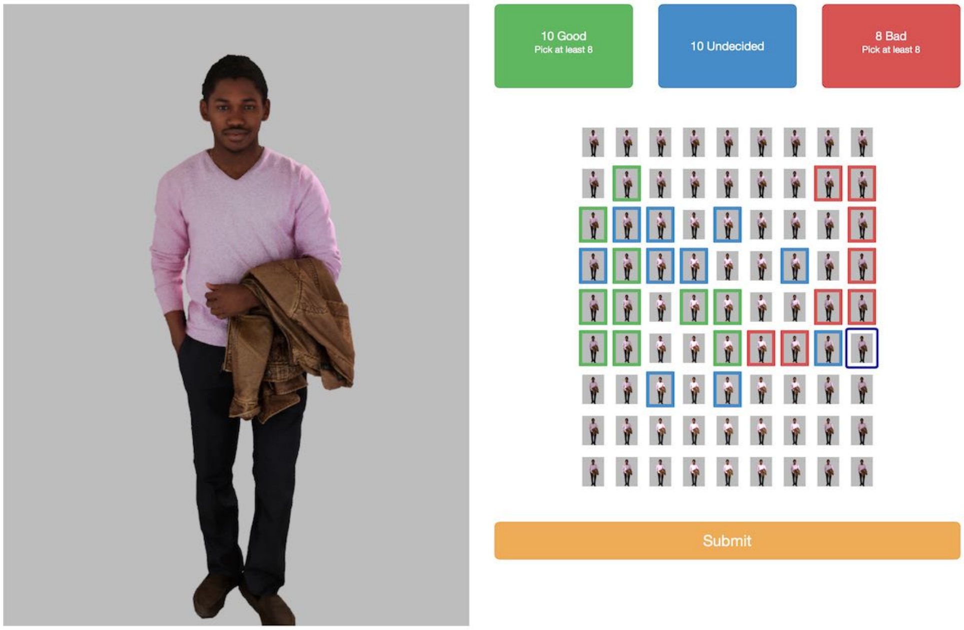


Figure 3.14: Assessment task user interface for lighting direction.

from the camera, approximately less than 30% of a rule-of-thirds grid cell, are rated lower. But most pronounced are faces very close to the camera, the score dips below 0.08 at face size ratio of 1.3 (130% of a rule-of-thirds grid cell) down to the lowest score of -0.64 at 2.0 (when the face is the size of two rule-of-thirds grid cells).

Face Position

Figure 3.16 illustrates the results as a line graph. The standard error of the mean (SEM) is very stable ranging from 0.023 to 0.040 across all tested proportions indicating high consensus for the scores. In addition, all images across all tested proportions were assessed by more than 30% of the workers.

Interpreting the trends in scores by position is difficult in the one dimensional line charts, so we also plotted score as two-dimensional heat maps (Figure 3.17). The general trend is higher scores when the face is centred (centre column with position $\frac{6}{16}$), with higher ratings for higher face positions as the face size becomes smaller (Table 3.1 for detailed results).

Higher aesthetic ratings for a centred faces breaks from the accepted rule-of-thirds

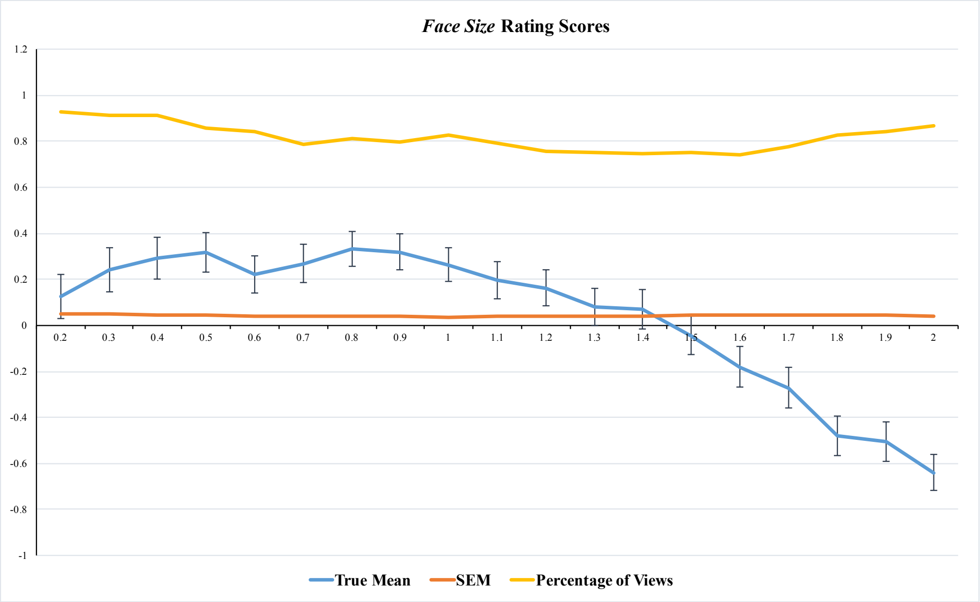


Figure 3.15: *Face size* rating scores for the ratio ranging from 0.2 to 2.0.

		Face Size Ratio			
		0.3	0.5	0.8	1.0
Best Positions (eyes centred at */12 of image)	y	2/12	3/12	4/12	4/12
	x	6/12	6/12	6/12	6/12
Score		0.82	0.88	0.90	0.88

Table 3.1: The detailed results for *Face Position*

principle which claims more aesthetically pleasing photos with primary objects of interest centred at one of the grid's "power-points." This is an interesting finding. The ratings decrease as the face deviates from the centre area and reach the worst once the face is partially cropped shown as the lighter area.

Light Direction

Same as for *face position*, Figure 3.18 illustrates the results as a line graph. The standard error of the mean (SEM) is very stable ranging from 0.028 to 0.044 across all tested proportions indicating high consensus for the scores. In addition, all images across all tested proportions were assessed by more than 45% of the workers.

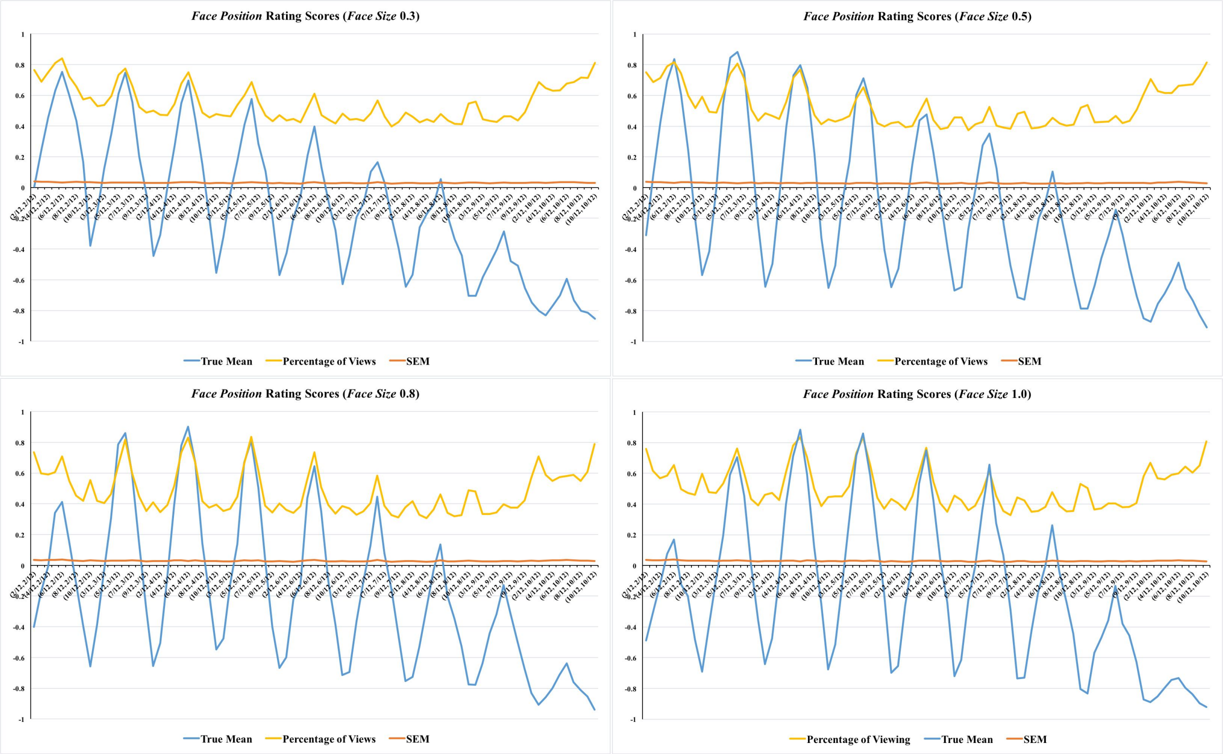


Figure 3.16: *Face position* rating scores for 4 face size ratios, 0.3, 0.5, 0.8, and 1.0. Blue line represents *true mean*; yellow line represents the percentage of being viewed among all participants; orange line represents SEM.

We drew similar heatmaps (shown as in Figure 3.20) as for 'face position. Examining the heatmaps, we can see that the region of the best lighting direction is in the bottom-middle. See Table 3.2 for detailed results. With Θ approximating 90°and Φ close to 0°, the face is lit more evenly. As the light direction deviates, more shadows are formed on the face. Since the pattern is similar, we aggregated all 4 sets of results when generating the lighting model (see Figure 3.20).

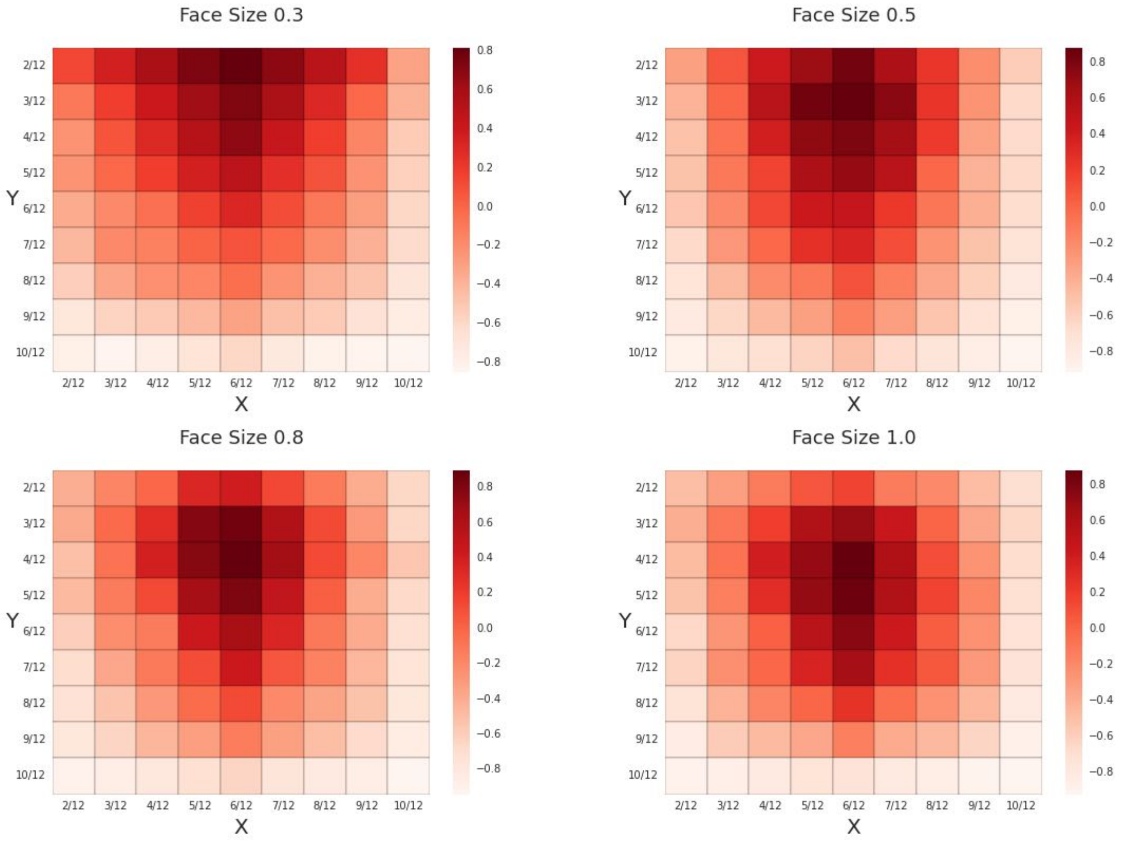


Figure 3.17: *Face size* rating scores: X represents the horizontal position and Y represents the vertical position. Both X and Y are in the fraction as the image is segmented by refined grid-lines to a 12 x 12 grids.

		Face Size Ratio			
		0.3	0.5	0.8	1.0
Best Light Direction (degrees)	$\Theta(elevation)$	78.75°	78.75°	90°	67.5°
	$\Phi(azimuth)$	30°	-30°	0°	-30°
Score		0.44	0.41	0.42	0.43

Table 3.2: The detailed results for *Lighting Direction*

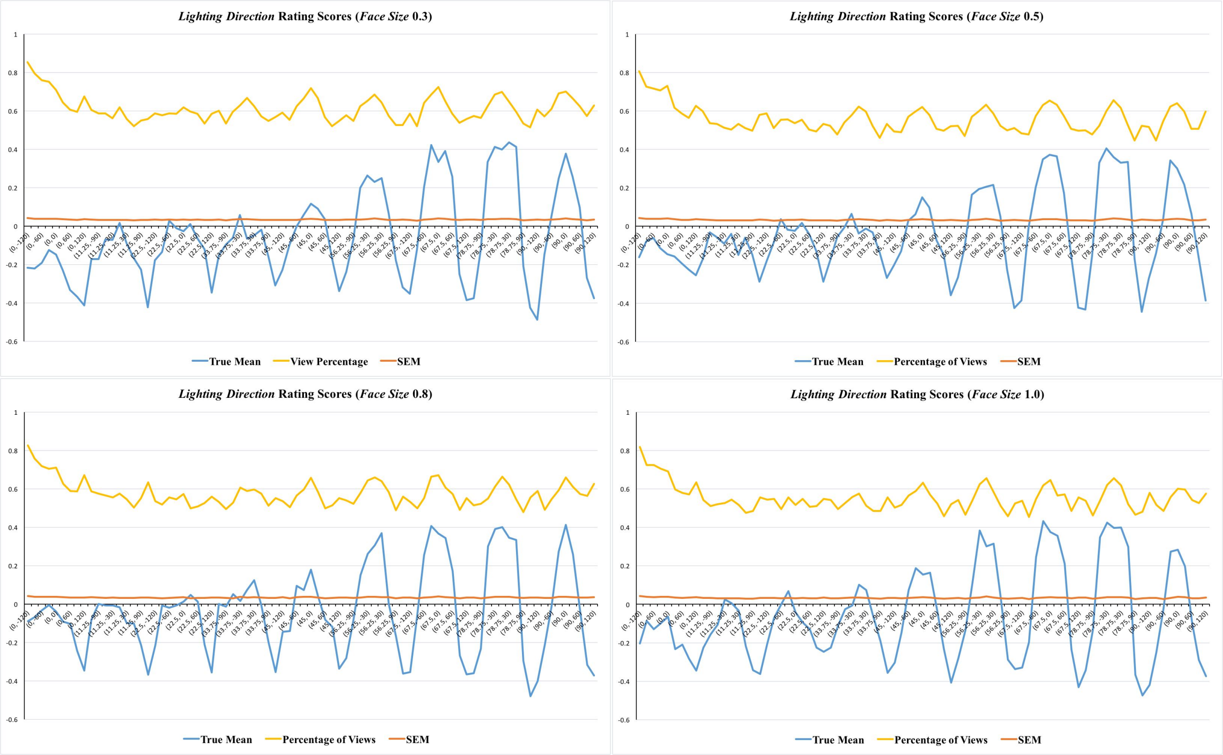


Figure 3.18: *Light direction* rating scores for 4 r's, 0.3, 0.5, 0.8, and 1.0. Blue line represents *true mean*; yellow line represents the percentage of being viewed among all participants; orange line represents SEM.

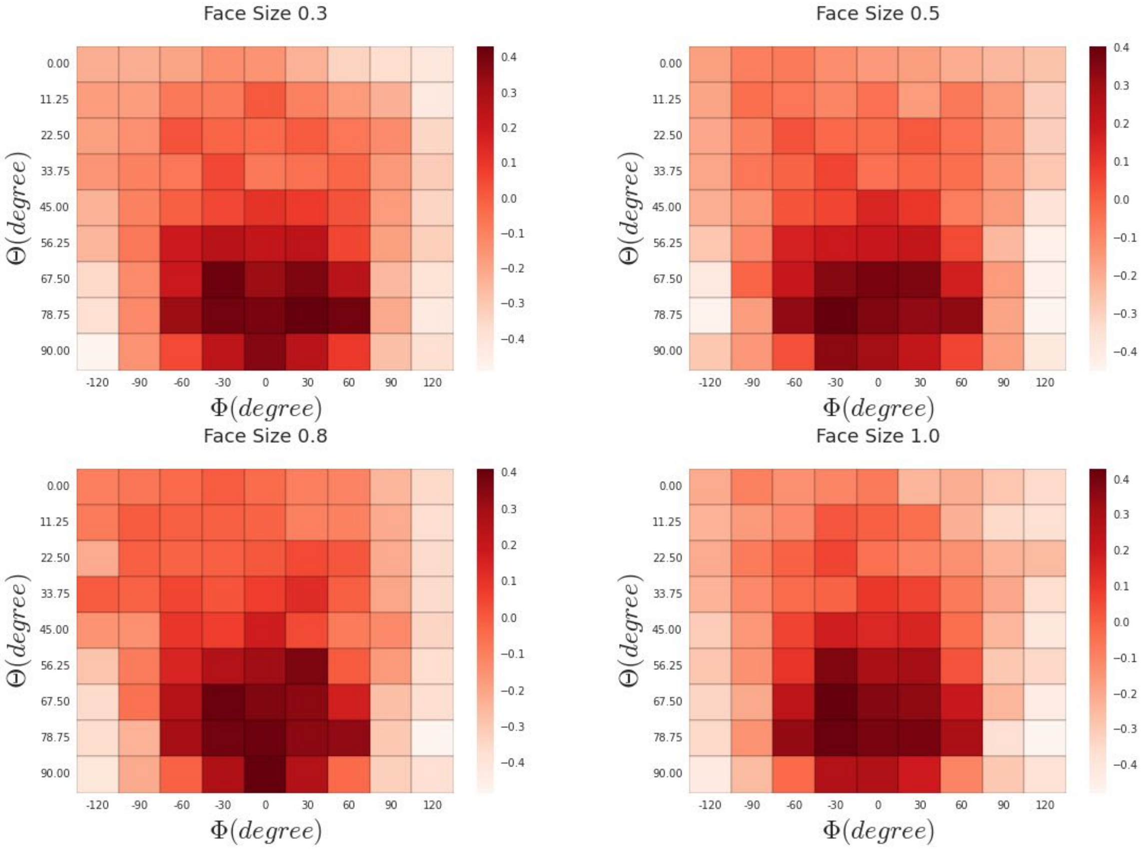


Figure 3.19: The result of assessment on *light direction* for all four r's: 0.3, 0.5, 0.8, 1.0.

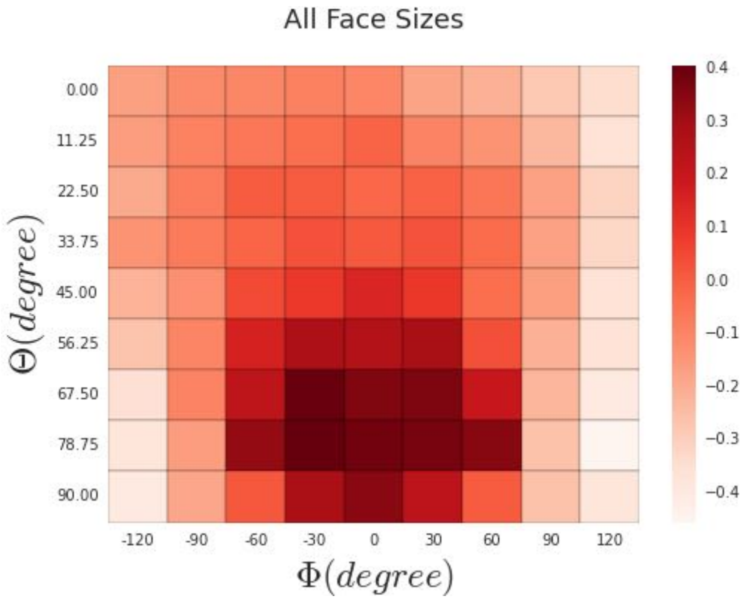


Figure 3.20: The aggregated result of assessment on *light direction* for all four face size ratios: 0.3, 0.5, 0.8, 1.0.

Chapter 4

Empirical Models of Aesthetics for Key Compositional Features

In order to translate our findings from the previous chapter into a system providing real-time guidance to improve aesthetics, we built models to estimate aesthetic ratings for the three compositional features: *face size*, *face position*, *lighting direction*. The objective is to build a system to detect the current state of these three compositional principles, then use that information with the models to find the current aesthetic rating and provide guidance to move to higher rating.

The main challenge in creating these models is how to transform the discrete scores from our rating experiments to continuous functions that each returns a score for any *face size*, *face position*, and *lighting direction*. In general, each model is a function f given a set of measured compositional features $\{\omega_0, \omega_1, ..., \omega_n\}$ that returns a score s and a vector $\boldsymbol{d}$ describing the *direction* in compositional feature space that will improve the score. Each model is therefore expressed in the form: $(s, \boldsymbol{d}) = f(\omega_0, \omega_1, ..., \omega_n)$.

The specific models we developed are:

- Face Size Models: $(s_s, \boldsymbol{d_s}) = f_s(r)$

  The results of the *face size* experiment ratings provide aesthetic scores for 19 values in one-dimension covering a range of reasonably plausible *sizes*. We use linear interpolation (Formula 4.1) to provide an aesthetic score for any reasonable value of *face size* that would be detected in a camera application. Given a detected face size ratio r, we first find the interval that r belongs to and then apply linear interpolation based on the distances between r and the two ends of the interval to find the score s_s. We use the highest of the interval scores to set the one-dimensional direction of improvement $\boldsymbol{d_s}$.

$$s = Itpl(x, a, b, s_a, s_b) = s_a + \frac{x-a}{b-a} \times (s_b - s_a) \tag{4.1}$$

- Face Position Model: $(s_p, \boldsymbol{d_p}) = f_p(x, y, r)$

  Recall that our experiment data for *face position* was sampled at four face size ratios (0.3, 0.5, 0.8, 1.0). Thus, the face position model requires the current *face size* ratio r as well as the current face position (x, y). Given the scores of the 9 × 9 = 81 two-dimensional positions across the four *face sizes*, we first perform a linear interpolation the scores of all 81 positions for the current *size*, and then apply a two-dimensional linear interpolation to find the score s_p. The direction $\boldsymbol{d_p}$ is found by iterating througth the interpolated 8 neighbouring positions and finding the one with highest score.

- Lighting Direction Model: $(s_l, \boldsymbol{d_l}) = f_l(\boldsymbol{u}, \boldsymbol{v})$

  Unlike face position and face size, we cannot directly measure three-dimensional lighting directions Θ and Φ from a two-dimensional image. Instead, we developed a computer-vision based lighting analysis algorithm to compute two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ representing x- and y-direction and magnitude for the pattern of shading around the nose (see Section *Model for Lighting Direction* for a full description). These two vectors are transformed in the model function f_l into the best estimate for Θ and Φ by finding the nearest neighbour to a set of canonical vectors $\boldsymbol{u^*}$ and $\boldsymbol{v^*}$ computed using the 3D human models with known Θ and Φ. With Θ and Φ, we can find the corresponding score in score matrix. The direction $\boldsymbol{d_l}$ is found by checking the neighbouring 8 *lighting directions* and finding the one with the highest score. Since there were no systematic difference of lighting direction aesthetic rating across the four different *face size* ratios we tested, we aggregated ratings across *face size* ratios into a single matrix of 9 × 9 = 81 ratings for Θ and Φ lighting directions.

In this chapter we provide the details for how these models work.

4.1 Model for Face Size

The *face size* model has three steps: *look-up* to find the interval where the ratio is located in the table of ratings from the experiment, *interpolation* to calculate the score based on the slope between the scores in that interval, and *direction* to find the direction to move to increase the score.

4.1.1 Look-up

Before applying linear interpolation, it is necessary to find the best interval in the table of ratios $r_0, r_1, ..., r_n$ and corresponding scores $s_0, s_1, ..., s_n$ computed from our experiment results. Given a *face size* ratio r, we seek two consecutive ratios in the table, (r_i, r_{i+1}), such that $r_i \leq r < r_{i+1}$.

For example, the interval for *face size* ratio equal to $r = 0.35$ is $(0.3, 0.4)$.

4.1.2 Interpolation

With the interval defined by (r_i, r_{i+1}), the score for the *face size* ratio r can be calculated based on the slope of the corresponding scores s_i and s_{i+1} using $Itpl(r, r_i, r_{i+1}, s_i, s_{i+1})$.

Continuing the example with $r = 0.35$ and table ratio interval $(0.3, 0.4)$, substituting scores $s_{i+1} = 0.29$ and $s_i = 0.24$ into Formula 4.1 becomes:

$$\begin{aligned} s &= 0.24 + \frac{0.29 - 0.24}{0.4 - 0.3} \times (0.35 - 0.3) \qquad (4.2) \\ &= 0.265 \qquad (4.3) \end{aligned}$$

In the case when the ratio is out of the range (i.e $r < 0.2$ or $r > 2.0$), the model will return the nearest sample scores respectively.

4.1.3 Direction

The goal of these models is to provide the guidance on the corresponding feature to increase the score. Since we used linear interpolation to find the locally best *face size* ratio, we can check the slope of the line: $\frac{s_{i+1} - s_i}{r_{i+1} - r_i}$. If the slope is positive, then the ratio should be bigger indicating that the camera should be closer, so $\boldsymbol{d_s} = (+1)$; otherwise, the ratio should be smaller to get a higher score indicating that the camera should be further so $\boldsymbol{d_s} = (-1)$.

4.2 Model for Face Position

The *face position* model transforms a *face position* $(x,\ y)$ into a score and direction using an additional parameter of the detected *face size* ratio r. This occurs in four steps. First, the ratio r is used to construct an interpolated position score matrix between two of the four matrices of position scores collected at four different *face size* ratios in the experiment. Then the cell location for the $(x,\ y)$ position is located in the interpolated position score matrix. Using that cell location, a bilinear interpolation using the four corners of the cell is applied to get the score s_l for exact position $(x,\ y)$. To find the direction that would increase the score from $(x,\ y)$, the scores of 8 neighbour positions are interpolated using the same methodology, and the one with the highest score is used to create the two-dimensional direction vector $\boldsymbol{d_p}$.

4.2.1 Interpolated Position Score Matrix

Recall that our experiment only assessed position scores a four *face size* ratios: $r_1 = 0.3$, $r_2 = 0.5$, $r_3 = 0.8$, and $r_4 = 1.0$. As the results in previous chapter shows, the best position is moving downwards as the *face size* ratio increases. Therefore, these four position score matrices divide the space into three sections, such that r would satisfy one of these cases: $r \leq r_2$; $r_2 < r \leq r_3$; or $r > r_3$. Given the two position score matrices defining the interval (one of three pairs matrices, defined with ratios (r_1, r_2), (r_2, r_3), or (r_3, r_4)), interpolate between all 81 corresponding pairs of cell values in the two matrices to compute the new matrix. Let s^{ij} be the score in cell i, j in each matrix, use Formula 4.1 to calculate the score s^{ij}_* as follows (using the matrix pair r_2, r_3 as an example):

$$s^{ij}_* = s^{ij}_2 + \frac{r - r_2}{r_3 - r_2} \times (s^{ij}_3 - s^{ij}_2) \tag{4.4}$$

4.2.2 Grid Position

After calculating the interpolated position score matrix, the next task is locate the cell of the matrix where (x, y) would fall given the pixel position in the image. As shown previously in Figure 3.6, the image is divided into 12 x 12 grids of which the size $(width_g, height_g)$ is calculated using Formula 3.5. Given the face position (x, y) in pixels, the grid position (x_g, y_g) can be acquired using the below Formula 4.5. Notice that we subtract 2 from both x and y because these are the valid positions without the face being cropped.

$$
\begin{aligned}
x_g &= \frac{x}{width_g} \\
y_g &= \frac{y}{height_g}
\end{aligned}
\tag{4.5}
$$

4.2.3 Score Interpolation

With the face grid position (x, y) and all position scores known, we can calculate the score $s_{x,y}$ for the current position using bilinear interpolation based on the scores of the surrounding positions (the right figure in Figure 4.1). The following is the detailed procedure to interpolate the score of position (x, y) where $x = 3.6$ and $y = 5.3$:

1. Find the surrounding 4 face positions $\{(x_1, y_1), (x_2, y_1), (x_1, y_2), (x_2, y_2)\}$ by taking the ceiling and the floor of both x and y. In this example, x_1 equals 3; y_1 equals 5; x_2 equals 4; y_2 equals 6. Therefore, the surrounding 4 face positions are $\{$(3, 5),(4, 5),(3, 6),(4, 6)$\}$. Then, get the scores based on the interpolated score data from the previous step.

2. Interpolate the scores s_{x,y_1} and s_{x,y_2} of positions (x, y_1) and (x, y_2) based on Formula 4.1. Here, the scores of positions $(3.6, 5)$ and $(3.6, 6)$ are interpolated.

$$s_{x,y_1} = \frac{x - x_1}{x_2 - x_1} \times (s_{x_2,y_1} - s_{x_1,y_1}) + s_{x_1,y_1} \tag{4.6}$$

$$s_{x,y_2} = \frac{x - x_1}{x_2 - x_1} \times (s_{x_2,y_2} - s_{x_1,y_2}) + s_{x_1,y_2} \tag{4.7}$$

3. After getting the scores of (x, y_1) and (x, y_2), the same methodology as the previous step will be applied to interpolate the score of (x, y) (i.e. $(3.6, 5.3)$) using the Formula 4.1:

$$s_{x,y} = \frac{y - y_1}{y_2 - y_1} \times (s_{x,y_2} - s_{x,y_1}) + s_{x,y_1} \tag{4.8}$$

4.2.4 Direction

To provide the guidance on *face position*, it is necessary to find the neighbours' scores at interpolated positions. Then, the position with the highest score indicates the direction to move the person's head to get a higher score on *face position*. It has the following process:

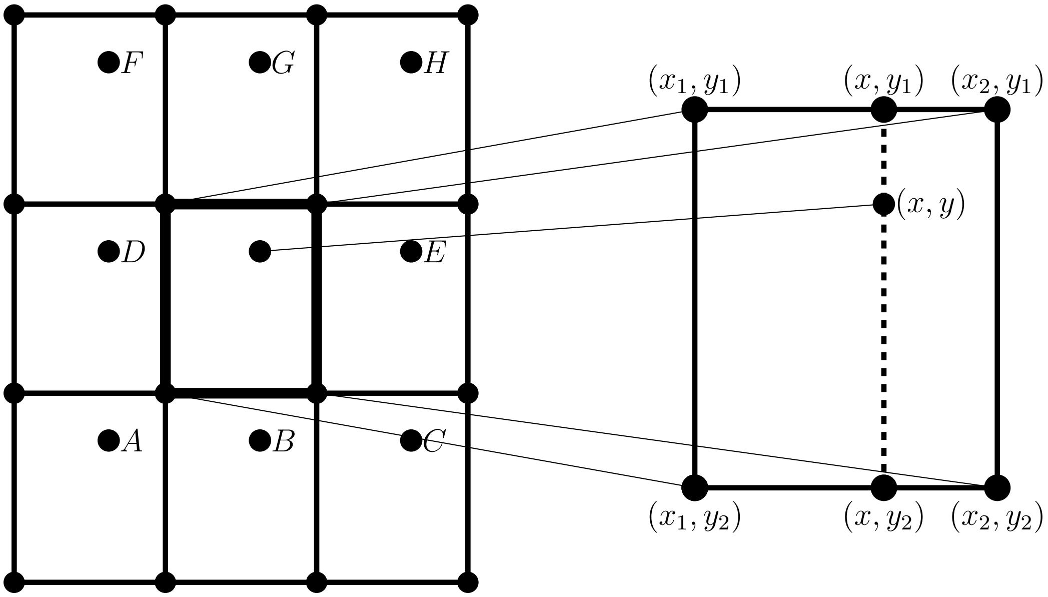


Figure 4.1: Face Position Interpolation

1. Find the 8 neighbours (A - H) by adding the grid width to x and the grid height to y shown as in Figure 4.1.

2. After locating the neighbours, we applied the same methodology as in “Score Interpolation” to interpolate the score for each neighbour.

3. Then, according to the position with the highest score, $\boldsymbol{d_p}$ is returned. For example, if it is D, then $\boldsymbol{d_p}$ equals (-1, 0); if it is E, then $\boldsymbol{d_p}$ equals (+1, 0); if it is G, then $\boldsymbol{d_p}$ equals (0, -1); if it is B, then $\boldsymbol{d_p}$ equals (0, +1); if it is A, then $\boldsymbol{d_p}$ equals (-1, +1); ...

4.3 Model for Lighting Direction

The *lighting direction* model manipulates the *lighting direction* ($\boldsymbol{u}$, $\boldsymbol{v}$) to a score and the direction of improvement. This also happens in 4 steps. First, the computer-vision based lighting analysis algorithm is run on all synthetic images which produces a pair of ($\boldsymbol{u}$, $\boldsymbol{v}$) for each image (1944(*images*) = 4(*face size* ratios) x 6(models) x 81(*lighting directions*)). For each *lighting direction* (Θ, Φ), the corresponding 24 pairs of ($\boldsymbol{u}$, $\boldsymbol{v}$) are aggregated to a single pair to represent the *lighting direction*. Therefore, 81 pairs of ($\boldsymbol{u}$, $\boldsymbol{v}$) are generated for all 81 *lighting directions*. Then, for each *lighting direction*, we compute the mean score of all 24 images. In the end, 81 mean scores are calculated, each for one *lighting direction*. The score of ($\boldsymbol{u}$, $\boldsymbol{v}$) is generated by calculating the squared Euclidean distance between ($\boldsymbol{u}$, $\boldsymbol{v}$) and all 81 aggregated pairs and finding the nearest one. To find the derection of improvement, 8 neibours of the matched *lighting direction* are checked, and the one with the highest score provides the direction.

4.3.1 Score Calculation

Given a *light direction* (Θ, Φ), the lighting analysis algorithm will generate a pair of ($\boldsymbol{u}$, $\boldsymbol{v}$). With ($\boldsymbol{u}$, $\boldsymbol{v}$) generated, the model iterates through all 81 aggregated pairs and calculates the squared Euclidean distance between ($\boldsymbol{u}$, $\boldsymbol{v}$) and each aggregated pair. The one with the minimal distance is found by applying Formula 4.9. After getting this match, the mean score for the corresponding *lighting direction* is assigned to be the score of (Θ, Φ).

$$\underset{x}{argmin}\ f(x) := \{x \mid \forall y : f(y) \geq f(x)\} \tag{4.9}$$

Assuming that the vector of the *lighting direction* in one image is (-3, 0), (0, 2) generated by the analysis algorithm, after calculating the squared Euclidean distance between all 81 aggregated pairs and the generated pair, one aggregated pair (corresponding to the *lighting direction* (Θ = 22.5, Φ = 30)) with the minimal distance is found. Then, the score of this *lighting direction* (Θ = 22.5, Φ = 30) is the score of the current *lighting direction*.

4.3.2 Direction

As the pair with minimal distance is found, its 8 neighbors are investigated. The one with highest score provides the direction to move to improve the score. The same methodology as for *face position* is applied here to represent the direction. For example, if it is D, then $\boldsymbol{d_l}$ equals (-1, 0); if it is E, then $\boldsymbol{d_l}$ equals (+1, 0); if it is G, then $\boldsymbol{d_l}$ equals (0, -1); if it is B, then $\boldsymbol{d_l}$ equals (0, +1); if it is A, then $\boldsymbol{d_l}$ equals (-1, +1); ...

Continuing the example with $\boldsymbol{u} = (-3, 0)$ and $\boldsymbol{v} = (0, 2)$ (the middle dot in Figure 4.2), the model finds that H has the highest score among the 8 neighbors. Then, the model returs $\boldsymbol{d_l}$ which equals (-1, +1).

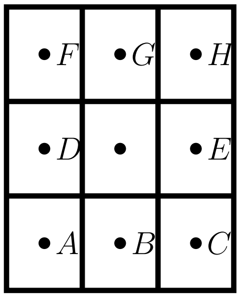


Figure 4.2: Light Interpolation

4.4 Summary

In this chapter, we explained how our models calculate the score, and the direction to move to improve the score, for each compositional feature. Linear interpolation is applied given the detected *face size ratio*; bilinear interpolation is used to calculate the score of the detected *face position*; different than previous two, the score of the detected *lighting direction* is calculated by finding the closest sampled *lighting direction* and assigning its score as the current score. To increase the score on each feature, different methods are used: slope for *face size*; checking surrounding ones for both *face position* and *lighting direction.* These methods are implemented in the application RealSelfie to provide real-time guidance on each feature.

Chapter 5

Camera Application to Improve Selfie Aesthetics

We developed a smartphone application (or “app”) called “RealSelfie” to guide people to take aesthetically pleasing self-portrait selfie photos using the empirical models described in the previous chapter. The RealSelfie app detects the current state of compositional features in a portrait photo using computer vision techniques. Based on the detected position of a human face, eyes, and nose, it directly determines the *face size* ratio and *face position.* For *lighting direction*, we developed a simple lighting direction analysis algorithm based on the brightness pattern around the nose. By using the current state of compositional features and our empirical models, we can guide the user to move their smartphone to improve the aesthetics of their selfie photo.

The most challenging part of building the RealSelfie app was detecting the lighting direction and making the app work fast enough for interactive guidance. The detection of face and eyes can be achieved quickly using a native API provided by iOS. However, detecting the nose with Haar Cascades [38, 21] is much slower than the others. To improve the efficiency, we downsample the region of interest by only focusing on the nose area and reducing the resolution. Also, we applied the singleton design pattern for importing the all three empirical models to avoid unnecessary reloading the models. In the end, our system running on an iPhone 6 running iOS 9.3 performs in near real time.

Compared to Xu et al. [40], our system does not require a three-smartphone-array or larger tablet for providing guidance – RealSelfie works on an unmodified smartphone. Xu et al. encode pre-existing compositional rules for guidance while we use our empirically derived models which represent what compositional qualities people like. We also include *lighting direction*, whereas Xu et al. only consider face position and size.

5.1 Computer Vision for Feature Detection

To provide guidance for each compositional feature, we first need to calculate each feature value from the preview image. Face size and position are both calculated from the distance between the eyes and their position respectively. Our *lighting direction* algorithm requires the position of the nose. Therefore, the first task is to detect a human face, eyes, and nose.

5.1.1 Head, Eyes, and Nose Detection

The preview image is captured with the `AVCaptureVideoDataOutputSampleBufferDelegate` protocol. As each frame is written to the buffer, it is first converted to a core image. Then, to speed up the analysis process, we downsample the image to half size using `CIFilter`. Once it is set up, a `CIDetector` for face detection is created. Since we are only focusing on self-portrait photos with an aspect ratio of 5:4, the `CIDetector` is set to work only in portrait, not landscape. Then, the `CIDetector` is applied on the downsampled core image to detect a human face. `CIDetector` returns a list of features including the eye positions and the mouth position (which will be used for nose detection later).

After testing on real portrait photos, we found that eye detection is not reliable when people are wearing glasses. This could be fixed with an additional step using a custom trained Haar classifier [32], but since this is not the focus of our research, we only recruit participants who do need to wear glasses to use a smartphone.

iOS does not provide any native API for nose detection. Therefore, we use OpenCV's Haar feature-based cascade classifier [32] for nose detection. Instead of searching for the nose in the scaled image, we first crop the image using the face bound provided by the `CIDetector`. To further improve the efficiency of the detection, we only search the area bounded by the eye and the mouth positions.

5.1.2 Face Size Feature Calculation

With eye positions detected, the *face size* ratio is calculated as follows:

$$face\ size\ ratio = \frac{(x_r - x_l) \times 2}{imageWidth\ /\ 3} \tag{5.1}$$

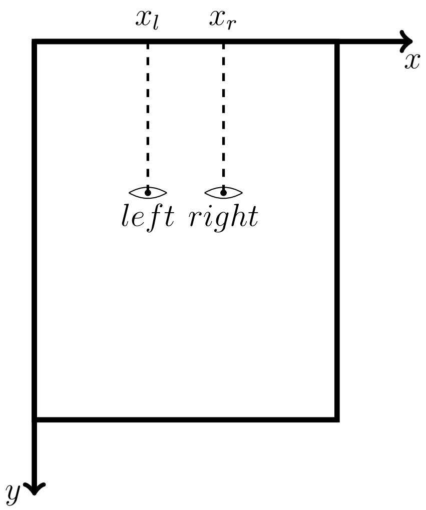


Figure 5.1: We assume both eyes are on the same vertical level so the *face size* ratio can be calculated by dividing the x difference to one third of the image width.

Note that $x_r - x_l$ is the interocular distance and recall that we define face size ratio relative to the width of one cell in a rule-of-thirds grid, thus we divide *image width* by 3. Since we assume the face is oriented straight in the photo without any tilt, the eyes are on the same level as shown in Figure 5.1. Therefore, the only matter to calculate *face size* ratio is the coordinates on the x direction (x_l and x_r).

The *face size* ratio is calculated continuously for each frame with face detected. A first order low-pass filter [4] is applied to stabilize the *face size* ratio across frames.

The direction of improvement on $face\ size$ is calculated by passing the stabilized *face size* ratio into the empirical model (as described in the previous chapter).

5.1.3 Face Position Feature Calculation

Using the detected two-dimensional eye positions $(x, y)_l$ and $(x, y)_r$, the eye centroid is calculated and used as the position of the face. The eye centroid $(x, y)_c$ is calculated in image resolution as follows:

$$(x, y)_c = \frac{(x, y)_l + (x, y)_r}{2} \tag{5.2}$$

Based on the empirical model on *face position*, the direction of improvement is calculated for providing the guidance.

Once the pixel position of the centroid is acquired, it is necessary to convert it to the grid ratio position used as the parameters to the empirical model. The image width w_i and image height h_i can be easily obtained by calling iOS native API. Based on w_i and h_i, the grid's width w_g and height h_g are calculated using the Formula 3.5. Then, Formula 4.5 in the previous chapter is applied to get the grid position (x_g, y_g).

As previously discussed, x_g and y_g should both be in the range 2 to 10. Otherwise, the face will be cropped, even with a small *face size* ratio. If the detected *face position* is out of the range, the closest position will be used.

Like face size, a first order low-pass filter is applied to filter the noise and stabilize the feature value. The stabilized *face position* is passed into the empirical model to calculate the corresponding direction of improvement.

5.1.4 Lighting Direction Feature Calculation

To calculate the lighting direction, we rely on the pattern of luminance around the nose area. We found this to be the most resilient place for this analysis since it contains most lighting information on the face because of nose geometry [10].

Using the detected nose position, a region-of-interest (ROI) around the nose is isolated. To speed up and normalize analysis, the nose ROI is downsampled to 100 pixels wide with the height selected to maintain the original aspect ratio. Since we only use luminance, the nose ROI is converted from BGR to HSV colour space.

After this preprocessing of the nose ROI, we sample the luminance in eight radial 9×9 pixel patches (Figure 5.2). A *patch direction vector* is constructed from the centre of the nose ROI to the centre of each patch. For each sample patch, we find the median luminance l_s and compute the ratio of it over the luminance at the centre of the nose ROI l_s/l_c where l_c is the luminance of the nose tip): . Each *patch direction vector* is then scaled by the corresponding luminance ratio of the patch it points to. Then we sum up the vectors from all eight patches to produce a single lighting direction vector (the red arrow in Figure 5.2). For robustness, we repeat the steps above for different patch radii (the distance from the centre of the region of interest to the centre of each sample patch). We begin with a radius of 9 pixels and increase the radius by 9 pixels after each round of eight patches is sampled.

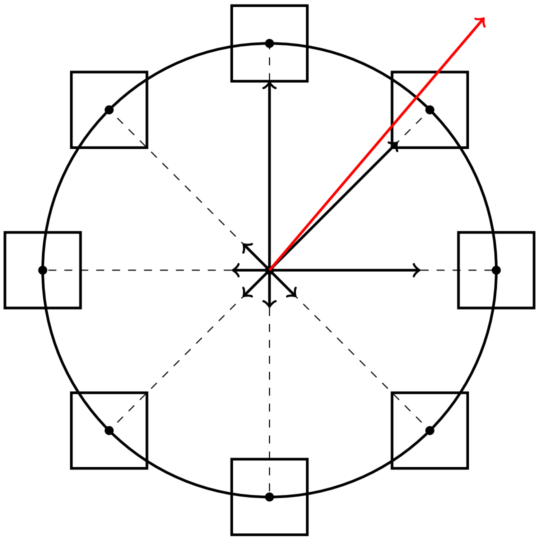


Figure 5.2: Lighting direction analysis for one distance – the *dotted lines* indicate the vectors pointing from the centre to sample regions; based on the ratio, the vectors are shrunk or expanded shown as the *black arrow lines*; the red arrow line is the sum of all *black arrow line* indicating the *lighting direction* vector for this iteration.

The final *lighting direction* vector is the sum of the vectors from all iterations. Pseudo code for the complete lighting direction estimation process is provided in Algorithm 1 and Algorithm 2. The final *lighting direction* vector is filtered by another first order low-pass filter and passed to the lighting model to compute the current lighting score and direction for improvement.

5.1.5 Validation and Testing

The APIs for face and eye detection work well even for extremely small and large faces. However, they are dependent on lighting conditions. If there is not enough ambient light, the face and eyes cannot be detected. Nose detection is not as reliable as face and eye detection, likely due to the limited Haar training dataset. Face detection can be achieved at approximately 12 FPS and nose detection at 8 FPS. Considering how fast people move their phones when composing a photo, these times are acceptable for real time guidance.

We tested our lighting direction estimation algorithm using the synthetic images used for the lighting direction aesthetic rating experiment (see Section 3.1.4). To prevent a confound from imperfect nose detection, we manually insert a transparent plane to each

Algorithm 1 Calculate the *lighting direction* vector *lightDir* for a given nose area image *noseAreaBGR*.

```
Require: $noseAreaBGR$: a BGR image containing a nose
  // Convert the image from BGR to HSV
  $noseAreaHSV \Leftarrow$ cvtColor($noseAreaBGR$)

  // Resize the image to a smaller size and get the V channel
  $aspectRatio \Leftarrow noseAreaBGR$.height / $noseAreaBGR$.width
  $noseAreaHSV \Leftarrow$ resize($noseAreaHSV$, Size(100, $100 \times aspectRatio$))
  $noseAreaV \Leftarrow$ split($noseAreaHSV$)

  // Get the luminance value of the centre
  $l_c \Leftarrow$ luminance of the cetner of $noseAreaV$

  // Initialize the result vector
  $lightDir \Leftarrow (0, 0)$

  // Calculate the lighting direction vector for each distance from innermost to outermost iteratively
  for $d_i$ such that sample regions are within the image do
    $lightDir_i \Leftarrow$ calLightDir($l_c$, $d_i$, $noseAreaV$) // Algorithm 2
    $lightDir \Leftarrow lightDir + lightDir_i$
    $d_i \Leftarrow d_{i-1} + 5$ // Sample size is (10 x 10) pixels
  end for

  return $lightDir$
```

Algorithm 2 calLightDir: Calculate the *lighting direction* vector *lightDir* for l_c and d_i in image *noseAreaV*.

```
Require: $l_c$: the luminance of the centre; $d_i$: distance; $noseAreaV$: an image containing the brightness information of nose area
  // Initialize the result vector
  $lightDir \Leftarrow (0,0)$

  // Calculate the vector for each sample region $s_i$
  for $s_i$ in $noseAreaV$ do
    // Get the median lumiance of the sample region
    $l_{s_i} \Leftarrow$ median luminance of $s_i$

    // Calculate the ratio between $l_{s_i}$ and $l_c$
    $r_{s_i} \Leftarrow l_{s_i} / l_c$

    // Calculate $lightDir_{s_i}$ for each $s_i$
    $lightDir_{s_i} \Leftarrow r_{s_i} \times$ vector from center of $noseAreaV$ to the center of $s_i$

    // Accumulation
    $lightDir \Leftarrow lightDir + lightDir_{s_i}$
  end for

  return $lightDir$
```

realistic human model in Blender and generate nose area masks. Then, combining the nose area mask with the synthetic image, we generate a controlled nose region of interest.

Figure 5.3 compares the estimated lighting direction from our algorithm (shown in red) with a two-dimensional projection of the known actual lighting direction (shown in blue) on all the tested nose regions of interest. As shown in the figure, the algorithm works well when the light comes from the front ($\Phi \in [-90°, 90°]$), but it deviates when the light comes from behind ($\Phi \in [-120°, -90°] \cup [90°, 120°]$) or top ($\Theta = 90°$). We believe that this will not be a problem in real life since people typically do not take photos with light coming from behind.

We also tested the performance of our algorithm on three sets of real photos. This time, we used the detected nose area. The results are shown in Figure 5.4. We took 3 sets of photographs of 3 people (2 male, 1 male, all wearing glasses): one set with the sunlight coming from the top left; one set with the sunlight coming from the top; one set with the sun light coming from the top right. As the figure shows, the algorithm's estimate is consistent for all 3 sets. It also shows that the glasses and moustaches do not affect our lighting direction estimation.

5.2 UI Design

The phone application consists of 2 modes: *Guidance* and *Debug*. In *Guidance* mode, the guidance on each feature is provided based upon the direction of improvement calculated; in *Debug* mode, all information including the face position, eye position, mouth position, the analysed lighting direction, and the scores for all 3 features are drawn. For guidance, different icons are drawn so that users can easily understand it and the experience is enhanced. The *Guidance* mode can be disabled by double-tapping.

The user interface of the RealSelfie app is shown in Figure 5.5. The primary interface components are the guidance visualizations for the three compositional features.

5.2.1 Face Size Guidance

For *face size* guidance, a circle surrounds the face and small arrows are drawn to point outward or inward (e.g. Figure 5.5-b). The arrows indicate whether to move the smartphone closer or further. When the arrows point inwards, the system is suggesting that the smartphone should moved farther away to decrease the size of the face. When the arrows

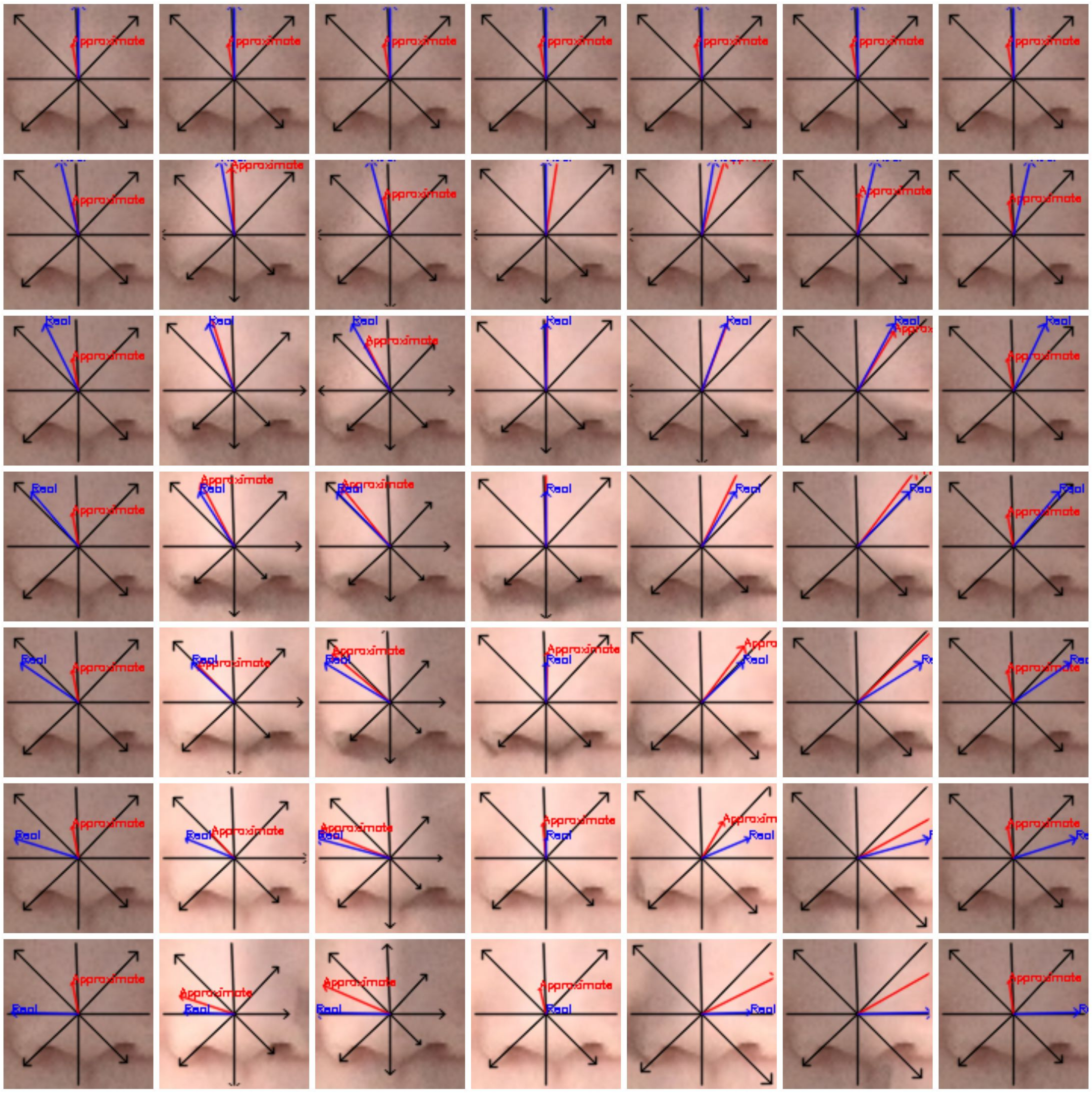


Figure 5.3: Lighting direction analysis algorithm test using synthetic images: *blue arrow lines* are the actual lighting direction projected on the two-dimensional image; the *red arrow lines* are the estimated lighting direction from the algorithm.

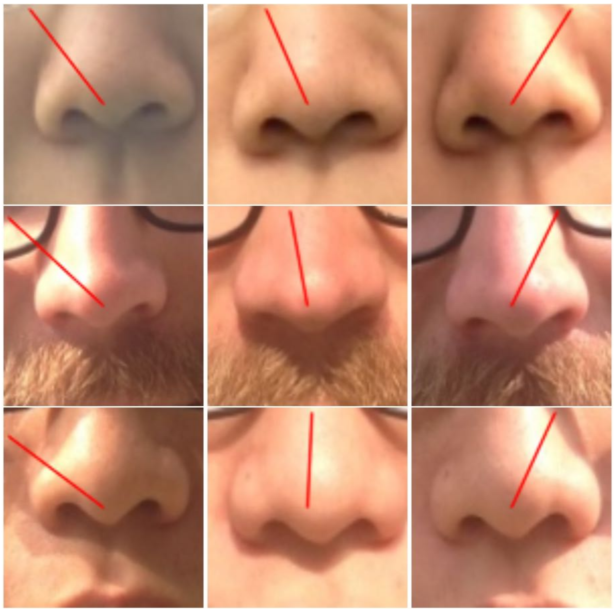
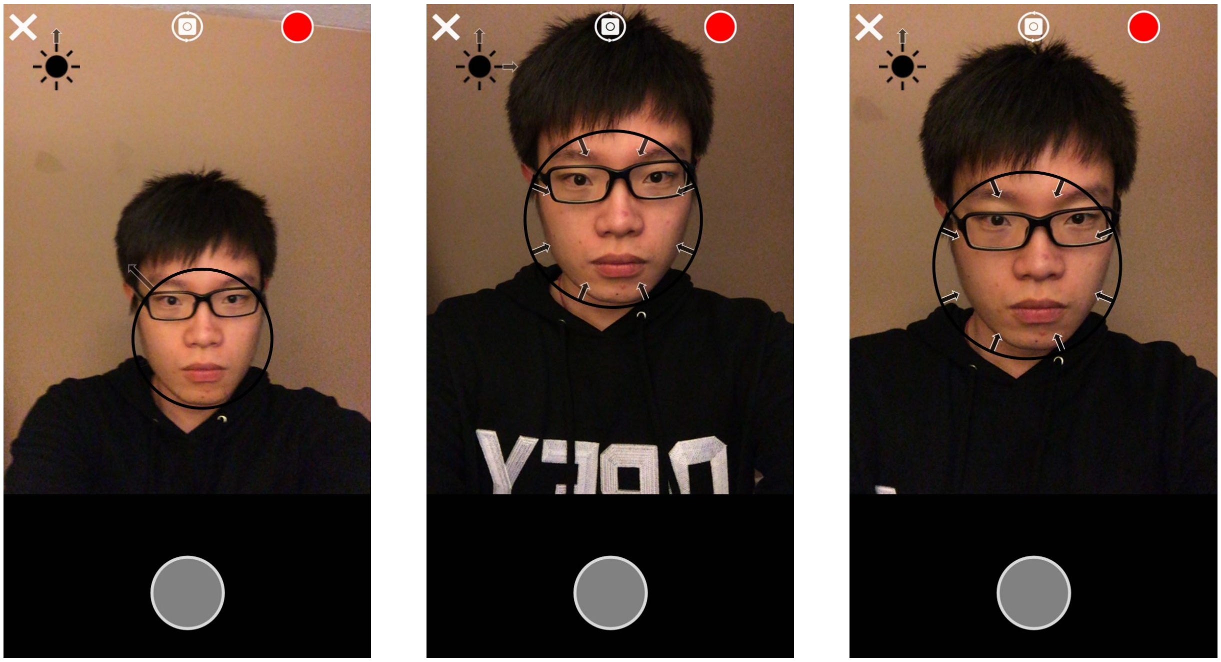


Figure 5.4: Lighting direction analysis algorithm test using photos of real people: *red lines* are the estimated lighting direction; left column: the light comes from top-left; middle column: the light comes from top-middle; right column: the light comes from top-right.



(a) Perfect *face size* (b) Perfect *face position* (c) Good *lighting direction*

Figure 5.5: The UI provides the guidance on each compositional feature: (a) the optimal *face size* is reached with small arrows missing; (b) the optimal *face position* is reached with the large arrow missing; (c) a good *lighting direction* is achieved with translucent arrows on the sun icon.

point outward, the system is suggesting that the smartphone should moved closer to increase the size of the face. The transparency of the arrows indicates the difference between the score of the current *face size* and the highest possible face size score. The metaphor is that when the arrows are very dark, guidance is highly suggested. If the arrows disappear, the current *face size* ratio is optimal (e.g. Figure 5.5-a). The circle also serves to indicate that the face is being tracked correctly.

5.2.2 Face Position Guidance

For *face position* guidance, an arrow with a longer tail is drawn emanating from the tracking circle (e.g., the long arrow pointing NW emanating from the top-left of the circle in Figure 5.5a). This arrow indicates the direction in which the face should be moved. For example, if the arrow points NW, then the smartphone can be slighting tilted SE to move the face up and over. The arrow is drawn based on the direction of improvement returned by the empirical model, thus it can be drawn in 8 directions. Like *face size* guidance, the transparency of the arrow indicates the difference between the current score and the highest score under the current *face size* ratio. If the position is optimal, the arrow will disappear (e.g. Figure 5.5-b).

5.2.3 Lighting Direction Guidance

For *lighting direction* guidance, an icon resembling a sun with two arrows are shown (top-left corner in Figure 5.5). The arrows indicate how to rotate the phone to improve the lighting. For example, arrows pointing up and right indicate that the smartphone should be rotated to the left and up to improve lighting (e.g. Figure 5.5-b) As before, the transparency of the arrows indicates how close the current lighting score is to the optimal score.

5.2.4 Debug Mode

The default mode for RealSelfie is guidance, but it also has a debug mode for testing. In debug mode, the guidance visualization is augmented with tracked positions for the face, eyes, mouth, and nose, the estimated lighting direction vector, as well as the numeric scores for all 3 features. The debug mode is activated or deactivated by double-tapping.

In next chapter, we present the results of an experiment to test the usability and effectiveness of the RealSelfie app.

Chapter 6

Evaluation

We evaluated RealSelfie from two perspectives: application usability and its effectiveness at improving selfie photograph aesthetics. We conducted a usability experiment where people took self-portrait selfie photos with and without RealSelfie in a controlled setting. To evaluate the aesthetic effectiveness, we conducted a second rating experiment on Amazon Mechanical Turk to rate the best pair of photos taken by each participant with and without RealSelfie.

6.1 RealSelfie Usability Experiment

The usability of RealSelfie is investigated through analysis of application logs and questionnaires when people take self-portrait photos with and without RealSelfie in a controlled setting. This first experiment also produced a dataset of photos taken with and without RealSelfie that we use in the second experiment to evaluate the effectiveness of RealSelfie.

6.1.1 Participants

We recruited 20 participants from a university campus (11 female and 9 male, mean age 24.4). Our participants had a high level of visible diversity with different skin pigments and facial features. We limited participants to people who could view the smartphone screen without eye glasses since the eye detection algorithm is less reliable with dark rimmed eye wear. Our participants had a variety of smartphone selfie taking experience: 7 took selfies

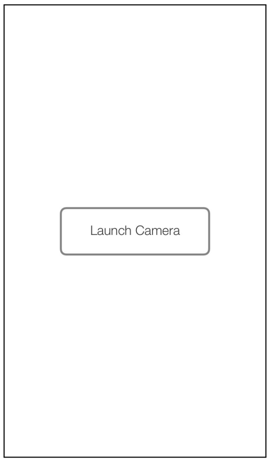


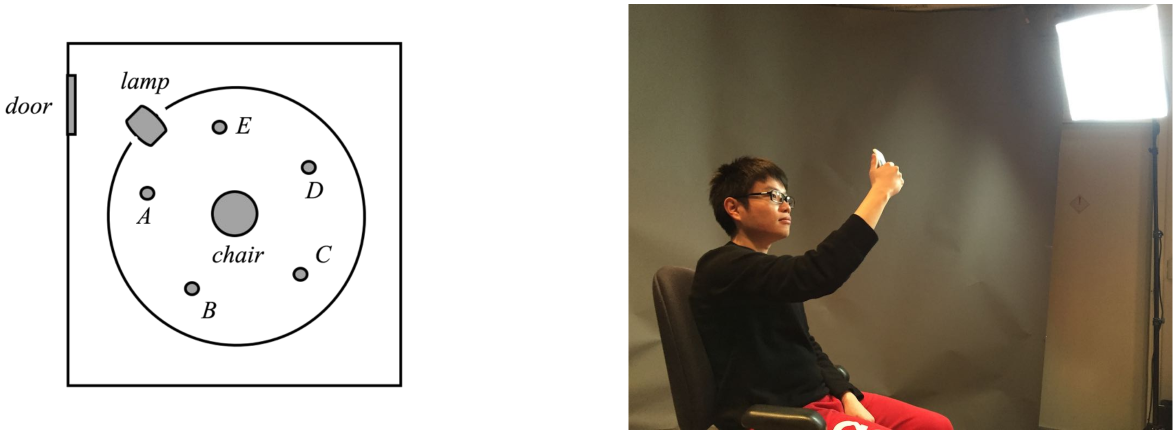
Figure 6.1: The timer is started once the *LaunchCamera* button in the middle is tapped whenever the participant is ready to take a photo.

with their smartphone daily or weekly, 10 took selfies monthly or yearly, and 3 almost never take selfies. Only 1 participant had taken a course in photography.

6.1.2 Apparatus

We used the RealSelfie described in the previous chapter running on an iPhone 6. The app can run with or without visual guidance. Regardless whether visual guidance was shown, we instrumented the app to run the full compositional feature analysis and compute the scores and direction of improvement. This ensured that the refresh rate of the preview mode was the same regardless of guidance and most importantly, provided a log of quantitative data to test whether people actually improved aesthetic ratings (as determined by our models) using guidance. To record the time for taking one photo, we implemented a *LaunchCamera* button (shown as in Figure 6.1) for experiments. As the button is tapped, the starting time stamp is logged. When the participant captures one photo by either tapping the circle button or pressing the volume button, a picture taken time stamp is logged as well. With these two events, we can calculate the time to take a picture.

To avoid the background affecting the participants' assessments in the second experiment, we set up a room as shown in Figure 6.2b. It is constructed as Figure 6.2a. Inside a room, we use grey background paper (height: 2.7m) to set up a circle. We fix a chair at the centre of the circle. Instead of doing the experiment outdoors with the natural sunlight, we chose to build an indoor studio so that the lighting condition is highly controlled. The participant can rotate to adjust the *lighting direction* while sitting in the chair. The ambient light is generated by using 5 bulbs to shoot straight on the ground. Two more poweful bulbs are used to mimick the sunlight. This setup enables us to highly control every features including both lighting and background so that either will not be the influential factor causing the aesthetic difference between the pair of images.



(a) *lamp* shoots parallel light mimicking the sunlight; *A* – *E* shoot straight light on the ground to generate ambient light.

(b) The participant sits in the chair, moves the camera to manipulate *face size, face position* or rotates to adjust *lighting direction*.

Figure 6.2: The plan view and real scene of the studio we set up for participants taking photos.

6.1.3 Task and Protocol

The experiment had two parts. Before starting each part, participants were asked to only focus on the three compositional factors, *face size*, *face position*, and *lighting direction*. They can move the camera closer or further to adjust the *face size*, change the *face position* by tilting their wrists, and rotate in the chair to change the *lighting direction*. To avoid the case of other factors such as hairstyle, clothing, posture, etc. affecting the aesthetics of photos, we specifically asked them to focus on those 3 features and keep anything else consistent. For example, if they have one facial expression in one photo, they have to do the same one in all the other photos. They were also warned that the ultimate goal was to capture 5 most appealing self-portrait "selfie" photos in each part.

In the first part, they took 5 self-portrait "selfie" photos without guidance. In the second part, RealSelfie guidance was turned on and they took 5 more self-portrait "selfie" photos with guidance. They were told they could follow or ignore the guidance suggestions. After both parts were completed, the participant selected the best photo among the five taken without guidance and the five taken with RealSelfie.

Then subjective feedback on RealSelfie was gathered using a post-experiment questionnaire Appendix B. The participants are asked to give a score from 1 to 5 on a continuous scale for *Ease of Learning*, *Ease of Use*, *Accuracy of Guidance*, *Operation Speed*, and *Hand Fatigue* (with 1 being worst and 5 being best). They were also asked to answer if they thought the guidance is helpful and would use an app like this.

The experiment took 31.25 mins on average,.

6.1.4 Design

This is a within subjects experiment. The independent variable is GUIDANCE with two levels: BASELINE, when the camera application has no guidance and REALSELFIE, when the full RealSelfie camera application is used. Since there would be a very strong carry over effect if REALSELFIE preceded BASELINE, each participant took 5 photos with BASELINE then took five photos with REALSELFIE as explained in the task above.

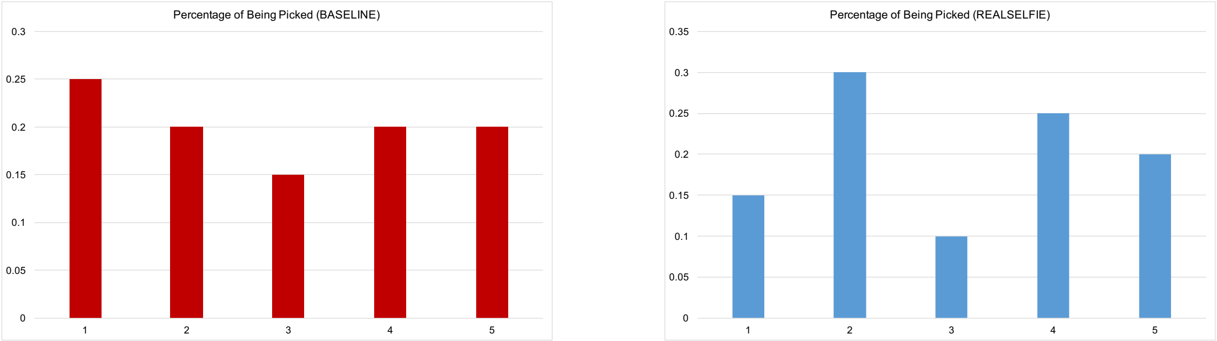
Our dependent measures are the photo-taking time and compositional feature scores for the two types of GUIDANCE, as well as subjective feedback for REALSELFIE in the questionnaire.

6.1.5 Results

Since we are only interested in differences between two levels of GUIDANCE, we use a t-test for 2 related samples with a critical value of .05 for statistical tests. We computed the dependent measures for photo-taking time and compositional feature scores from the application event logs using a custom parser written in Python.

Selected Photos

For each level, we calculated the percentage of each photo being selected as the most appealing one among the 5 photos. As the higstograms in Figure 6.3 show, there is no preference in BASELINE or REALSELFIE.



(a) Participants' selection in BASELINE

(b) Participants' selection in REALSELFIE

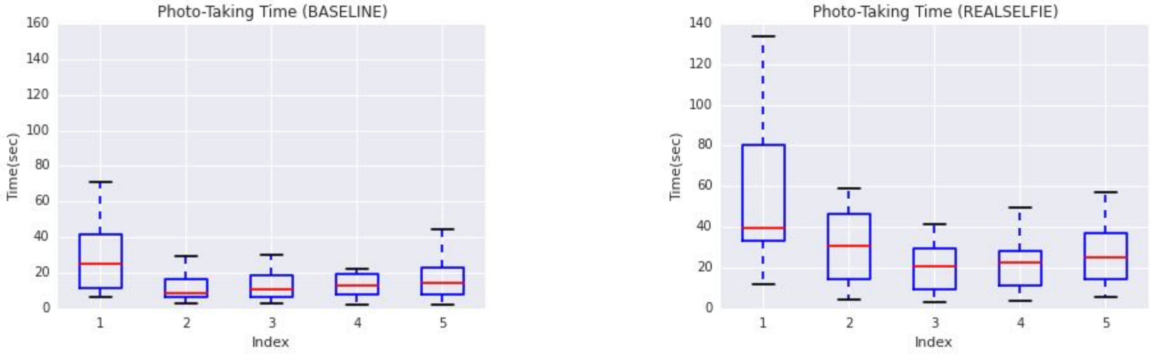
Figure 6.3: Histograms showing selected photo index percentages: (a) BASELINE; (b) REALSELFIE.

Photo-Taking Time

The photo-taking time is the duration between the "launch camera" button press and the shutter button press. Two box plots (Figure 6.4) are drawn to visualize the time spent for both BASELINE and REALSELFIE by all 20 participants. As the plots show, the time for the first photo-taking in each set is much larger than the others. We believe it is because participants tried to learn the application and get to know the new circumstance (the studio).

First, we calculated the average photo-taking time for BASELINE and REALSELFIE for all 5 photos. Then, we repeated the process but only for 4 photos with the first one excluded since the average time is independent of the learning process. Both the average and SEM are shown in Figure 6.5. By comparison, without the time cost for first photo, both the average and SEM are reduced.

We found a significant difference in photo-taking time when considering all photos; $t(198) = -4.79$, $p < 0.01$ (Figure 6.5a). The average time in BASELINE was 20.0 sec (SD 22.0) and 32.9 sec (SD 26.8) for REALSELFIE. A significant difference in photo-taking time is also found when considering only 4 photos; $t(158) = -4.22$, $p < 0.01$ (Figure 6.5b). The average time in BASELINE was 16.04 sec (SD 14.66) and 27.61 sec (SD 21.24) for REALSELFIE. The photo-taking time is much more than the reality. We believe the reason is that the studio is a new scene for the participants, not the one they find interesting in the first place and decide to take a photo in as in real life. It takes them time to investigate the space to find the interesting spots. After spending time on exploration, they take photos



(a) Photo-taking time in BASELINE

(b) Photo-taking time in REALSELFIE

Figure 6.4: Box plots showing photo-taking time for both (a) BASELINE and (b) REALSELFIE.

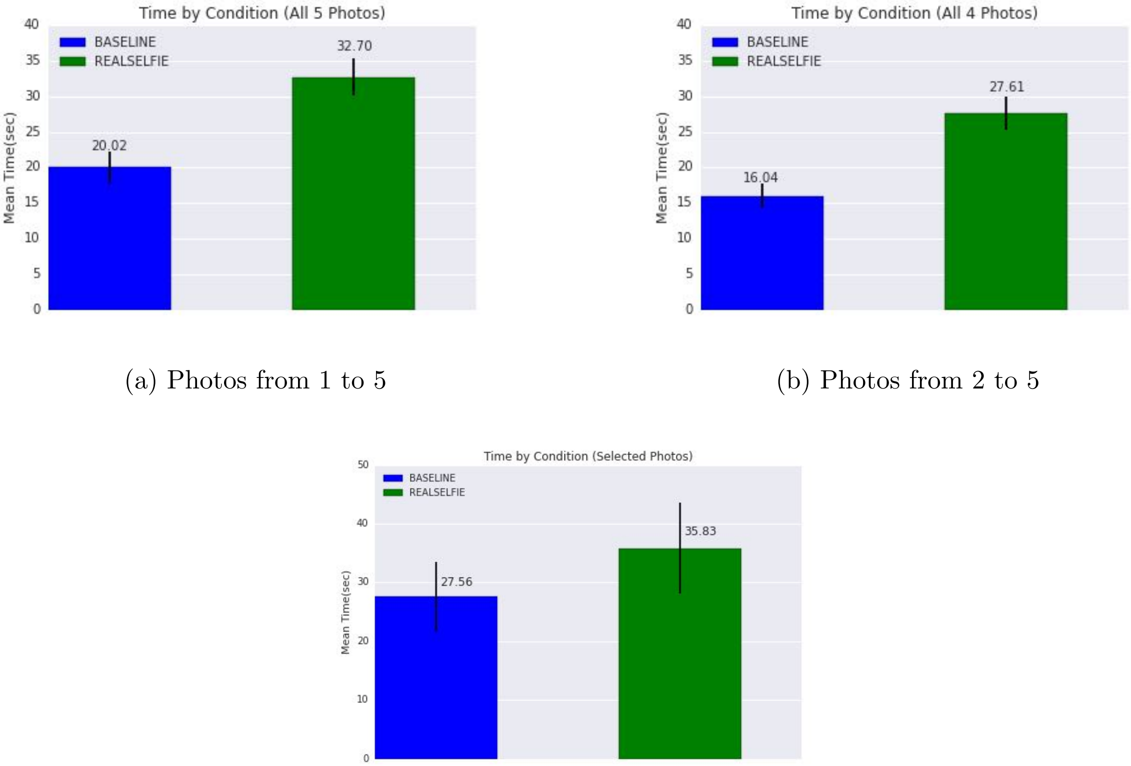
then.

However, we did not find a significant difference in photo-taking time when considering the selected photo only; $t(38) = -0.81$, $p = 0.43$ (Figure 6.5c). The average time in BASELINE was 27.6 sec (SD 27.5) and 35.8 sec (SD 35.7) for REALSELFIE.

Feature Scores

The feature scores of the best photo in each part are also compared. Besides the time stamp, the scores for all 3 features are also logged when the photo is captured. Therefore, for each participant, 10 sets of feature scores are recorded. Once the experiment is done, the participant is asked to pick the most appealing photo from each set. The indices of the two photos are recorded. In the post-experiment analysis, the parser filters the feature scores for the best 2 photos based on the indices and grouped into 2 sets: BASELINE and REALSELFIE.

The average score, standard deviation, and standard error for each feature in both BASELINE and REALSELFIE are calculated shown as in Table 6.1. The average score and standard error are drawn in Figure 6.6. We also run a T-test for each feature which is also shown in Table 6.1. There is a significant difference between BASELINE and REALSELFIE for *face size* and *lighting direction*. However, there is no enough evidence to prove the existence of the significant difference for *face position*. From the statistics, we can conclude that RealSelfie does improve the aesthetics on *face size* and *lighting direction* but not *face*



(a) Photos from 1 to 5

(b) Photos from 2 to 5

(c) Selected Photos

Figure 6.5: Average and SEM photo-taking time for two sets: (a) all 5 photos, (b) 4 photos without the first one, (c) selected photos.

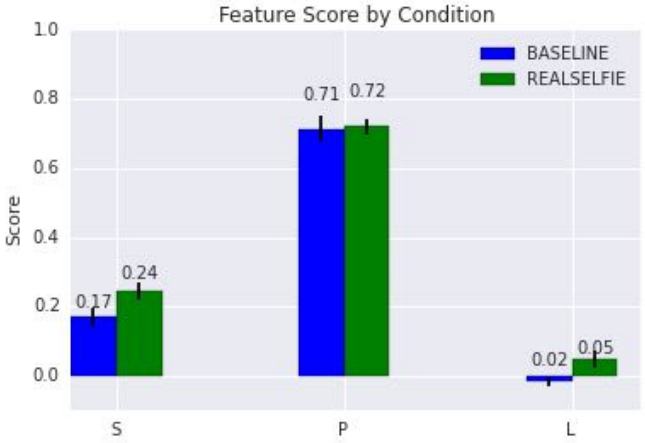


Figure 6.6: The average time for taking one photo without and with guidance.

position. By checking the photos taken by participants, we find that participants normally just place their faces in the center area of the image even without guidance which is not very different than the position provided by the synthetic model. This is why there is no significant difference between without and with guidance on *face position*.

User Experience

The user experience of RealSelfie is investigated through a post-experiment questionnaire shown in Appendix B. The participants are asked to give a score from 1 to 5 on a continuous scale for *Ease of Learning*, *Ease of Use*, *Accuracy of Guidance*, *Operation Speed*, and *Hand Fatigue* (with 1 being worst and 5 being best). The average is shown in Table 6.2. We can tell that RealSelfie does provide a good user experience for participants to take self-portrait photos.

For the relatively lower scores on *Ease of Use* and *Accuracy of Guidance*, we found that it was caused by the flickering of the arrows on *lighting direction* guidance by investigating the feedback provided by the participants. Also, in the experiment, we realized we did not give partitipants enough time to rest. This could be the reason for the lower score on *Hand Fatigue*.

	Baseline	RealSelfie	*t value*	*p value*

Face Size	Mean = 0.170
	STD = 0.122
	SEM = 0.027
		Mean = 0.245
		STD = 0.103
		SEM = 0.023
			-4.16	0.0001

Face Position	Mean = 0.714
	STD = 0.163
	SEM = 0.037
		Mean = 0.723
		STD = 0.100
		SEM = 0.022
			-0.23	0.82

Lighting Direction	Mean = -0.019
	STD = 0.052
	SEM = 0.012
		Mean = 0.049
		STD = 0.108
		SEM = 0.024
			-2.30	0.033

Table 6.1: The *average*, *standard deviation*, *standard error*, and *t-test* results for each feature in baseline and realselfie.

Ease of Learning	4.225
Ease of Use	3.95
Accuracy of Guidance	3.725
Operation Speed	4.25
Hand Fatigue	3.825

Table 6.2: The average scores for evaluating the user experience of RealSelfie.

6.1.6 Discussion

By recruiting participants to take photos without and with RealSelfie, we found that RealSelfie does improve the aesthetic scores on *face size* and *lighting direction*. And there is no enough evidence to make the same statement on *face position*. By looking at the photos taken without RealSelfie, we found that most participants centre their faces consciously. Therefore, there is no significant difference on between the scores with and without RealSelfie. In the end, We can conclude that RealSelfie does improve the overall aesthetic score of selfies but with the sacrifice of time.

There are short term refinements to RealSelfie. Some participants complained about the sometimes ambiguous guidance on *lighting direction* and participants with very dark hair sometimes found it difficult to notice the arrow for *face position*. We believe both of these minor problems can be corrected with improved graphic design.

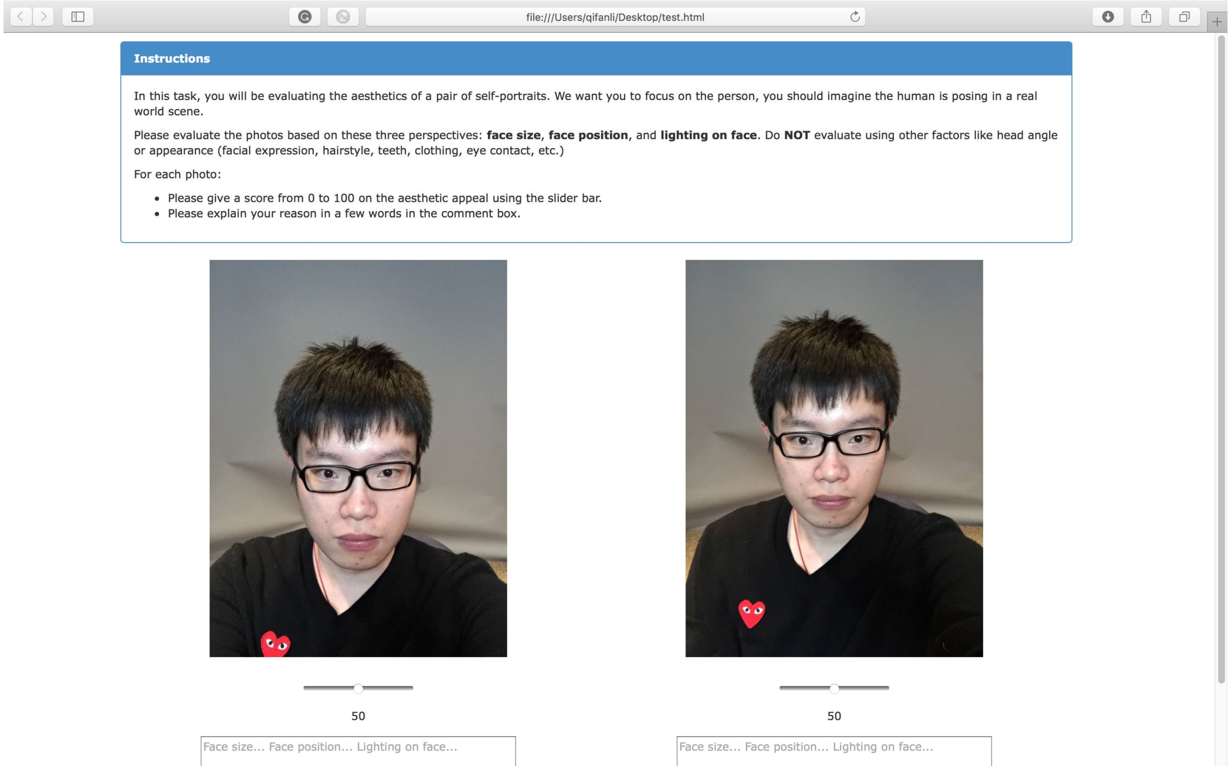
In next section, we describe an experiment to evaluate selfie photos taken with and without RealSelfie to evaluate the *effectiveness* of RealSelfie.

6.2 Effectiveness Experiment

The goal of this experiment is to evaluate whether RealSelfie actually improved the overall aesthetics of a selfie photograph. Through the experiment described in the previous section, 20 pairs of self-portrait photos were acquired: one without guidance and the other one with guidance. Then we created a task for each pair of photos on AMT and recruited 100 workers to rate each pair of photos. The photos are graded with a scale from 0 to 100. Then, the scores of photos are grouped into 2 sets: BASELINE (without guidance) and REALSELFIE (with guidance). Finally, we apply statistical methods to find if a significant improvement is made by using RealSelfie.

6.2.1 Participants

To investigate the aesthetics, we recruit 100 workers for each task (Human Intelligence Task — HIT). For general assessment, no prerequisite is asked for participants.



Instructions

In this task, you will be evaluating the aesthetics of a pair of self-portraits. We want you to focus on the person, you should imagine the human is posing in a real world scene.

Please evaluate the photos based on these three perspectives: **face size**, **face position**, and **lighting on face**. Do **NOT** evaluate using other factors like head angle or appearance (facial expression, hairstyle, teeth, clothing, eye contact, etc.)

For each photo:

- Please give a score from 0 to 100 on the aesthetic appeal using the slider bar.
- Please explain your reason in a few words in the comment box.

50

Face size... Face position... Lighting on face...

50

Face size... Face position... Lighting on face...

Figure 6.7: The user interface on AMT for workers to grade both photos with a scale of 0 to 100 through a slider bar. The two containers show the photos, and the text boxes below require the feedback on *face size*, *face position*, and *lighting direction*.

6.2.2 Apparatus and Task

For each pair of photos, we created a HIT using the UI shown in Figure 6.7. In each HIT, the pair of photos is shown in the two containers. For workers' convenience, two slider bars are implemented to allow workers to grade each photo with a score ranging from 0 to 100. The participants were also asked to provide the feedback regarding the only 3 features: *face size*, *face position*, and *lighting direction*. Then, they could use the slider bar to grade each photo.

6.2.3 Design

To avoid that any participant cheating in all HITs, we randomize the order of the photos in each HIT (i.e. the photo taken using RealSelfie could be on the left or right).

Only 15 out of the 20 pairs were assessed by the workers on AMT. One pair was disregarded due to inconsistent *facial expression*. Two pairs were abandoned because the participants completely ignored the guidance which results in very similar photos. The other two pairs were ignored because the participants were distracted by the guidance and captured much less appealing photos.

From the assessments of the 15 pairs of photos, we group the scores to 2 categories: BASELINE and REALSELFIE.

6.2.4 Results

Since we are only interested in differences between two levels of GUIDANCE, we use a t-test for two related samples with a critical value of .05 for statistical tests. We computed the aesthetic scores for both 2 categories using a custom parser written in Python.

Aesthetic Rating

We found that selfie photographs taken with REALSELFIE are rated as more aesthetic appealing; $t(2998) = 16.33$, $p < .01$ (Figure 6.8). The average rating for REALSELFIE photos is 68.93 (SEM 0.53) compared to 54.82 (SEM 0.62) for BASELINE. This is a 26% improvement in aesthetic rating when using RealSelfie. The average score, standard deviation, and standard error are provided in Table 6.3.

With furher investigation on each pair, we found the pair with greatest improvement. A t-test was run over the 100 pairs of scores with $t(98) = -12.47$, $p < .0001$. The average score of the photo without RealSelfie is only 45.43 whereas the average is 77.58. By looking at the photos, we found that with RealSelfie, both the lighting and face position are dramatically improved. Within the 15 pairs, we also investigated the worst pair for which the average score with RealSelfie (61.08) is lower than that wihout RealSelfie (61.53). Another t-test was runn and proved that there was no significant difference between REALSELFIE and BASELINE ($t(98) = 0.12$, $p = 0.90$). The aesthetics of the photo with RealSelfie was improved on *face size* but diminished on *lighting direction*. We believe that it was caused by the flickering of the arrows on *lighting direction*.

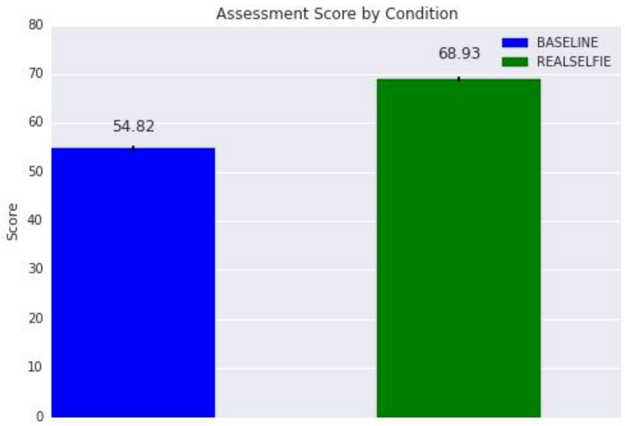


Figure 6.8: The average scores and standard errors of the means for *Baseline* and *RealSelfie* are drawn for visualization.

	RealSelfie	Baseline
Mean	68.93	54.82
Standard Deviation	20.27	23.72
Standard Error	0.61	0.72

Table 6.3: The *average*, *standard deviation*, and *standard error* for each feature score without and with guidance.

6.2.5 Worker Comments

Other than the aesthetic scores of all photos, we also took a quick look at the feedback regarding the three compositional rules: *face size*, *face position*, and *lighting direction*. We found that RealSelfie improved the aesthetics on *face position* and *lighting direction*. It appeared that people had various opinions on *face size*. Some workers also critiqued that the photos taken with RealSelfie were washed out. We think this can be fixed with more natural light.

6.2.6 Discussion

By recruiting participants on AMT to assess the aesthetics of the photos taken without and with RealSelfie, we find that RealSelfie does improve the aesthetics of self-portrait photos with guidance provided while participants are taking photos.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we presented a multi-phase research methodology to build and validate RealSelfie, a smartphone camera application assisting people to take more appealing self-portrait "selfie" photos.

We first generated many sets of synthetic selfies using realistic 3D models of humans where we tightly control three key compositional features: *face size*, *face position*, and *lighting direction*. We used these synthetic selfies in a large crowdsourcing experiment in which we gathered thousands of ratings about their aesthetic quality. These ratings illuminated some fundamental patterns in preference, some of which diverge from accepted principles like the rule-of-thirds. But most importantly, we used these ratings to build three empirical models to estimate an aesthetic score, and direction for aesthetic improvement, given the current state of each compositional principle. The RealSelfie app uses computer vision techniques to detect the state of each compositional principle in realtime and by passing these detected states to the three empirical models, the app provides on-screen guidance indicating how to move the smartphone to improve the selfie. In the process of developing this app, we also contribute a simple, but effective algorithm to estimate the direction of a dominant light source. Finally, we evaluated the usability of RealSelfie with 20 participants and show that our entire approach can increase the aesthetic ranking of selfies by 26%.

We not only introduced a system which improves the aesthetic quality of selfies, but we believe our methodology can be extended to enhance other features or other classes of photographs.

7.2 Future Work

Our work also suggests avenues for future research.

We only considered the three compositional features. There is a large space of other features that could be investigated in a similar method. These include other compositional features, such as head tilt, camera angle, multi-point lighting, focal length, colour balance, depth of field, background contrast. But also non-compositional features like facial expression, eye gaze, hair style, clothing.

Our focus was on single person portraits, but related classes of selfies could be explored directly using our methodology. For example, two person selfies, selfies of a person in front of a structure, and small group selfies.

We also believe our methodology can be applied for photographs taken of other objects, like cars, children, pets, buildings, sunsets, sports, etc. A closely related class of photographs is the "mirror selfie" which would require a redesign to our user interface since the smartphone display is not the focus of the picture taker.

APPENDICES

Appendix A

Pre-Experiment Questionaire for In-Lab Study

PARTICIPANT # ______________ DATE ______________________

Pre-experiment Questionnaire

1. Gender: *Male* *Female*
2. Age: _______
3. Have you ever taken a course in photography?

   Yes *No*

4. How frequently do you take *photos* using your smartphone?

   Daily

   Weekly

   Monthy

   A Few Times Per Year

   Almost Never

5. How frequently do you take *self portrait photos (selfies)* using your smartphone?

   Daily

   Weekly

   Monthy

   A Few Times Per Year

   Almost Never

Appendix B

Post-Experiment Questionaire for In-Lab Study

PARTICIPANT # _______________ DATE ________________________

Post-experiment Questionnaire and Interview

Part 1: Taking and Choosing Photos

1. What kinds of things were you considering when you took photos without guidance from the app?

2. What kinds of things were you considering when you chose the best photo ?

PARTICIPANT # ________________ DATE _________________________

Part 2: Technique Ratings

Please fill out the following questionnaire in the scale from 1 to 5
(with 1 being worst difficult and 5 being best).

Technique: ___________________________________

Ease of learning		**Ease of use**	
Score	Comments	Score	Comments
Accuracy of guidance		**Operation speed**	
Score	Comments	Score	Comments
Hand Fatigue			
Score	Comments		

PARTICIPANT # ______________ DATE ______________________

Part 3: Comments

1. Do you think the guidance is helpful?

2. Would you use an app like this?

3. Additional comments?

References

[1] Saeideh Bakhshi, David A. Shamma, and Eric Gilbert. Faces engage us: Photos with faces attract more likes and comments on instagram. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 965–974, New York, NY, USA, 2014. ACM.

[2] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 33–42, New York, NY, USA, 2011. ACM.

[3] Subhabrata Bhattacharya, Rahul Sukthankar, and Mubarak Shah. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 271–280, New York, NY, USA, 2010. ACM.

[4] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530. ACM, 2012.

[5] Xiaowu Chen, Xin Jin, Hongyu Wu, and Qinping Zhao. Learning templates for artistic portrait lighting analysis. *Image Processing, IEEE Transactions on*, 24(2):608–618, Feb 2015.

[6] John Child. *Studio Photography: Essential Skills.* Focal Press, 2008.

[7] Casio Computer Co. Life style digital cameras. [Hardware: Digital Cameras].

[8] Ritendra Datta, Dhiraj Joshi, Jia Li, and JamesZ. Wang. Studying aesthetics in photographic images using a computational approach. In Ale Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision ECCV 2006*, volume 3953 of *Lecture Notes in Computer Science*, pages 288–301. Springer Berlin Heidelberg, 2006.

[9] S. Dhar, V. Ordonez, and T.L. Berg. High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664, June 2011.

[10] Dmitry O Gorodnichy and Gerhard Roth. Nouse use your nose as a mouseperceptual vision technology for hands-free games and interfaces. *Image and Vision Computing*, 22(12):931–942, 2004.

[11] Melanie Hall. Family albums fade as the young put only themselves in picture, June 2013.

[12] Bill Hurter. *Portrait Photographer's Handbook*. Amherst Media, 3 edition, 8 2007.

[13] Wei Jiang, A.C. Loui, and C.D. Cerosaletti. Automatic aesthetic value assessment in photographic images. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 920–925, July 2010.

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

[15] Robin Kelsey and Blake Stimson. *The Meaning of Photography*. Clark Art Institute, 2008.

[16] Shehroz S. Khan and Daniel Vogel. Evaluating visual aesthetics in photographic portraiture. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, CAe '12, pages 55–62, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

[17] Normand M. Laurendeau. Appendix i: The spherical coordinate system, 2005. [Online; accessed April, 2016].

[18] C. Li, A. Gallagher, A. C. Loui, and T. Chen. Aesthetic quality assessment of consumer photos with faces. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3221–3224, Sept 2010.

[19] Congcong Li and Tsuhan Chen. Aesthetic visual quality assessment of paintings. *Selected Topics in Signal Processing, IEEE Journal of*, 3(2):236–252, April 2009.

[20] Arnaud Lienhard, Patricia Ladret, and Alice Caplier. How to predict the global instantaneous feeling induced by a facial picture? *Signal Processing: Image Communication*.

[21] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900–I–903 vol.1, 2002.

[22] LuMee LLC. Lumee phone case. [Hardware: Phone Accessory].

[23] Ty-Lite LLC. Ty-lite phone case. [Hardware: Phone Accessory].

[24] Jorge Lopez-Moreno, Sunil Hadap, Erik Reinhard, and Diego Gutierrez. Light source detection in photographs. In *Congreso Espanol de Informatica Grafica*, pages 161–168, 2009.

[25] Wei Luo, Xiaogang Wang, and Xiaoou Tang. Content-based photo quality assessment. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2206–2213, Nov 2011.

[26] Shuang Ma, Yangyu Fan, and Chang Wen Chen. Finding your spot: A photography suggestion system for placing human in the scene. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 556–560, October 2014.

[27] M. Males, A. Hedi, and M. Grgic. Aesthetic quality assessment of headshots. In *ELMAR, 2013 55th International Symposium*, pages 89–92, Sept 2013.

[28] Lev Manovich. Selfiecity. http://selfiecity.net. 2014.

[29] F. Mazza, M. P. Da Silva, P. Le Callet, and I. E. J. Heynderickx. What do you think of my picture? investigating factors of influence in profile images context perception, 2015.

[30] Filippo Mazza, Matthieu Perreira Da Silva, Patrick Le Callet, and IEJ Heynderickx. What do you think of my picture? investigating factors of influence in profile images context perception. In *IS&T/SPIE Electronic Imaging*, page 93940D. International Society for Optics and Photonics, 2015.

[31] P. Obrador, L. Schmidt-Hackenberg, and N. Oliver. The role of image composition in image aesthetics. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3185–3188, Sept 2010.

[32] OpenCV. Cacade classification. [Online; accessed April, 2016].

[33] Bahareh Rahmanian and Joseph G Davis. User interface design for crowdsourcing systems. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 405–408. ACM, 2014.

[34] Miriam Redi, Nikhil Rasiwasia, Gaurav Aggarwal, and Alejandro Jaimes. The beauty of capturing faces: Rating the quality of digital portraits. *CoRR*, abs/1501.07304, 2015.

[35] Rossano Schifanella, Miriam Redi, and Luca Maria Aiello. An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. *CoRR*, abs/1505.03358, 2015.

[36] Spencer Soper. Selfie sticks rule holiday season as must-have accessory, December 2014. [Online; posted 31-December-2014].

[37] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[38] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[39] Lai-Kuan Wong and Kok-Lim Low. Saliency-enhanced image aesthetics class prediction. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 997–1000, Nov 2009.

[40] Yan Xu, Joshua Ratcliff, James Scovell, Gheric Speiginer, and Ronald Azuma. Real-time guidance camera interface to enhance photo aesthetic quality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1183–1186, New York, NY, USA, 2015. ACM.

[41] Shao-Fu Xue, H. Tang, D. Tretter, Qian Lin, and J. Allebach. Feature design for aesthetic inference on photos with faces. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2689–2693, Sept 2013.