Aesthetic Image Rating (AIR) Algorithm

David Reaves
College of Natural Science,
University of Texas at Austin,
Texas,
nystagmust@gmail.com

April 18, 2008

Contents

Abstract

Rapidly advancing technologies offer a greater volume of people the possibility to both create and consume information. And, with this widening of opportunity, the volume of digital information has increased in mammoth proportion. Indeed, this age of information is marked by quantity, but what of quality? It has become necessary to formulate a systematic method to sift through the vast amount of data. This paper presents an algorithm that seeks to emulate the manner by which a human might judge an image's aesthetic value. The notion that a machine could imitate human thought processes is not necessarily novel, and, as such, a fair amount of work has been done regarding algorithmic aesthetic digital image rating. Most of these proposed algorithms, however, have been unable to satisfactorily mimic actual human ratings. This paper builds on these past works and yet goes further by significantly improving on these prior accomplishments. The result of our focus on the discovery of an optimal vector of image features is a highly accurate emulation of human ratings.

1 Introduction

The so-called "Information Age" denotes our current era in which the global economy has become increasingly dependent on the manufacture and exchange of digital information. Rapidly advancing technologies offer a greater number of people the possibility to both create and consume information. And, with this widening of opportunity, the volume of digital information has increased in mammoth proportion.

Currently, a prevailing currency of our global economy is information, and this information is available in gargantuan amounts. A growing number of companies engage solely in the business of exchanging information such as music, videos, and images. One such company is Corbis. Since its inception, Corbis has accumulated more than 100 million photographs and over 25,000 videos[5]. By understanding how the masses may respond to a video or photograph, a company such as Corbis can more quickly determine the potential value of their merchandise.

An article published three years ago entitled *Hit Song Science* brought attention to a service that, essentially, places a value on music[9]. The service grades a song based on how the masses might rate that song. This program plots songs in a multidimensional space based on various features, and automatically rates the songs according to their placement in that space. I used this as evidence that any artistic medium can be decomposed into basic components, and can be subsequently rated accordingly.

I sought to create an Aesthetic Image Rating (AIR) algorithm that will faithfully replicate an average human's rating for any given photograph. Such an algorithm has a myriad of applications. The most immediate application would be to provide users of photograph management software with initial ratings for imported images, allowing them to focus on more promising images. This algorithm can also be used to aid media dealers, such as Corbis, by setting appropriate prices for images. Finally, such an algorithm is useful for the process of image retrieval by search engines.

It is necessary to differentiate the goal of my algorithm from the goals of previous image Quality Assessment(QA) algorithms. There are three types of image QA algorithms Full-Reference QA, Reduced-Reference QA, and No-Reference QA. Full-Reference QA and Reduced-Reference QA are typically used to measure distortions or artifacts introduced by image compression algorithms. Full-Reference QA implies the original image is available whereas Reduced-Reference implies some knowledge about how the image was altered or compressed. My algorithm falls within the realm of No-Reference QA, and specifically focuses on the aesthetic aspect of the images.

I chose to approach this No-Reference QA problem in the same manner as had been presented in the *Hit Song Science* article. First, I decompose the image into a vector of features, such as saturation, contrast, and more. To clarify, a feature is an algorithm that returns a rating when applied to an image. I was able to plot each image in a multidimensional space based on the feature vector I obtained. This allowed me to then perform a regression on the points in this space; that regression gave me the rating function I desired.

2 Related Works

2.1 Related Works Methodology

The concept of aesthetic-based image rating is not a novel one. The goal of understanding aesthetics is as old as art itself; from Leonardo da Vinci's golden ratio analysis to Rudolf Arnheim's *Art and Visual Perception: A Psychology of the Creative Eye* [2]. More recently the image retrieval community has made more substantive progress towards understanding aesthetics. Their motivation stems from the desire to create algorithms that return the most pleasing image with a particular tag rather that just a random set of those images. In particular, four recent papers reflect the advancements made towards understanding what features correlate to users' ratings, and what is an optimal approach to process these features.

Each paper applies a standard machine learning technique to attempt to empirically determine what features are most aesthetically important. The technique can be reduced to a simple high-level algorithm. The algorithm requires you to start by generating as many features as possible. Then, because features that are poor discriminators actually tend to hurt performance, they are removed through a feature selection process. Finally, a classification or regression algorithm is applied on the optimal set of features.

Classification is simply the process of using a algorithm to determine if a document with a certain feature vector belongs to one of an integer number of classes. Regression involves fitting a continuous function to the multidimensional feature space. A two-dimensional visual representation of these processes is displayed in Figure 1.

Figure 1: Classification and Regression

2.2 Previous Publications

Tong et al. [13] is the first paper published that I was able to find that dealt with automatic image rating. Initially it seems as if the paper should be the most insightful since it has the largest set of features, training images, and classification techniques. However this paper is rendered useless by its refusal to discuss, in sufficient detail, the features or the feature selection results.

The Design of High-Level Features for Photo Quality Assessment by Ke et al. [11] even made a point to note deficiencies in the Tong et al. paper. Ke et al. took a large dataset from dpchallenge.com and classified the top 10% as professional photographs and the bottom 10% as snapshots. The paper then explored the fundamental differences between these two sets. Through this process they discovered several important features. The two best features were the blurriness of an image and the spatial distribution of edges present in an image. It is surprising how discriminative the features are, considering the small number presented. To determine the optimal features, Ke et al. employed a precision-recall curve which, given their goal of image retrieval, is reasonable.

The next paper, *Studying Aesthetics in Photographic Images Using a Computational Approach* [6] , published by Datta et al. in 2006, focused on using Support Vector Machine (SVM) based classification. In addition to introducing more novel features than Ke et al., it proposed a feature selection technique by simply using SVMs to determine a feature's discriminating power. Subsequently, Datta published a paper [7] that further explored techniques to analyze his original set of features. He confirmed previous results that advanced types of regression are powerful techniques in extracting information for the image retrieval problem.

Regardless of the various classification or regression techniques applied to them, the set of features is what remains vital to any algorithm's success. This encouraged me to focus on improving the set of features which is clearly the current limiting factor.

3 Features

Features are at the heart of any machine learning algorithm, and are the key to its success. Because aesthetic based image rating is a fairly unexplored area I take care to try as many avenues as possible. Due to the fact that I have a large number of features, and these features need to be computed for a vast quantity of images, I chose to cut down on computation time by recursively halving both dimensions in the input images until the largest side was smaller than 1024. This process is reasonable, since most images are similarly resized for viewing on a monitor.

3.1 Dimension Features

Figure 2: Image Dimensions

3.1.1 X dimension and Y dimension

The x and y dimensions of an image are features f_0 and f_1. They determine if the original size of the image has any correlation to a user's ratings. The reasoning is that if the original image is large, then the resized image will

perhaps have a shallower depth of field (DOF). This correlation is true if the original image you are discussing is a physical negative and the resized image is a fixed-size print. An 8x10 image, printed from a 4x5 negative, tends to have a shallower DOF than an 8x10 image printed from a 35mm negative, simply due to the physics of the systems.

3.1.2 Aspect Ratio

Feature f_2 is the same as Datta's [6]. I made certain to include it due to Datta's observation that aspect ratio is highly correlated with a users rating :

$$f_2 = \frac{X_{dim}}{Y_{dim}} \tag{1}$$

3.1.3 Largest Side Aspect Ratio

Building on Datta's observation, I wanted to determine if perhaps the shape of the image mattered more than orientation. Therefore, the next feature is defined as:

$$f_3 = \frac{L_{dim}}{S_{dim}} \tag{2}$$

where L_{dim} is simply the larger of the dimensions, and S_{dim} is the smaller of the dimensions.

3.1.4 Heuristically filtered largest side aspect ratio

Feature f_4 is based solely on my observation that people tend to prefer square and panoramic images, but dislike awkward ratios in between as well as extreme panoramic images. I generated this feature by going through a small set of images and empirically determining crops that appeared to be transition points from good to bad crops. I viewed the aspect ratios of these images and subsequently created a filter. The response of that filter is plotted in Figure 3.

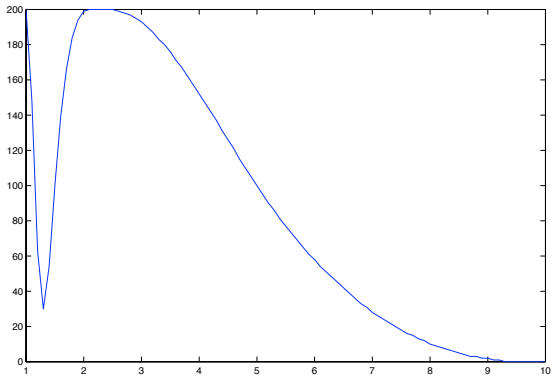


Figure 3: Heuristic Largest Side Aspect Ratio Filter

3.2 Point Operation Features

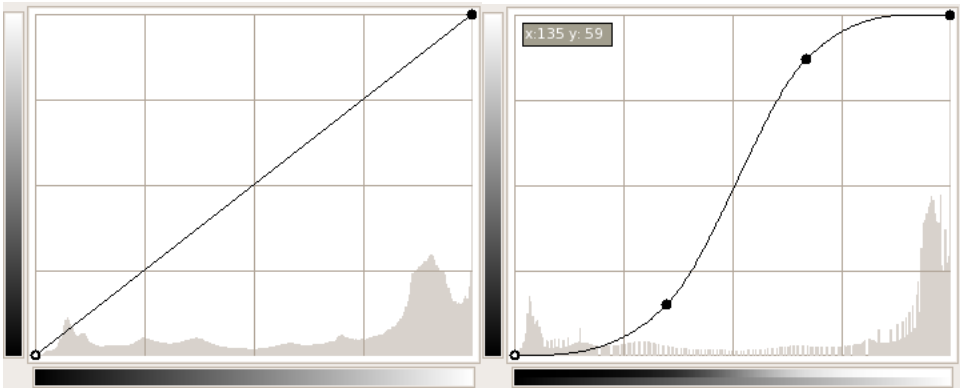
3.2.1 Lab tests

Features f_5 - f_7 are simply the means of the channels from the Lab space, respectively. The L channel is lightness while the a and b channel are color opponent dimensions. The Lab space is attractive because it is perceptually uniform, meaning that a single change in value corresponds to a single increase in visual importance. Transforming the image from Figure 2 into Lab space results in Figure 4.

(a) Lightness (b) a Channel (c) b Channel

Figure 4: Lab Space

3.2.2 Colloquial Contrast



(a) Original Histogram (b) Histogram after S Curve Mapping

Figure 5: S Curve Mapping of Histogram

By colloquial contras, I refer to the distribution of luminosity levels in an image. An image is contrasty if there are few pixels with mid valued levels and more grouped at the black and white levels. This can be confirmed by noting that a remapping of the luminosity values in an image with an S curve is well known to increase contrast and produces the effect of grouping luminosity values towards the extremes. This effect is displayed in Figure 5. To create this feature, I compute the distance between the top and bottom halves of the histogram, Equation 3.

$$d = \frac{1}{.5 \cdot max}\left(\sum_{k=0}^{.5 \cdot max} H_I(k) - \sum_{k=.5 \cdot max}^{max} H_I(k)\right) \tag{3}$$

where d is the distance, max is the maximum number of levels from the image, $H_I(k)$ is the value of the histogram of image I at level k

I then filtered this distance with the heuristic filter displayed in Figure 6.

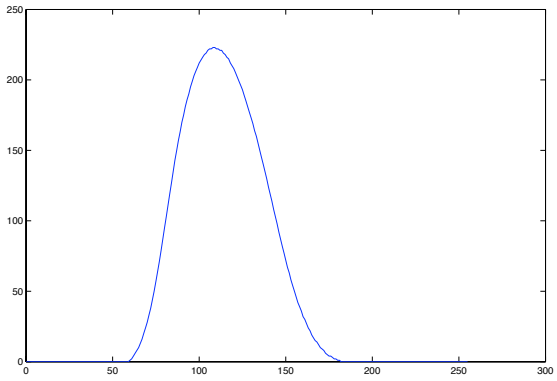


Figure 6: Colloquial Contrast Filter

The function in Figure 6 was generated by taking a small series of images and finding the locations where the image became gray, contrasty, and ideal. The resultant function shows that there exists a "sweet spot" where images have an optimal contrast.

3.2.3 Exposure Ratings

In trying to determine if an image is over- or under-exposed, it is important to determine what the contrast is in the image. The standard deviation is a good measure of contrast. Therefore I made an attempt in each of these novel algorithms to relate the standard deviation to the percentages of the various regions of the histogram. This should help differentiate extreme contrast images that contain significant samples in these regions from images that are closely grouped towards white or black.

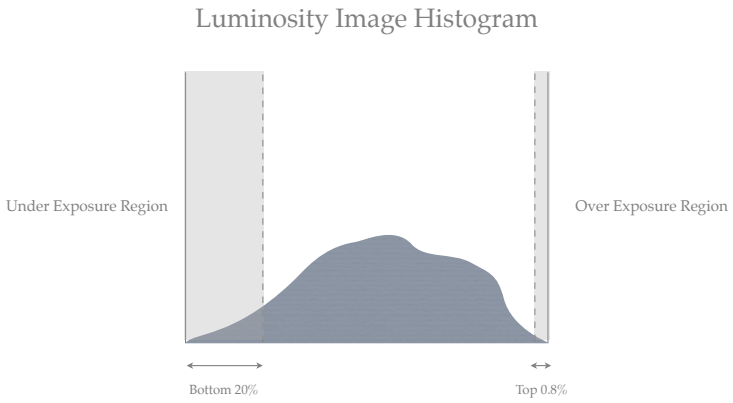


Figure 7: Over and Under Exposure

Over Exposure Over exposure refers to when visually important regions of an image are rendered white and featureless by clipping. It is not currently possible to determine what areas of an image are supposed to be regions of interest. What we *can* say is that a large portion of an image is blown out. The only instance where this is acceptable is silhouettes.

This algorithm uses the top 0.8% of the value image histogram because clipping's effect is relegated to the levels which correspond to white and nothing else.

More explicitly:

$$f_8 = \sum_{k=0}^{.008 \cdot max} H_I(k) - \sqrt{\frac{1}{max} \sum_{k=0}^{max} (H_I(k) - \bar{H}_I)^2} \tag{4}$$

where f_8 refers to the over exposure feature, max is the number of levels -1 in the histogram, $H_I(k)$ is the number of pixels of Image I with level k, $\bar{H}_I$ is the mean of the histogram.

Under Exposure Under exposure describes an underutilization of the dynamic range of a camera. It tends to result in the majority of the values in the image's histogram residing in the lower half of the histogram.

$$f_9 = \sum_{k=0}^{.2 \cdot max} H_I(k) - \sqrt{\frac{1}{max} \sum_{k=0}^{max} (H_I(k) - \bar{H}_I)^2} \tag{5}$$

This algorithm is identical to the over exposure algorithm except that it takes into account the first 20% of the image. Under exposure tends to be simply an underutilization of the dynamic range of the camera and results in the majority of the values being located in the lower valued bins.

3.2.4 Average RGB HSV Tests

(a) Hue (b) Saturation (c) Value

Figure 8: HSV Space

Transforming an RGB pixel into HSV (Hue, Saturation, Value) space provides a more intuitive sense of information at that pixel. Transforming the image from Figure 2 into HSV space results in Figure 8.

The first features simply average the RGB of every pixel in an image and then convert this pixel into HSV. Then feature f_9 - f_{10} are the H and S channel from that pixel, respectively. The Value channel is not a feature here because there is an equivalent feature in Section 3.2.6.

3.2.5 Hue Tests

Color has a very influential effect on humans. Psychological tests [17] have suggested that certain colors have specific mood altering effects. Other tests tend to display that people have a propensity towards blue and green over other colors. Due to this evidence, I focused on the H channel and subjected it to a series of tests to see how closely the color in this image lies to colors that people tend to enjoy.

I applied the following tests to the Hue from the previous section. The first test, feature f_{11} , simply finds the distance from the Hue to blue (hue of 240). The second test, feature f_{12} , is the distance from the Hue to green (hue of 140) as the target hue. The final test, feature f_{13} , is the minimum of the previous two tests.

3.2.6 Fixed HSV Tests

Figure 9: Highlight of Abnormalities in HSV Space

Ke et al. [11] touched on the fact that unintuitive results arise from certain channels of the HSV image when the value channel and the saturation channel are not within a given range. If we look at Figure 9 we see that in the shadows and the highlights there are Hue values and saturation values that are nonsensical. Notice as the shirt gets darker, the saturation becomes greater! Equally disturbing is that as the sky transitions to being clipped, the hue changes drastically. The average for each channel is altered to account for these abnormalities. Those averages are computed by Algorithm 1 and correspond to features f_{14} - f_{16} .

Algorithm 1 Fixed HSV Algorithm:

Value always added to the running average.
if *Value*≥ 0.35 &&*Value* $\leq .95$ **then**
  Saturation of current pixel added to running average.
  if *Saturation* ≥ 0.2 **then**
    Hue of current pixel added to running average.
  end if
end if

Feature f_{17} simply uses a heuristic function that takes feature f_{15} as input. Like previous filters, I generated it by looking at a small set of images. I adjusted a saturation slider in an image-editing application until I hit obvious thresholds separating good regions and bad regions. These thresholds became the inflection points in the function. The function displays that black and white images and well-saturated images in a certain range are

favored, but images that are under-saturated or over-saturated tend to look poor. The function shown in Figure 10 reflects these general trends.

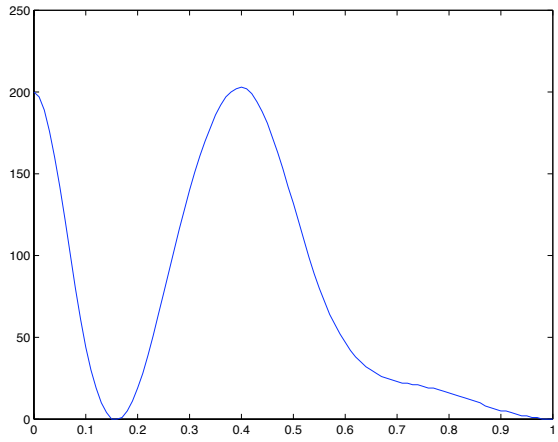


Figure 10: Saturation Filter

I use the same set of hue tests from Section 3.2.5 run for feature f_{14} generate features f_{18} - f_{20} .

3.2.7 Most Common Hue

Another feature that has not previously appeared in AIR literature, feature f_{21} , is the most common hue (MCH). In an attempt to find a more meaningful feature than simply a mean, I located the hue that occurs most in the image, visually represented in Figure 11. To achieve this I took the hue histogram and computed its derivative. Then, I found the maximum Hue from the places where the derivative is either zero or changes from positive to negative. This is the MCH.

This most common hue is then subjected to the same battery of tests in Section 3.2.5, which correspond to feature f_{22} - f_{24} .

3.2.8 Complementary Color Hue Features

It is common knowledge in the art community that using a complementary color scheme makes a piece more aesthetically pleasing. Datta et al.[6] created features that tried to measure complementary color using the average

color of various segments. I do not feel this properly deals with the corner case of the subject being one color and the background being a complementary color, therefore, I chose to use a different approach. This feature uses the MCH, feature f_{21}, and looks for the second most common hue (SMCH). The second most common hue is defined as the hue that has the most occurrences and lies more than 20% of the total hue space away from the MCH. Taking the normalized distance between these two hues results in my complementary color hue feature, f_{25} . Figure 11 displays the MCH and SMCH concept and the distance between the two as I have described.

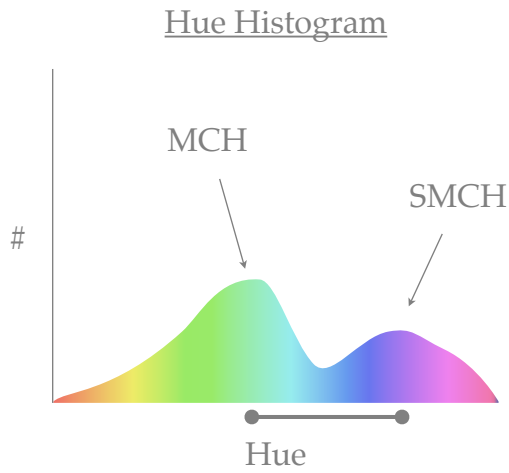


Figure 11: MCH and NMCH

3.2.9 Hue Count

Feature f_{26} is the "Hue Count" technique from Ke et al. paper [11]. The algorithm involves simply creating and arbitrarily thresholding a 20 bin Hue histogram. Then, Equation 6

$$N = \{i \mid H(i) > \alpha m\} \tag{6}$$

is computed. Where m is the maximum value and α is a variable to reduce noise sensitivity. Ke suggests 0.05 as a good value for α. $20 - N$ is the Hue Count. According to Ke et al., a smaller N correlates to higher rated images because professional images are compositionally simpler than snapshots.

3.3 Spacial Features

3.3.1 Fourier Space tests

For these tests, the magnitude of the DFT applied to the the luminosity image is computed. Then the bottom half is removed, as is displayed in Figure 12.

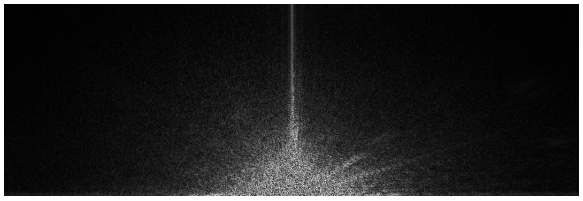


Figure 12: Top Half of DFT Magnitude

Angle frequency tests One property of the (DC centered) DFT is that the angle from the center of the image to a frequency gives the angular propagation of that frequency. These features attempt to determine if the human visual system sees more frequencies with a given angle, relative to all others, as aesthetically pleasing. Put another way: are images with more horizontal content than vertical content pleasing?

This novel metric requires taking the top half of the magnitude of the DFT and dividing it into six angular sections, as is shown in Figure 13. Features f_{27} - f_{32} are the ratios of the sums of the magnitudes in each individual region compared to the sum of all magnitudes.

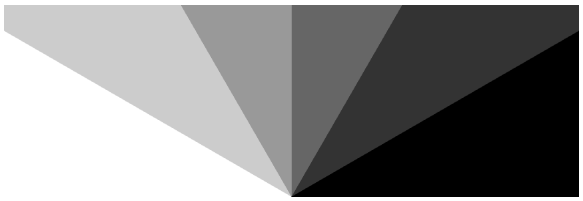


Figure 13: Angular Regions of the DFT magnitude Image

Frequency range cutoff The next metric I invented took various ranges of frequencies from the DFT. For example, the ratio obtained from comparing the zeroeth frequency region to the whole image's power tells you if the majority of the image content is low frequency. Low frequency content just refers to largest scale changes such as a gradient across an image.

The top half of the magnitude of the DFT is taken and divided into four radial sections, as is shown in Figure 14 . Features f_{33} - f_{36} are the ratios of the sums of the magnitudes in each individual region compared to the sum of all magnitudes.

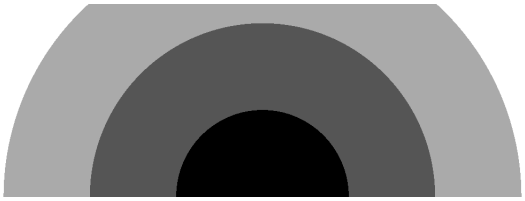


Figure 14: Frequency Cutoffs for the DFT magnitude Image

Mean Feature f_{37} is the mean of the magnitude of the DFT. It is simply a measure of the power in the image.

The above Fourier tests were similarly run on the B channel image to produce features f_{38} - f_{48} .

Ke Blur Analysis Feature f_{49} is another from Ke et al. [11]. It takes the magnitude of the DFT and, using a threshold of five, obtains a binary image. The mean of this binary image is the Ke Blur feature. Ke claimed that this, combined with the Tong et al. algorithm [14], was his most discriminating feature. I chose not to use the Tong et al. algorithm because I could not obtain the binaries, as Ke et al.[11] did, and the paper did not explain the algorithm in sufficient detail (or appropriate terminology) to re-implement it.

3.3.2 Ratio of edges at borders versus total edges

Ke et al. claimed that the distribution of the edges was the second most effective feature in differentiating low-rated images from high-rated ones. I chose not to implement Ke et al'.s algorithm because it involved two image masks that I did not have access to. Also, the two masks were obtained from summing up the images in two classes, making the metric somewhat suspect. Essentially, using their method seemed to tune the feature to the particular dataset they were using.

My feature f_{50} essentially measures the same effect. The algorithm takes the original image and transforms it into luminosity space. Then, I apply a 1 sigma, first order, Canny-Deriche filter on the image, see Figure 15. This

edge image is squared to obtain absolute changes. Then, all the pixels within a border, the width being 10% of the smallest dimension, are summed and the ratio of the pixels within the border to all the other pixels in the edge image is computed, shown in Figure 16. The result of the ratio for the Value space image generates feature f_{51} .

Figure 15: Luminosity Edge Image

Figure 16: Border of Luminosity Edges Image

3.3.3 Wavelet Tests

All wavelet analysis uses a three level Daubechies wavelet transform, as described in Datta et al. [6]. A three level Daubechies wavelet transform applied to the image in Figure 2 produces Figure 17. A level consists of three images. The three images, LH, HL, and HH, correspond to an image that has had a highpass or lowpass filter applied to a given direction. LH has had a lowpass filter applied to it in the horizontal direction and a highpass filter applied in the vertical direction. Running the image through the filter bank is recursively applied three times to produce Figure 17. Level one corresponds to the content with the smallest spatial variation, whereas

level three corresponds to image content with the largest spatial variation; therefore noise is commonly captured by the first level.

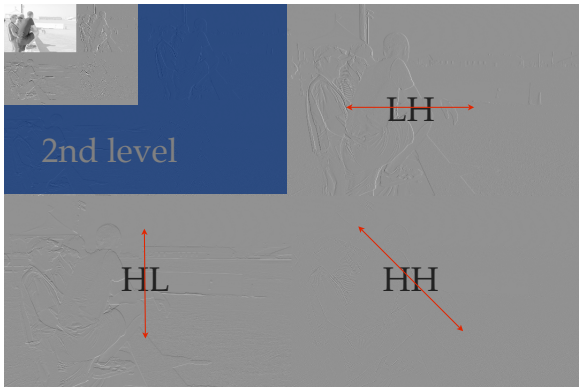


Figure 17: 3 Level Daubechies Wavelet Transform of Image in Figure 2

Dattas wavelet level analysis Datta et al [6] introduced the concept of taking the ratio of the sum of each wavelet level versus the sum from all levels, see Equation 7. Features f_{52} - f_{60} are this analysis for each channel in Lab space, where the levels are in ascending order. Similarly, the features f_{61} - f_{69} are for the channels in the HSV space.

Absolute means I wanted to see if the average of each wavelet level had more importance than the relative power of a level with the others. Thus, features f_{70} - f_{78} are the mean of each level for Lab space and features f_{79} - f_{87} are the same analysis for the HSV space.

Ratio of wavelets in Luminocity at borders versus total In light of the discovery by Ke et al., that the edge distribution is important and given the wavelet levels measure edges in a weak sense, I chose to test if comparing the sum of the power at the border of each wavelet level to the total power was a discriminate a feature. I used the method of determining the border as in Section 3.3.2. Features f_{88} - f_{90} are the ratios for each level

3.3.4 DOF measures

These features are taken from Datta et al. [6]; they are a comparison of the sum of the central region of the third wavelet level versus the sum of all the

pixels in this level. Figure 18 displays the division scheme and Equation 7 explicitly describes this feature for a single level.

$$f_x = \frac{\sum_{(x,y)\in M_6 \cup M_7 \cup M_{10} \cup M_{11}} w_l(x,y)}{\sum_{i=0}^{16} \sum_{(x,y)\in M_i} w_l(x,y)} \tag{7}$$

where {M1, ..., M16} are the 16 equal rectangular blocks, numbered in row-major order, created by the division scheme; w_l is $\{w_l^{hh}, w_l^{lh}, w_l^{hl}\}$ for a given wavelet level l.

I chose to apply this same analysis to each level. Features f_{91} - f_{99} correspond to doing this analysis for each level and every channel in the Lab Space and features f_{100} - f_{108} are similarly defined for the HSV space.

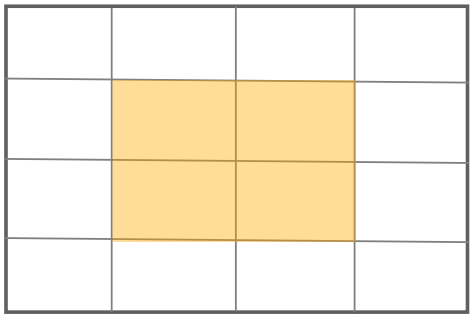


Figure 18: Depth of Field Sectioning

3.3.5 Novel ROT/ROH tests

Ratio of one region of image verses others

I have a novel interpretation about how to determine one aspect of the rule of thirds that I have not seen discussed or presented previously. Most images that follow the rule of thirds seem to follow two sets of rules. First is the rule that states that the center of the subject matter should lie near the third regions intersection points, see Figure 19. The second rule is that one should divide regions on one of the third region division lines. I interpret this to mean third regions to help divide up an image in terms of color, texture, luminosity, etc.

(a) Rule of Thirds Divisions (b) Rule of Thirds Applied

Figure 19: Rule of Thirds

Thus, I devised a set of tests that I feel accurately assess the first and second aspect of the rule of thirds. My feature for the first test involves simply summing up the pixels of the image weighted by the inverse of the distance from one of the rule of thirds points, see Equation 8.

$$S_{p_{(k,l)}} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \frac{1}{\sqrt{(n-k)^2 + (m-l)^2 + 1}} I(n,m) \tag{8}$$

$$S_{p_{(k,l)}} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \frac{1}{(n-k)^2 + (m-l)^2 + 1} I(n,m) \tag{9}$$

where $S_{p_{(k,l)}}$ is the sum of all the weighted pixels relative to point p, p is an intersection point at coordinate (k,l) and $I(n,m)$ is the value of image I at point (n,m), N is the x dimension of the image, and M is the y dimension of the image.

The sums from each of the four rule of thirds intersection points discussed above are features f_{109} - f_{112} for the luminosity edges image. The sums from all the intersection points is feature f_{113}. I also weight the pixels by the inverse of the radius squared, Equation 9, to generate features f_{114} - f_{118}. This whole procedure is repeated for the value edges image to produce features f_{119} - f_{128}.

For the second rule of thirds test, I simply take the average of one of the third regions for a given image and compare it to the combination of the other two regions. Figure 20 gives a conceptual sense of this and Equation 10 formally describes how I compute the ratios.

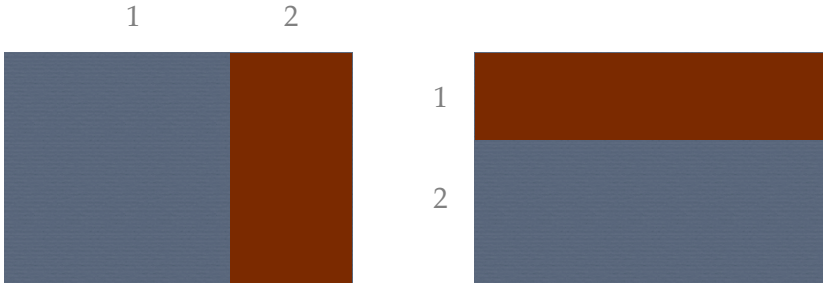


Figure 20: Rule of Thirds Sectioning

$$f_x = \frac{(\sigma_1 - \sigma_2)}{(\sigma_1 + \sigma_2)} \tag{10}$$

where f_x is the feature for the current third region, σ_1 is the sum of all third regions other than the current one and σ_2 is the sum of the current third region.

These ratios produce features f_{129} - f_{131} for the luminosity edge image, f_{132} - f_{134} for the value edge image, f_{135} - f_{143} for the set of channels from the fixed HSV space, and f_{144} - f_{152} for the set of channels from the Lab space.

Santella et al. [12] discuss preferring crops created using a rule of halves procedure over crops created using the rule of thirds. I concede that images using the rule of halves might be more aesthetically pleasing than rule of thirds images. Therefore, I repeated all the above tests using two divisions of the image rather than three. This generated features f_{153} - f_{173}.

3.4 Saliency Map Features

The saliency map, developed by Itti et al. [10], can be interpreted as an image where a larger pixel intensity corresponds to image content that is most interesting to the human eye. The saliency map for Figure 2 is shown in Figure 21 The saliency map was utilized in *Gaze-based interaction for semi-automatic photo cropping* [12] for determining an optimal composition for the crop of an image.

Figure 21: Saliency Map

3.4.1 Ratio of saliency from ROT/ROH locations

By repeating my analysis from Section 3.3.5 for the saliency map, I generate features f_{174} - f_{200} .

3.4.2 Ratio of saliency at borders versus total saliency

I applied the same analysis from Section 3.3.2 to the saliency map. The reasoning being that both salient content and edges near the borders are just as distracting. Then, the ratio is computed between the saliency within that border and the total amount of saliency in the image. This is feature f_{201}.

3.4.3 Saliency Center of Mass

A common concept in graphic design is to reduce tension in an image by keeping it balanced. Based on this concept that the salient content within an image should be balanced, I postulate that images with a center of mass near the center of the image will be more aesthetically pleasing. The x and y location of the center of mass for all the saliency in the image, computed by Equation 11, are features f_{202} and f_{203}.

$$\begin{aligned} I_{COM_x} &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} n \times I(n,m) \\ I_{COM_y} &= \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} m \times I(n,m) \end{aligned} \tag{11}$$

where I_{COM_x} is the center of mass for the x dimension, N is the size of the X dimension of the image, M is the size of the Y dimension of the image, and $I(n, m)$ is the value of the pixel in image I at point (n, m)

3.4.4 Saliency Map Segments

In the same manner described in Santella et al., [12] I created segments by thresholding the saliency map based on a percentage of the maximum salience. The threshold I chose was 25% of the maximum saliency. Empirically, this yielded appropriately sized segments. These segments were then labeled, which created an image like the one in Figure 22. I subsequently analyze the segmented image using the techniques listed below.

Figure 22: Saliency Map Segments

I computed the center of mass for the largest contiguous segment using the same approach as Equation 11, but applying it to a binary image where the pixels from the largest region are assigned 1 and all other pixels are 0. These normalized coordinates, x and y, are features f_{204} and f_{205} .

Datta et al. [6] also performed segmentation, but used a different approach. One way they analyzed their largest segment was by locating which quadrant it is located in. I copied this concept for feature f_{206}, which is the quadrant where the largest saliency segment's centroid lies.

Closeness to ROT/ROH locations For feature f_{207} I follow in the footsteps of *Gaze-based interaction for semi-automatic photo cropping* [12]. In this paper, Santella et al. attempted to determine the best way to perform automatic cropping. I utilized an algorithm that is similar to the one they

introduced. It involves finding the centroid of all the regions and calculating each region's distance to the closest rule of thirds intersection point. The sum of all these distances is feature f_{207}.

3.5 Frameworks Employed

Due to the complex nature of some of the image decomposition techniques various software packages were used to aid in the overall processing of images. CImg [15] was the framework that provided the most support for implementation of the various algorithms. Waili [16] was necessary for performing the three level Daubechies discrete wavelet transform. Finally, the iLab Neuromorphic Vision Toolkit [1] was used to create saliency maps.

4 Methodology

4.1 Datasets

I chose to use two datasets in my study. The first dataset is from the 2006 Datta et al. publication[6]. My set consists of a 3,181 image subset from Dattas original set of photographs taken from the photo.net photo rating community. However, there is a major flaw in this dataset; there are a large quantity of images with a small number of ratings. This flaw is apparent from the graph Datta published shown in Figure 23. Datta et al. highlighted the seriousness of this issue by plotting the threshold of minimum number of ratings required per image versus the amount of classification improvement in his algorithm.

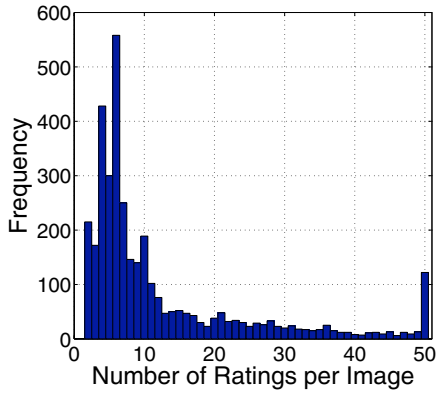


Figure 23: photo.net Dataset's Number of Ratings per Image [7]

The second dataset I utilized was introduced in the paper by Ke et al.[11]. It is a substantially larger dataset, consisting of 12,116 photos, and was taken from the dpchallenge.com community. In addition, the number of ratings per photograph in this dataset are significantly higher on average, obvious from Figure 24.

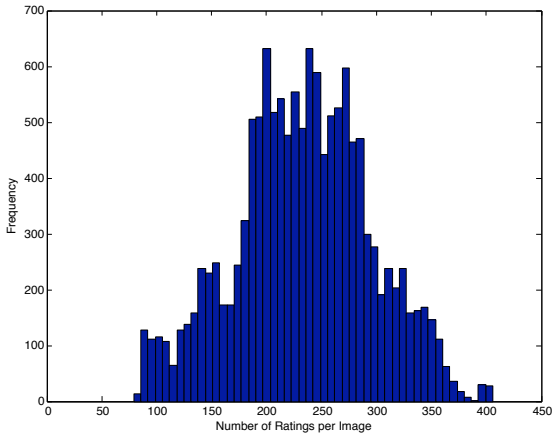


Figure 24: dpchallenge.com Dataset's Number of Ratings per Image

Finally, it is important to note that the datasets I discuss above are a subset of the originals. This occurred because the researchers were unwilling to provide the images, for fear of copyright infringement. I crawled the

various sites to collect these images; many, however, had since been removed.

4.2 Feature Selection

The feature selection technique I chose to use is a variant of the F-Score + SVM technique that was created by Chen and Lin[4]. F-score is a simple measure of the discrimination of two sets of real numbers. Equation 12 shows the f-score of the ith feature, where the number of negative and positive instances are n_- and n_+ and the training vectors are x_n for $n = 1, 2, ..., l$.

$$F(i) \equiv \frac{(\bar{x_i}^{(+)} - \bar{x_i})^2 + (\bar{x_i}^{(-)} - \bar{x_i})^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x_i}^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x_i}^{(-)})^2} \tag{12}$$

here the average of the ith feature of the whole, positive and negative data sets are $\bar{x_i}$, $\bar{x_i}^{(+)}$, $\bar{x_i}^{(-)}$, respectively. The ith feature of the kth positive and negative instance are $x_{k,i}^{(+)}$ and $x_{k,i}^{(-)}$, respectively. In Equation 12 the numerator indicates the discrimination between the positive and negative sets, while the denominator indicates the discrimination within each set. The larger the F-score is, the more likely this feature is to have a greater discriminative ability. [4]

This metric is used to test each feature and allows us to gain some sense of the order of discrimination ability of the individual features. The features are sorted by their F-Score and placed into a list. Recursively, the top half of this list has its classification accuracy tested using SVM until the list becomes smaller than two. The SVM classification is tested using a five fold cross validation through LibSVM [3]. The set with the best SVM classification is selected as optimal. From my initial tests, this method showed similar results to the brute force method, introduced by Datta [6] of acquiring the SVM classification accuracy of each individual feature. I did, however, choose to obtain the individual accuracies for the top fifteen highest f-scoring features, attempting to get a more reliable measure of their relative abilities.

The above technique requires binary classified data but the data has a continuous range of ratings; therefore, I must threshold the data into two classes so that I can apply the feature selection technique. For the photo.net data, I threshold it in the same manner Datta et al.[6] discussed, by setting the lower threshold at 4.2 and the upper threshold at 5.8. I chose to threshold the dpchallenge data at 5, since Ke et al. already separated the data into two groups; those groups being the top and bottom 10%.

I validate the optimal feature set by computing an ROC curve on the binary classified data. The ROC is a powerful tool which allows one to discern how well the optimal feature set performs regardless of which classification technique is applied to it. It also allows us to compute the area under the curve (AUC), which represents how well the optimal feature set separates the two classes. Basically, the AUC for the ROC is a more universally comparable metric than the output of my SVM classifier.

4.3 Regression

After finding the optimal feature set using classification, I wanted to obtain the goal of this paper: to acquire an Aesthetic Image Rater. To obtain my AIR function, I used regression. The data is prepared for regression by normalizing the ratings from both datasets to ensure that they ranged from 0 to 1, rather than 0 to 7 or 0 to 10, then adjusting the means of the datasets to be 0.5. The specific regression technique I chose to employ is nu-SVR (support vector regression).

5 Results

5.1 Feature Selection

From the binary classification technique discussed in Section 4.2, I determined the optimal feature set, for both datasets, to be a 63 feature set. This feature set resulted in a 74.96% classification accuracy for the photo.net dataset and a 75.32% classification accuracy for the dpChallenge.com dataset. These two accuracies are higher than the ones published by Datta et al. or Ke et al., which led me to believe that my feature set is better; but, because I am using subsets of of the original datasets, it is impossible to claim this with certainty. Also, trying to compare my two classification accuracies is unproductive since the thresholds are substantially different. I can, however, judge the relative performance of the individual features, thus allowing me to compare my novel features with those that were previously introduced. The top 15 features and their individual accuracies for the dpchallenge.com dataset and the photo.net dataset are shown in Figure 25.

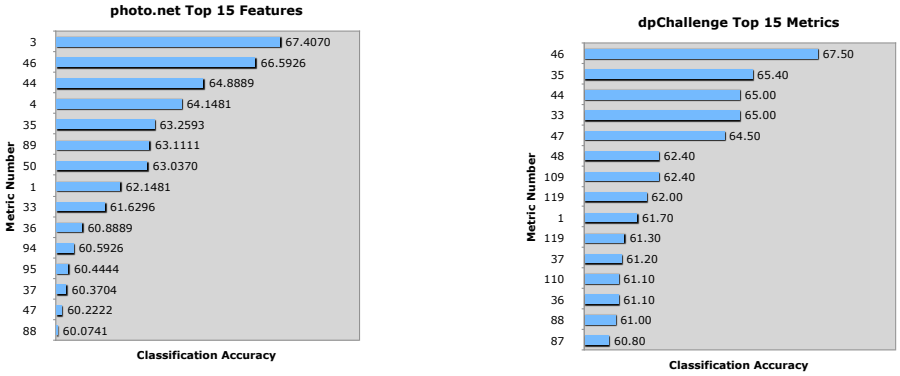
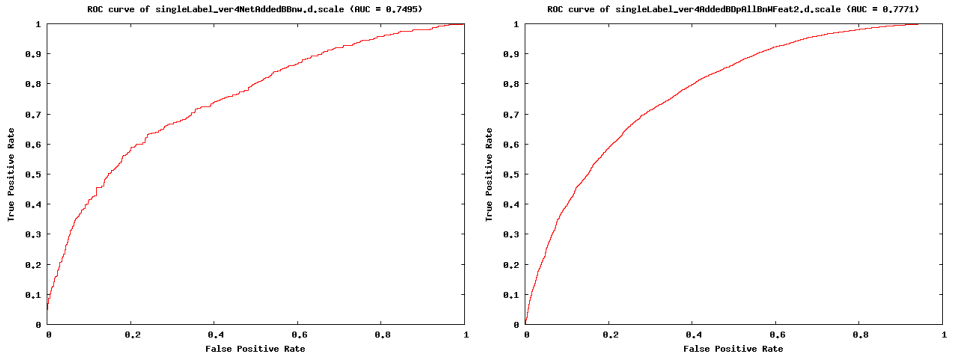


Figure 25: The Top 15 Ratings for Each dataset

It is worth noting that when we combine the two sets of features the top five features from this set $\{f_{46}, f_{35}, f_{44}, f_1, f_{47}\}$ are all features that I created.

The ROC curves for the two datasets are shown in Figure 26. The AUC value is 74.95% for the photo.net curve and 77.71% for the dpchallenge.com curve. These curves confirm that the chosen set of features have excellent performance.



(a) photo.net ROC Curve (b) dpchallenge.com ROC Curve

Figure 26: ROC Curves

5.2 Regression

After performing nu-SVR on the normalized datasets, I found the mean squared error (MSE) to be 0.0113 for the photo.net dataset and 0.0127 for the dpChallenge.com dataset. Surprisingly, combining the two datasets results in a smaller MSE of 0.0109. To generate a more intuitive result I rescale the normalized data to a rating range from 1 to 5, the standard range for photo management software, and on average I was only off by .52 of a rating point. This error is very reasonable and makes it possible to say with certainty that I have discovered an efficient set of features for photo management software users and more generally to achieve my goal of creating an AIR algorithm .

6 Discussion and Future Work

During the development of this AIR algorithm, I added to the set of known features that strongly correlate to human rating of images. I was able to unify previous features that were claimed to perform well and attest to their relative performance with my features. My features outperformed all the features I was able to recreate from previous papers. More importantly than individual feature performance my feature set shows excellent performance at the task of creating absolute image ratings on par with a human's.

Regardless of my improvements, more efficient features need to be created to ensure that the ratings assigned to images are reliable. To compare the performance of the new features, a standard set of photographs is needed. Without this standard set of images, papers will continue to be published that contain useless performance measures.

I believe that if even a small amount of more efficient features are discovered, this algorithm will be reliable enough to be included in every piece of photo management software; and, as camera's processors become more powerful, soon be included directly in digital cameras as a source of immediate feedback.

Including the algorithm in cameras is a step in the right direction for helping users improve their images. However, with all the information one is able to gather in this process, it seems as if more than just a rating should be relayed to the end user. A concept I had, which appears to be novel, is to attempt to expose the internals of the multidimensional space to the user. Exposing more information to the user would allow her feedback based on all the criteria the algorithm is using; thus allowing her to learn quickly and adjust her photographs accordingly. This can be accomplished

by providing the end user with an approach on how to adjust her image to direct it towards a higher rating area of the multidimensional space. I do not know of an optimal technique to achieve this end. Although, it appears that there are papers, such as *Discriminative Direction for Kernel Classifiers* [8], that offer a promising approach. Aiding humans is really the goal of AIR algorithm, and, turning this algorithm into an informative guide rather than a simple judge, would surely solidify its use in all image related applications.

7 Acknowledgements

I would like to thank everyone who helped me to create this thesis. Dr. Alan Bovik, for being my advisor. Dr. Alan Cline, for being a surrogate advisor and friend; I especially appreciate him talking to me about being scooped. Elana Clift, for editing my backwards way of writing. Yonatan Bisk for applying various neural network techniques to my data to help obtain additional results (even if they didn't make it into this thesis). Dr. Kristen Grauman and Dr. Raymond Mooney, for discussing various aspects of my thesis with me and pointing me in fruitful directions. David Carr, Mickey Ristroph, Tarun Nimmagadda, Gilbert Bernstein, Justin Hilburn, Mark Reitblatt, and Jeremy Powell for allowing me to bounce ideas off of them. Rob Peters, for helping me understand the intricacies iNVT toolkit.

I would also like to thank everyone who didn't have a direct affect on my thesis, but had a direct affect on me. My parents for being infinitely supportive and understanding. Mr. and Mrs. Chappell for being supportive friends. Dr. Don Winget (and everyone from his family and his lab), for providing me with the opportunity to work on a telescope and making me a co-author on papers while I was still a high school student. The Bibble Labs staff for being supportive and helpful while I was a intern there. I apologize if I am forgetting certain individuals. I hope everyone understands that your support has made me who I am today and has made this thesis possible. I appreciate everything that you have done for me.

References

[1] *The iLab Neuromorphic Vision C++ Toolkit.*

[2] Rudolf Arnheim. *Art and Visual Perception: A Psychology of the Creative Eye.* University of California Press, 1974.

[3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] Yi-Wei Chen and Chih-Jen Lin. *Combining SVMs with Various Feature Selection Strategies*, 2005.

[5] Corbis. *Corbis Corporate Fact Sheet*.

[6] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Studying aesthetics in photographic images using a computational approach. pages III: 288–301, 2006.

[7] Ritendra Datta, Jia Li, and James Z. Wang. Learning the consensus on visual quality for next-generation image management. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 533–536, New York, NY, USA, 2007. ACM.

[8] P. Golland. Discriminative direction for kernel classifiers, 2001.

[9] The Guardian. *Together in electric dreams*, January 2005.

[10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.

[11] 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, 2006.

[12] Anthony Santella, Maneesh Agrawala, Doug DeCarlo, David Salesin, and Michael Cohen. Gaze-based interaction for semi-automatic photo cropping. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 771–780, New York, NY, USA, 2006. ACM.

[13] Hanghang Tong, Mingjing Li, Hong-Jiang Zhan, Jingrui He, and Changshui Zhang. Classification of digital photos taken by photographers or home users. In *Advances in Multimedia Information Processing - PCM 2004*, volume 3331/2005. Springer Berlin / Heidelberg, 2004.

[14] Hanghang Tong, Mingjing Li, Hongjiang Zhang, and Changshui Zhang. Blur detection for digital images using wavelet transform. *Multimedia*

and Expo, 2004. ICME '04. 2004 IEEE International Conference on, 1:17–20 Vol.1, 27-30 June 2004.

[15] David Tschumperlé. *The CImg Library : C++ Template Image Processing Library.*

[16] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. WAILI: A software library for image processing using integer wavelet transforms. In K.M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338, pages 1490–1501. International Society for Optical Engineering, 1998.

[17] T. W. Whitfield and T. J. Wiltshire. Color psychology: a critical review. *Genetic, Social, and General Psycology Monographs*, 1990.