LEARNING IMAGE SALIENCY FROM HUMAN TOUCH BEHAVIORS

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree

Master of SCIENCE

by

Shaomin Fang, B.S.

San Marcos, Texas
August 2013

LEARNING IMAGE SALIENCY FROM HUMAN TOUCH BEHAVIORS

Committee Members Approved:

Yijuan Lu, Chair

Dan Tamir

Ziliang Zong

Approved:

J. Michael Willoughby
Dean of the Graduate College

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

Dedicated to my husband, Ribel Fares, who has been a constant source of support and encouragement during the challenges of graduate school and life. This work is also dedicated to my parents, who have always loved me unconditionally and whose good examples have taught me to work hard and smart for the things that I aspire to achieve.

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor Dr. Yijuan Lu, whom to work with has been a great pleasure to me, with valuable guidance, financial support, encouragement, and patience. Without her guidance and persistent help, this thesis would not have been possible.

I am also grateful to my committee members, Dr. Dan Tamir and Dr. Ziliang Zong, for constant support and insightful comments in this thesis.

This manuscript was submitted on June 17, 2013.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

ABSTRACT

LEARNING IMAGE SALIENCY FROM HUMAN TOUCH BEHAVIORS

by

Shaomin Fang, B.S.

Texas State University-San Marcos

August 2013

SUPERVISING PROFESSOR: YIJUAN LU

The concept of touch saliency has recently been introduced as a possible alternative for eye tracking in usability studies. This touch saliency study shows that image saliency maps can be generated based on human simple zoom behavior on touch devices. However, when browsing images on touch screen, users tend to apply a variety of touch behaviors such as pinch zoom, tap, double tap zoom, scroll, *etc.*, in order to look at their regions of interest on images. Several questions naturally draw our attention: Do these different behaviors correspond to different human attentions? Which behaviors are highly correlated with human eye fixation? How to learn a good image saliency map from various/multiple human behaviors? In order to address those open questions, a series of studies are designed and conducted. Two novel and comprehensive touch saliency learning approaches are also proposed to derive good image saliency maps from a variety of human touch behaviors by using different machine learning algorithms. The experimental results demonstrate the validity of our study and the potential and effectiveness of the proposed approaches.

CHAPTER I

INTRODUCTION

Human perceptual systems are only able to process a subset of the visual inputs they receive at one time. Visual attention refers to the selective concentration on meaningful regions of a scene [Marques et al., 2006]. The map to display the spotlights of the concentrations on an image is called image saliency map. Visual attention allows us to select the information that is most relevant to ongoing behaviors. In order to learn visual attention, many researchers present images as stimulus to their subjects [Itti et al., 1998; Judd et al., 2009; Ramanathan et al., 2010; Tsotsos and Bruce, 2006].

Visual attention learning is a thriving area of research. It has been proven useful in a variety of applications. These applications include object detection [Oliva et al., 2003], face detection [Goodrich and Arel, 2012], compression of multimedia data [Yang et al., 2006], image segmentation [Hua et al., 2010], image retargeting [Setlur et al., 2005], information retrieval [Bamidele et al., 2004], *etc.*

In the traditional visual attention study, the user's eye fixation data is required and the eye tracking device is the only equipment to collect the data. Eye trackers use infrared cameras to record and project the pupil movements onto a two-dimensional plane: the display screen. The projection coordinates on the screen are then interpreted as gaze coordinates, in other words, visual attention coordinates.

Although eye tracking has been developed for years and has very useful applications, it is not widely popularized due to four major reasons:

1) The cost is very high. The exorbitant price tag on commercial systems has resulted in limited use of eye-tracking technology.

2) Complicated operation, which requires non-trivial calibration, validation, and chin-and-forehead-rest for stabilization.

3) Specialized knowledge is needed. The users have to be trained to operate it.

4) Low mobility. It is not easy to carry it everywhere due to its considerable size and weight.

Recently, with the popularity of touch phones, touch tablets and touch laptops, more and more people rely on touch devices for daily image or video browsing, sharing, and surfing as these touch screen displays bring applications and entertainment to life with our fingertips.

When using a limited size touch screen for image browsing, users tend to use fingers to tap, pinch zoom, double tap zoom, and scroll to have a closer view of a particular region of interest. These touch behaviors may indicate user's interest and attention on certain regions of the image, and perhaps capture similar information as the eye-fixation data in the visual attention study. Many interesting questions naturally arise:

1) How to learn a good image saliency map from various/multiple human touch behaviors?

2) Do different touch behaviors (tap, pinch zoom, double tap zoom, scroll etc.) correspond to different human attentions? And how to learn such a relationship?

3) Which behaviors are more correlated with human eye fixation?

4) Are there any algorithms that can be applied to answer these questions?

To address these questions, in this work, a series of studies are designed and conducted, with the conventional eye-fixation based saliency data served as the ground truth. In order to collect user's touch behavior data, an image browsing app is designed on Android development platform. Two novel image saliency learning approaches are also proposed to derive a good image saliency map from a variety of human touch behaviors. During the process of building a supervised learning model, the weights of different human touch behaviors are learned, which indicate the different contributions of these behaviors to the image saliency information.

Compared with eye-tracking devices, touch devices have many advantages. They are much more popular, cheaper, and also easier to operate and carry. The user's finger touch behaviors data, which is referred as Touch Saliency [Xu et al., 2012], are much easier to be recorded than eye-movements using an eye tracker. Therefore, touch saliency can be easily obtained and it will definitely have wide applications in various fields where eye tracking or computational visual attention models have been applied, such as image compression, image segmentation, image retargeting, etc., in the near future.

The main contributions introduced in this thesis are summarized as follows:

1) We build a dataset containing 446 images with touch behavior data from 15 observers in an image browsing task on a touch screen phone.

2) A set of valuable features from the touch information related to visual attention is proposed.

3) Visual attention from a variety of touch behaviors is quantitatively studied and analyzed.

4) Two supervised learning methods are proposed to automatically learn the correlation between different touch behaviors and human eye fixations.

5) The learned models derive good image saliency maps from a variety of touch behaviors.

6) This work explicitly guides the research in touch saliency ability estimation and opens broad research possibility for touch behavior based visual attention learning.

CHAPTER II

BACKGROUND

In this chapter, we introduce the detailed background about the popular image saliency map generation methods and background about the Machine Learning methods we use for our proposed image saliency learning approaches.

2.1 Image Saliency

Visual Attention is to learn which elements of a visual scene are likely to attract the attention of human observers. It has been proven very useful in many fields. When people look at images, they selectively concentrate on some meaningful regions of images which attract them. Figure 2.1 gives examples showing some human's visual attention on images.

Image saliency is a term which indicates human's regions of interest on images.

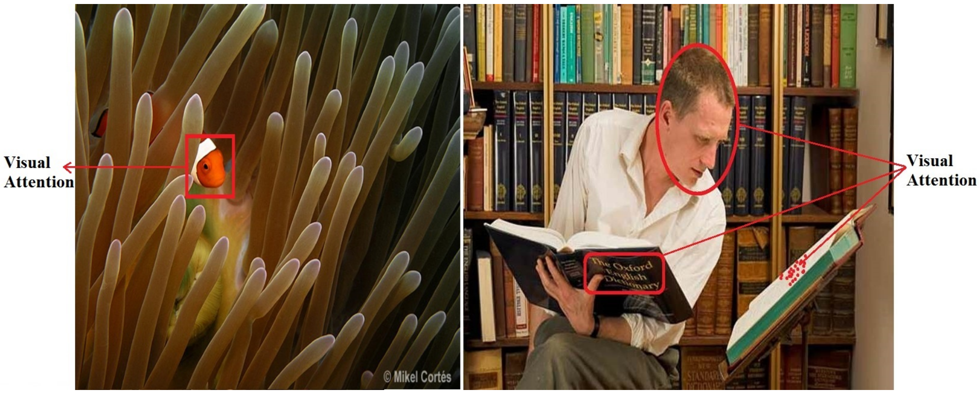


Figure 2.1: Examples of visual attention on images.

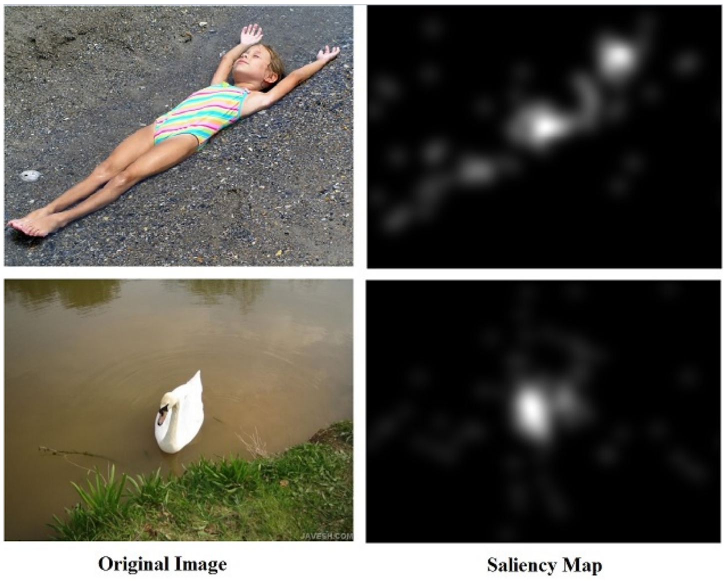


Figure 2.2: Examples of image saliency map.

An image saliency map represents the salient regions on an image. Most commonly, the image saliency map is a gray-scale image, where the pixel values (0-255) indicate the saliency values. The greater the value is, the more salient is, and the brighter the pixel is. Figure 2.2 gives an example of the image saliency maps corresponding to the original images by using eye tracking from NUSEF data set [Ramanathan et al., 2010].

There are many different ways to generate an image saliency map depending on the methods used to learn the image saliency. The following two sections indroduce the most popular two types of methods used to generate the image saliency maps.

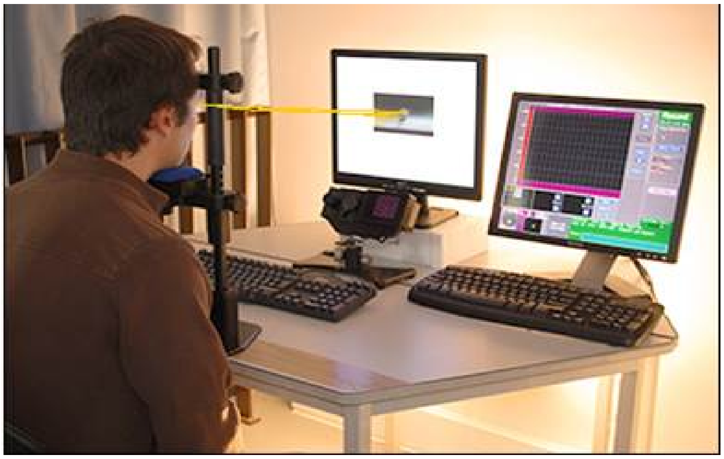


Figure 2.3: Examples of eye tracking.

2.1.1 Image Saliency from Eye Tracking

Eye tracking is a method used to record the gaze coordinates of a user. Eye trackers, which have different ways of operating, are the tools used in the process. Most commonly, an infrared light source and a camera is used in combination in order to capture the reflection the user's eyes.

Eye tracking technology brought about many academic and industrial applications. In usability studies, eye movement analysis has become one of the predominant methodologies. The basic approach is to classify eye movements into fixations (when user focuses with little observed eye movement) and saccades (when user makes a ballistic eye movement between fixations). This classification leads to the interpretation of fixation time as a measure of interest, and total distance of saccades as a measure of effort. Figure 2.3 gives an example of eye tracking [LEAD, 2010]. It shows that a user's eye movements are tracked and recorded by an eye tracker when he is looking at an image on screen.

Eye tracking is used in a wide variety of fields including human-computer interaction, cognitive science, psychology, marketing research and medical research. Eye tracking is being used more and more for web and software usability specifically around usage patterns, online advertising, branding, and navigation usability. These uses of eye tracking have been highly promising for many years.

When human focus on particular region of an image with little observed eye movement above a threshold, the region of the image is considered as their interest or salient region. One method used to calculate the pixel value of the image saliency map is based on how long they fix their eyes on that region. Obviously, the longer they fix on that region, the greater pixel value is on that region.

For example, one simple and popular way [Ouerhani et al., 2003] to generate image saliency maps from eye tracking data collected in the experiments with human subjects is described as follows: First, only fixations with greater than a threshold (for example: threshold =120 ms) are recorded. Second, each fixated location gives rise to a gray-scale (values are between 0-255) patch whose activity is gaussian distributed, the gaussian width should be approximate the size of the fovea and the amplititude is proportional to the fixation duration. Third, the non-fixated locations are considered as background which is represented as black with pixel values of 0.

2.1.2 Image Saliency from Visual Content

Visual content based prediction methods for image saliency learning detect image saliency based on image's visual content, such as content contrast, color, texture, intensity, orientation information, *etc.* A combination of some of these information is filtered from the original image, and then the features of each content information are extracted. With these image visual content features, different computational methods are used to calculate the image's saliency value which represents where human are interested in on the image.

Figure 2.4 [Foulsham, 2008] gives an example of a model of a popular state-of-the-art visual content based method by [Itti et al., 1998]. The input image is first decomposed into a set of topographic feature maps based on the color, intensity, orientations information of the image. These feature maps are then normalized and summed into a final input S to the saliency map:

$$S = \frac{1}{3}(N(\bar{I}) + N(\bar{C}) + N(\bar{O})) \qquad (2.1)$$

Where $N(\bar{I})$, $N(\bar{C})$, and $N(\bar{O})$ are normalized intensity, color, orientation feature maps respectively. In our evaluation experiment for Itti model, the final saliency map S is converted into a grayscale image.

2.2 Background of Machine Learning Methods Used for Proposed Image Saliency Learning Methods

There are two regression machine learning methods used to learn image saliency maps in this thesis: Linear Regression (LR) and Support Vector Regression (SVR). They both try to learn a weight function that best fits the training data. Then during the testing step, given a new image touch behavior data, the learned weight function can compute the image saliency values.

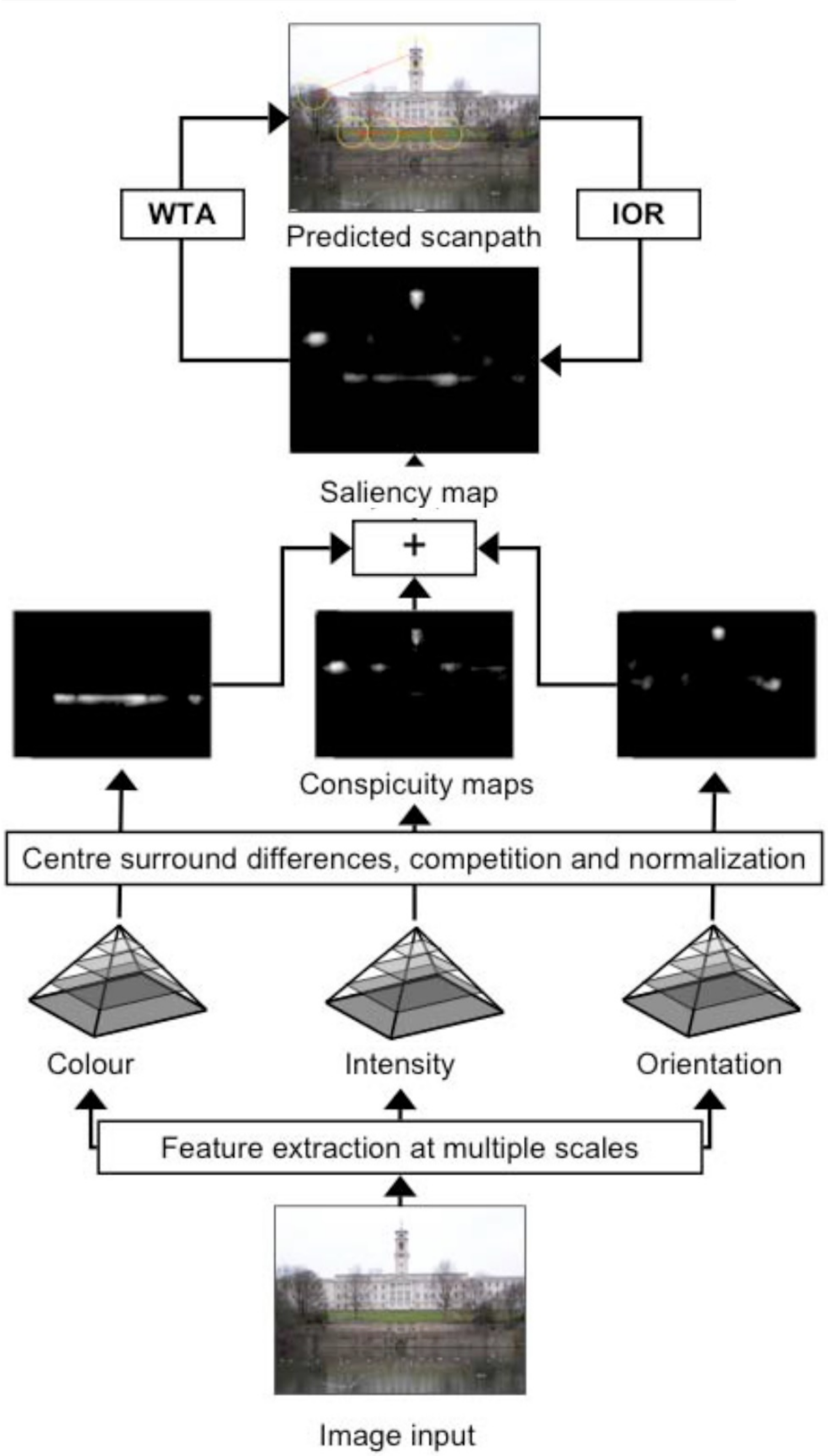

WTA
Predicted scanpath
IOR
Saliency map
+
Conspicuity maps
Centre surround differences, competition and normalization
Colour
Intensity
Orientation
Feature extraction at multiple scales
Image input

Figure 2.4: A model of visual content based method by Itti *et al.*

2.2.1 Introduction of Linear Regression

Linear regression is an approach to model the linear relationship between a scalar dependent variable H and one or more explanatory variables denoted X. A linear regression is based on a linear discriminant function of the form:

$$h(x) = w^T x \tag{2.2}$$

Where x_0=1, thus the first term $w_0 * x_0$=w_0, which is called bias. The vector $\boldsymbol{w}$ is known as the weight vector. The goal is to find a function $h(x)$ that returns the best fit on the data. Thus the modeling algorithm is to determine the weight vector $\boldsymbol{w}$ from the training data by using Function (2.3). Once the function $h(x)$ is learned, given the new data $\boldsymbol{x}$, the estimated value of $h(x)$ can be calculated using the function $h(x)$.

$$\min \sum_{k=1}^{m} \left(h(x^{(k)}) - t^{(k)} \right)^2 \tag{2.3}$$

2.2.2 Introduction of Support Vector Regression

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced by [Vapnik, 1997]. The SVM classifier is widely used in many fields due to its high accuracy, ability to deal with high-dimensional data [Schölkopf et al., 2004].

Support Vector Machine can also be used as a regression method, which is called as Support Vector Regression (SVR), maintaining all the main features that characterize the algorithm (maximal margin). SVR uses the same principles as the SVM for classification, with only a few minor differences.

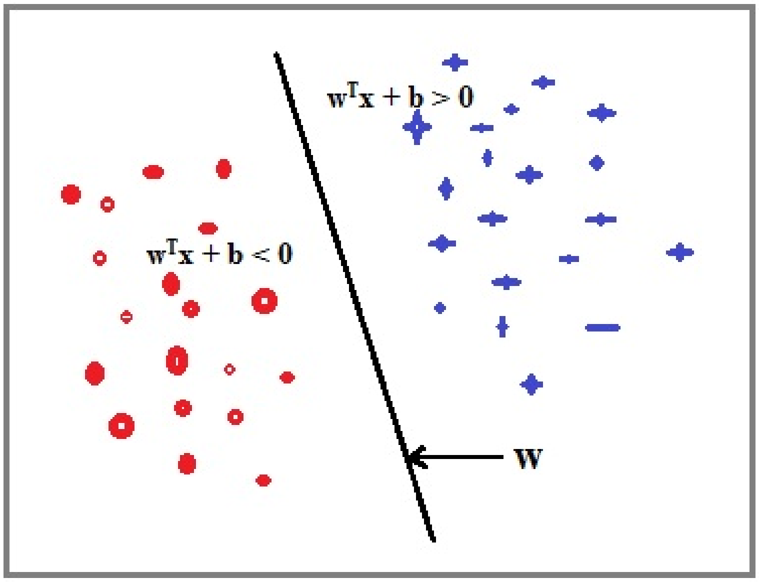


Figure 2.5: A linear SVM classifier.

In order to understand the application of SVR on our touch saliency learning, the basic background about Support Vector Machine is introduced as follows.

Suppose the data are linearly separable, then there exists a linear decision boundary that separate two classes. An illustration is shown in Figure 2.5. The decision boundary line $w^T x + b = 0$ divides the plane into two categories based on the sign of $w^T x + b$.

The circled data points in Figure 2.6 are the support vectors. They are the examples that are closest to the decision boundary. They determine the margin with which the two classes are separated.

In many applications, the data are non-linear separable. In order to provide better accuracy, a kernel function is used to make a non-linear classifier out of a linear classifier. The method is to map data from the input space X to a feature space K using a non-linear function F. The kernel can be computed without

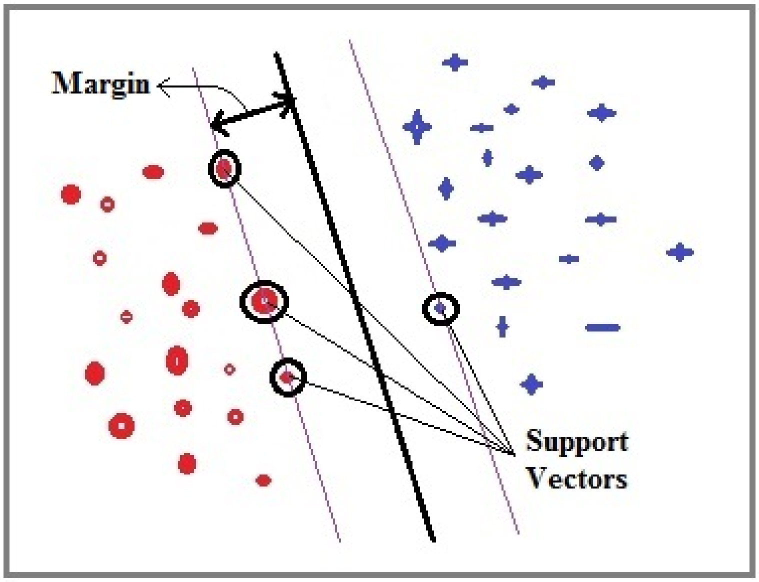


Figure 2.6: Support vectors of a linear SVM classifier.

explicitly computing the mapping K.

Up to now, support vector machines have been concerned with classification. While in case of support vector regression, the solution generated is a real number. Same as Linear Regression, Support Vector Regression also tries to find a function, $f(x)$, with at most ϵ-deviation from the target y [Smola and Schölkopf, 2004]. The SVR with soft margin is explained here.

Given training data (X_i, t_i) i = 1, 2, ..., m. Minimize

$$\frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{2.4}$$

Under constraints:

$$t_i - (\mathbf{w} \cdot x_i) - b \leq \epsilon + \xi_i$$

$$(\mathbf{w} \cdot x_i) + b - t_i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0, i = 1, ..., m$$

The above is called as primal problems. It finds primal variables w for each feature. The constant C is greater than 0, it is a parameter to control the amount of the influence of the error.

Figure 2.7 [Paisitkriangkrai, 2012] gives a visual explanation of SVR. Up until the threshold ϵ, the error is considered 0, after the error it becomes calculating the tolerant to errors: error-epsilon ξ.

In most cases, the parameters can be learned more easily in its dual information. The dual formulation also extends support vector machine to nonlinear functions. Below is the dual problem:

$$\max \begin{cases} -\frac{1}{2}\sum_{j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\Big(\varphi(x_i), \varphi(x_j)\Big) \\ -\epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} t_i(\alpha_i + \alpha_i^*) \end{cases} \tag{2.5}$$

$$s.t. \sum_{j=1}^{m}(\alpha_i - \alpha_i^*) = 0; 0 \leq \alpha_i, \alpha_i^* \leq C$$

where $(\varphi(x_i), \varphi(x_j)) = K(x, X_i)$, which is the Kernel Function.

Once the SVR is trained (by solving the dual problem), it caculates values of α_i and α_i^*, which are both 0 if x_i has no contribution to the error function. Given a new data x_{new}, prediction will be generated using the following formula:

$$f(x_{new}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, x_{new}) + b \tag{2.6}$$

There are different ways to compute the value of b, one of the ways can be found in [Gunn, 1998].

$$b = -\frac{1}{2}(\mathbf{w} \cdot (x_r + x_s)) \tag{2.7}$$

where x_r and x_s are the support vectors.

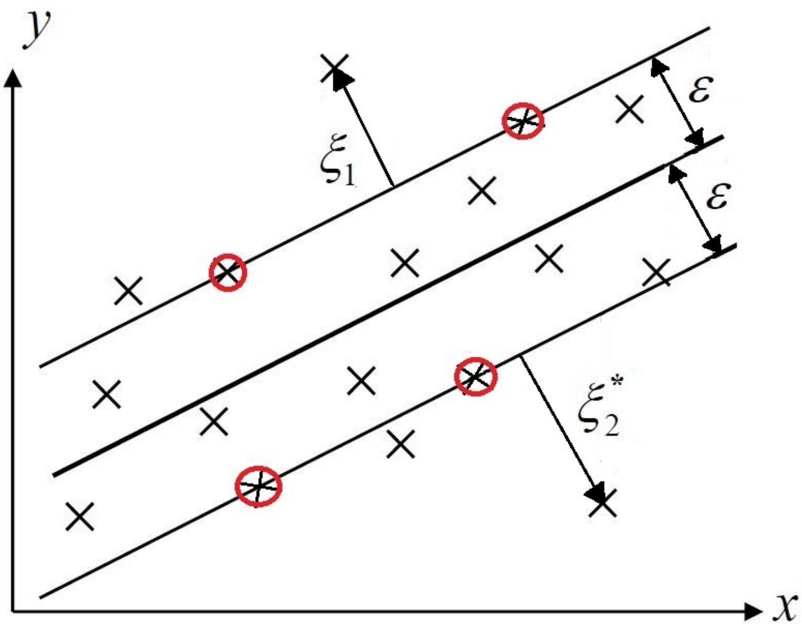


Figure 2.7: Support Vector Regression with soft margin loss function.

CHAPTER III

RELATED WORK

There have been different ways to learn where people look on a scene/image. The main two widely acceptable ways are by using eye tracking and by prediction based on the image visual content. Eye tracking is so far the most accurate method to determine human's visual attention on images. However, as stated in Chapter1, eye tracking is not widely used because of its limitations. Therefore, researchers have developed computational methods to estimate human's visual attention based on image's visual content, such as color, contrast, intensity, edge orientation information, and so on. Recently, more and more people rely on small touch devices such as touch phones, touch tablets to browse Internet, images and watch videos. When people interact with those devices, the touch information may explicitly show their interests on the screen. Hence, learning visual attention based on touch behavior becomes a hot top recently.

In the following sections, the details about these methods are introduced.

3.1 Eye Tracking Based Image Saliency

To learn the preferential visual attention given by humans to specific image content, many researchers have collected eye fixation data on different image datasets using eye tracking devices. Those data sets are very valuable and are used in the research as ground truth data that the fixation locations indicate human's real attentions on the images.

NUSEF data set [Ramanathan et al., 2010] has eye tracking fixations from a pool of 75 subjects free-viewing 758 images, which are manually collected from Flickr, Photo.net, Google Images and IAPS, containing semantically affective objects or scenes such as expressive faces (human and mammal), nudes, unpleasant concepts, and interactive actions (look, read and shoot).

MIT data set [Judd et al., 2012] has eye fixations from 300 images (223 landscape images and 77 portrait images) which are collected from Flickr Creative Commons and personal image collections. The recorded eye tracking data is from 39 users who free-viewed these images. The longest dimension of each image is 1024 pixels and the second dimension ranged from 457 to 1024 pixels with the majority at 768 pixels. This data set is not made public yet by the authors.

Another MIT data set [Judd et al., 2009] has eye tracking fixations from 15 viewers who free-viewed 1003 natural indoor and outdoor images. Created under similar conditions to the above saliency benchmark data set and can be used to train new models of saliency.

Toronto data set [Tsotsos and Bruce, 2006] contains eye fixation data from 11 subjects free-viewing 120 color images of outdoor and indoor scenes. A large portion of images here do not contain particular regions of interest.

Jianli data set [Jian Li and He, 2011] provides human eye fixations from 19 observers on 235 color images which collected using Google as well as by consulting the recent literature. The images in this database are 480 x 640 pixels and are divided into six different categories.

However, these collected data sets require eye tracking devices, which has following disadvantages that limits its wide usage:

1) Eye tracking devices are normally expensive.

2) Operating the eye tracking devices is tedious and complicated, it requires non-trivial calibration, validation, and chin-and-forehead-rest for stabilization.

3) The users have to be trained to operate it.

4) It has low mobility. It is hard to carry it everywhere due to its considerable size and weight.

3.2 Visual Content Based Prediction Image Saliency

Many computational methods of saliency have been developed from a wide variety of different approaches to detect where people are interested in images. These methods are mainly based on image's visual content, such as color, contrast, intensity, edge orientation information, and so on. In order to know the performance of the methods and have comparisons between models, researchers have built different benchmark data sets. A brief introduction of the important and popular computational models of saliency is given as follows:

[Itti et al., 1998] introduced a model for bottom-up selective attention based on serially scanning a saliency map, which is computed from local feature contrasts (In total, 42 feature maps are computed: 6 for intensity, 12 for color channel, and 24 for local orientation information), for salient locations in the order of decreasing saliency. Each feature is computed by a set of linear center-surround operations related to visual receptive fields. The feedback from higher cortical areas was used to weight the importance of different features and such that only those with high weights could reach higher processing levels.

The Graph Based Visual Saliency (GBVS) model [Harel et al., 2007] is a graph-based implementation of the Itti model that by using dissimilarity and saliency to define edge weights on graphs which are interpreted as Markov chains. The mass-concentration on individual activation maps prior to additive combination is performed in order to have the resulting master map informative. As the result of concentrating activation, a few key locations are considered as salient areas.

Bruce and Tsotsos' Attention based on Information Maximization (AIM) [Tsotsos and Bruce, 2006] is a model for visual saliency computation built on a first principles information theoretic formulation which is called as Attention based on Information Maximization. It aims to maximize information sampled from a scene and is derived from mathematical principles.

[Hou et al., 2012] proposed to calculate the saliency map using the inverse cosine transform of the signs of the cosine transformed image, the discrete cosine transform (DCT) image signature approach, which defines the saliency using the inverse DCT of the signs in the cosine spectrum. The approach was evaluated on the Toronto eye-tracking data set (Bruce and Tsotsos, 2009) to determine how well it predicts human eye fixations. It was reported not just to be faster than other approaches but also to outperform several established approaches.

These computational models of image saliency have been proven useful in many fields. However, it has several disadvantages:

1) This type of methods are based on image's visual content information. It predicts human's attention without involving human subjects. Therefore, the image saliency map is identical ro everyone, it can not be personalized.

2) The performance one some types of images is low, such as low contrast images, high intensity images. It can not detect the human's interests on these types of images well.

3) The computational cost is generally high.

3.3 Touch Behavior Based Image Saliency

[Xie et al., 2005] made the first attempt to extract user attention by analyzing the touch information on images in 2005. They learned the user attention from 10 subjects' touch data on 26 images. Several attributes are considered in the users attention learning including region of interest, minimal allowable spatial area of the attention, minimal duration of the attention etc. This study demonstrates that users attention can be easily obtained from touch behaviors. However, its performance is not quantitatively evaluated. Therefore, its validity is unknown.

[Xu et al., 2012] introduced a new concept of touch saliency, which is to generate image saliency maps based on human simple zoom behavior. In their data collection, 16 participants freely viewed 440 images in NUSEF database [Ramanathan et al., 2010] on a touch-screen mobile device. The center point of the screen is treated as the fixation point and the zoom scale is used as Gaussian filter parameters to generate the touch saliency map. This study shows that touch saliency map and eye fixation map are highly correlated with each other in an image browsing task.

It is observed that when users browse images, they tend to perform a variety of touch behaviors, such as pinch zoom, scroll, tap, double tap, ans so on. However, [Xu et al., 2012] generates saliency maps based solely on simple zoom behavior. Meanwhile, the image pixel of center point of the screen is selected as the fixation point, which always causes some bias in the saliency map learning. It is observed that when the image is zoomed in, the users do not always adjust the most salient area to the center of the screen. An example is shown in Figure 3.1 .

Figure 3.2 gives another example when user uses pinch zoom to zoom in particular region of an image. It shows that when pinch zoom behavior is used to browse images, the most salient area is more likely between two fingers. Meanwhile, as the zoomed-in region is already clear on the screen, user does not necessarily bring the most salient region to the center of the screen.

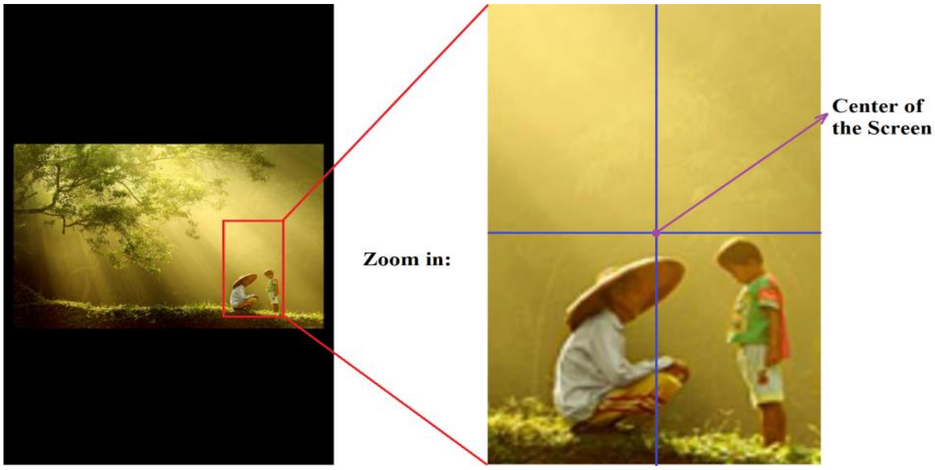


Figure 3.1: An example of Zoom-in behavior on an image.

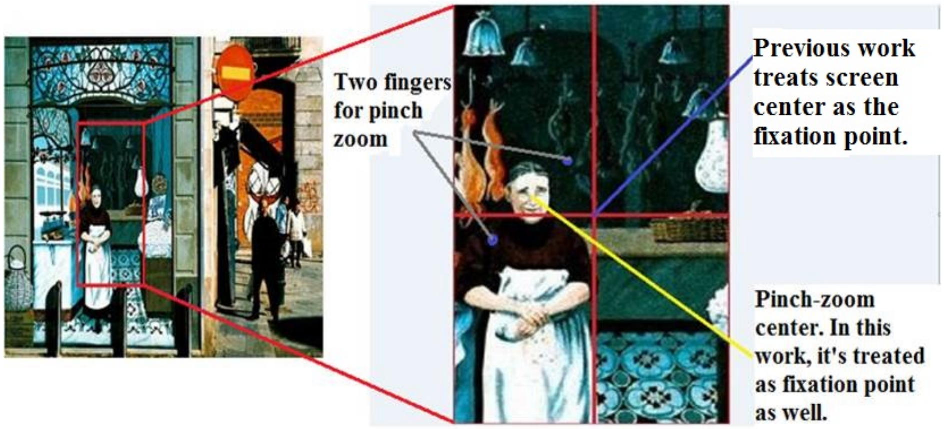


Figure 3.2: An example of pinch zoom-in behavior on an image.

CHAPTER IV

IMAGE SALIENCY LEARNING FROM MULTIPLE TOUCH BEHAVIORS

In the preliminary study, it is observed that when browsing images on the limited size touch screen, users tend to apply a variety of touch behaviors, such as tap, pinch zoom, double tap zoom, and scroll to find a particular region of interest and look them closer. What correlations between these different behaviors and human attention are, whether they contribute equally to the human eye fixation, and how to learn good image saliency maps from multiple touch behaviors have not been explored in the existing studies. To our best knowledge, this is the first attempt that conducts a series of studies to explore these questions.

4.1 Framework

In order to learn the relationship between different touch behaviors and the human attentions on the images, all the touch behavior data collected from the user study is thoroughly analyzed and five features that may indicate human's interest and attention on certain regions of the image are proposed.

Different from previous touch saliency generation methods, two novel learning based approaches are proposed to generate image saliency maps from the touch behaviors data.

The learning framework is shown in Figure 4.1, it contains two stages: training and testing. During the training stage, the weight of each behavior can be learned by using machine learning methods, such as Linear Regression and Support Vector Machine Based Regression, and the weight indicates how many contributions each touch behavior makes to the touch saliency. In the testing stage, given collected touch behavior data of a new image, its touch saliency map can be predicted with the learned weights. Above all, the proposed learning based approach can successfully explore the correlation between each touch behavior feature and human attention. This thus leads to a good saliency map from these touch behaviors.

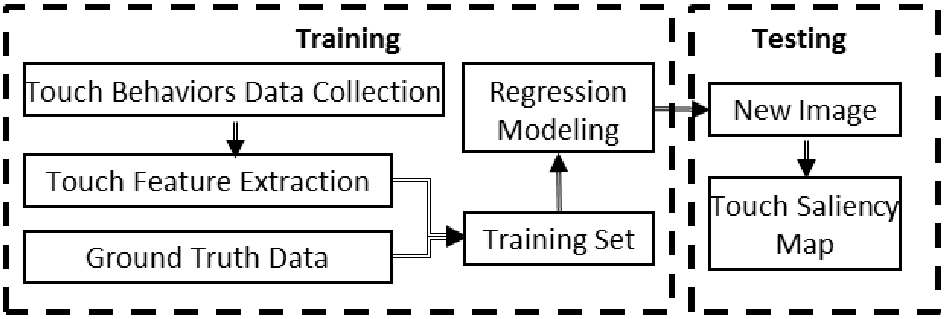


Figure 4.1: Touch Saliency Learning Framework.

4.2 Touch Behavior Data Collection

4.2.1 Image Browsing App Design

In order to collect user touch behavior data, an image browsing interface on a multi-touch mobile phone is developed. The interface is designed as same as most popular image browsers which support tap, pinch zoom, double tap zoom, scroll, etc. The platform is Android Development and the programming language is Java. The device used to install and test is Samsung Galaxy S3 Android phone. It has 4.8 inch HD Super AMOLED display with 1280x720 pixels, 2GB RAM, 16GB storage.

The image browsing application displays each image for 12 seconds. A black screen is shown for 2 seconds between any two consecutive images to avoid interference. Every time we start the application, the images are shown in a random order. Thus, the display orders may be different for each participant to avoid bias. The program has ability to keep recording the touch gesture type, center pixel coordinates of the pinch zoom, double tap coordinates, image pixel coordinates of center point of the screen, scroll target position, tap point coordinates, and zoom out pixel coordinates. In general, all the touch gestures and data are collected while a user freely browses the images using this application on the Samsung Galaxy S3 Android phone.

This application is similar to the one designed in [Xu et al., 2012] in terms of user experience. The main differences are:

1) Their application was developed in iOS development environment and they used an iPhone which only has 320x480 resolution. The application in this study is developed in Android development, and Samsun Galaxy S III phone with 1280x720 resolution is used.

2) The application in this study not only record the center location of the screen corresponding to the image pixel location after zoomed in but also collect all the touch gesture data (Such as touch gesture type: pinch zoom in/out,

double tap zoom in/out, scroll, tap, time, image width, image height and so on.) and corresponding image coordinates where each gesture performed.

4.2.2 Image Data Set

The same data set NUSEF [Ramanathan et al., 2010] used in the work [Xu et al., 2012] is chosen in our study by considering its two unique attributes. First, this data set contains 446 images (size is around 1024x768 pixels) and the corresponding ground truth eye fixation data acquired from an eye-tracking device with a pool of 75 subjects. Second, the images in this dataset are everyday scenes and manually collected from Flickr, aesthetic content from Photo.net, Google Images and emotion-evoking IAPS, and they are representatives of various semantic concepts, scales, orientations, and illuminations.

Figure 4.2 gives some image samples from NUSEF image dataset.

Table 4.1 summarizes the diverse categories covered in the NUSEF eye-fixation database.

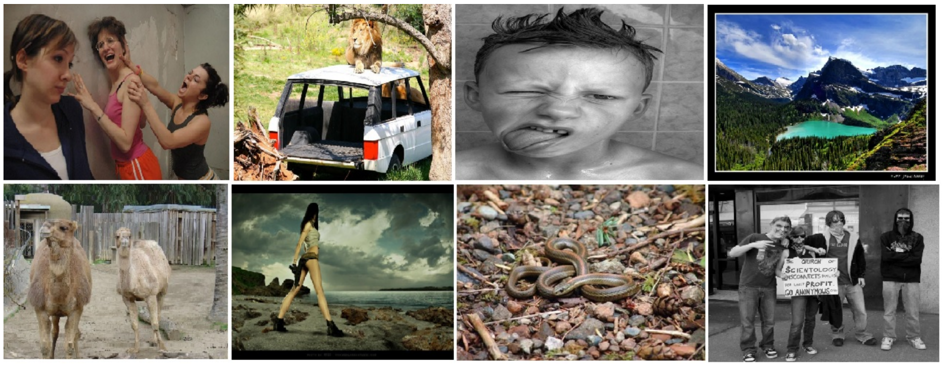


Figure 4.2: Image samples from NUSEF image dataset.

Table 4.1: Semantic categories of images in NUSEF data set.

Semantic Category	**Image Description**	**Image Count**
Face	Single or multiple human/mammal faces.	77
Portrait	Face and body of single human/mammal.	159
Nude		41
Action	Images with a pair of interacting objects (as in *look*, *read* and *shoot*).	60
Affect-variant group	Group of 2-3 images with varying affect.	46
Other concepts	Indoor, outdoor scenes, *world* images comprising living and non-living entities, *reptile*, *injury*.	375

4.2.3 User Study

15 users (4 females, 11 males) from Computer Science Department at Texas State University participated in our user study. Their ages are between 24 and 33 $(\mu = 26.6, \sigma = 2.75)$. Before they started to browse the image, they signed the consent forms and filled out the participation forms which record their basic information.

Before start of the user study, each participant was told to freely browse the image as he/she usually does when browse the image on touch screen phones. Each user viewed all the 446 images (from NUSEF data set) on the Samsung Galaxy S3 Android phone using the application described in section 3.2.1. Each user can use any touch behavior to move to a particular region of interest. During the image browsing process, each image is displayed for 12 seconds. In order to avoid interference, a black screen is shown for 2 seconds between any two consecutive images. For every user, all images are displayed in a random order. Thus, the display orders may be different for each participant to avoid bias. The program keeps recording the touch behavior type, center pixel coordinates of the pinch zoom, double tap coordinates, pixel coordinates of center point of the screen, scroll target position, tap point coordinates, *etc.*

In order to protect participants' confidentiality, each participant was assigned a number. Any data collected during the user study was recorded by number, not by name.

4.3 Touch Behavior Feature Extraction

Do different touch behaviors correspond to different human attentions? Which behaviors are highly correlated with human eye fixation? How to learn a good image saliency map from various/multiple human behaviors? In order to find answers to those questions, analysis for the touch data acquired from user study is necessary. It is observed that during the user study, users use different touch gestures to manipulate the image on the phone to have a better and closer view on the image. In addition to the feature (the center point of the screen) published in [Xu et al., 2012], four main touch features that may indicate human's interest and attention on certain regions of the image are proposed.

The total five features abstracted from the touch data are listed as follows with the corresponding descriptions:

- Tap (T): Image pixel coordinates of the tap point.
- Pinch-zoom-in (P): Image pixel coordinates of the center point between two

fingers after zoom in.

- Scroll (S): Image pixel coordinates of the scrolling point after zoom in.
- Double-tap-zoom-in (D): Image pixel coordinates of the double-tap point and the zoom scales of the doubletap zoom in/out on images.
- Center (C): Image pixel coordinates of the center point of the touch screen after zoom in.

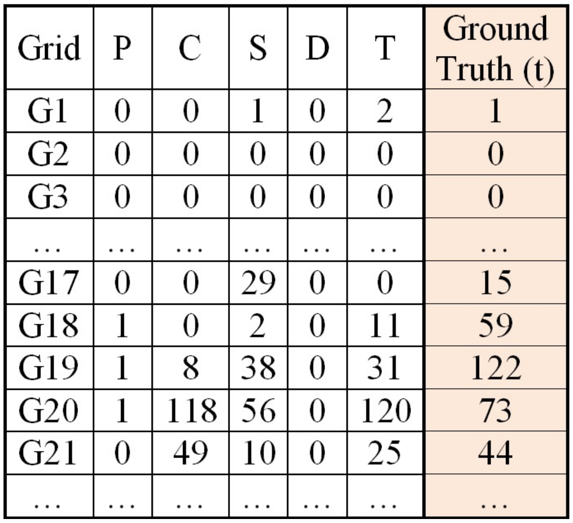
4.4 Touch Saliency Learning

This section will explain design and implementation of the proposed Touch Saliency Learning methods on images based on users touch behavior data. The key work is to learn models from a training data set which contains touch behavior data. The two major Machine Learning methods used to learn image saliency in this thesis are supervised regression methods: Linear Regression and Support Vector Machine based regression (SVR).

Let $R = \{I_1, I_2, I_3, ..., I_m\}$ be a set of training images, then divide an I_k into a by b grids, $G_{I_k} = \{g_{I_k}^1, g_{I_k}^2, g_{I_k}^3, ..., g_{I_k}^{ab}\}$, where $g_{I_k}^j = (g_{I_k}^{T_j}, g_{I_k}^{P_j}, g_{I_k}^{S_j}, g_{I_k}^{D_j}, g_{I_k}^{C_j}) \in R^5$ is a touch feature vector extracted from the j-th grid. The value of these five touch behavior features $g_{I_k}^{T_j}, g_{I_k}^{P_j}, g_{I_k}^{S_j}, g_{I_k}^{D_j}, g_{I_k}^{C_j}$ are calculated by counting the number of times the corresponding behavior happens in the j-th grid of image I_k. For example, if 10 tap points are found in the j-th grid of image I_k , its corresponding value $g_{I_k}^{T_j}$ is 10. Obvioursly, the more frequent the touch behaviors happen in one grid, the more attentions are given to that grid by users.

The eye fixation maps acquired from the eye-tracking device are used as the ground truth data, which reflect real visual attention information. The eye fixation map is a grayscale image and each pixel's value ranges from 0 to 255. The higher the value is, the more salient that pixel is. Each eye fixation map is also divided into a by b grids. More pixels in one grid has high value and indicate that grid attarcts a lot of attention. Therefore, the target real visual attention value of the j-th grid in image I_k: $t^j_{I_k}$ is approximated as the average of all the pixel values in the j-th grid. Apparently, if more pixels in one grid has high value, it indicates that grid attarcts a lot of attention. A table of sample training data set is shown in Table 4.2. The value for each grid is calculated by counting the number of times the corresponding feature happens in the grid of an image.

Table 4.2: Sample of training data.



Grid	P	C	S	D	T	Ground Truth (t)
G1	0	0	1	0	2	1
G2	0	0	0	0	0	0
G3	0	0	0	0	0	0
…	…	…	…	…	…	…
G17	0	0	29	0	0	15
G18	1	0	2	0	11	59
G19	1	8	38	0	31	122
G20	1	118	56	0	120	73
G21	0	49	10	0	25	44
…	…	…	…	…	…	…

4.4.1 Learning Image Saliency from Multiple Touch Behaviors Using Linear Regression

In this thesis, we propose a method, touch saliency from multiple touch behaviors by linear regression (TSMB-LR), is the simple, easy, powerful and fast way to predict the salient value of the regions on the image.

Since different touch behaviors may contribute differently to the touch saliency value of each grid, we propose to use linear regression method to generate the touch saliency value for the i-th grid in image I_k in a linear function:

$$h(g_{I_k}^i)=w_0+w_T g_{I_k}^{T_i}+w_P g_{I_k}^{P_i}+w_S g_{I_k}^{S_i}+w_D g_{I_k}^{D_i}+w_C g_{I_k}^{C_i} \tag{4.1}$$

w_T, w_P, w_S, w_D and w_C are the corresponding weights of the five features, which implicitly indicate correlation between each behavior and touch saliency value $h(g_{I_k}^i)$.

The touch saliency learning problem is formulated as a linear regression algorithm, which learns the weight of each behavior by solving the following minimization function:

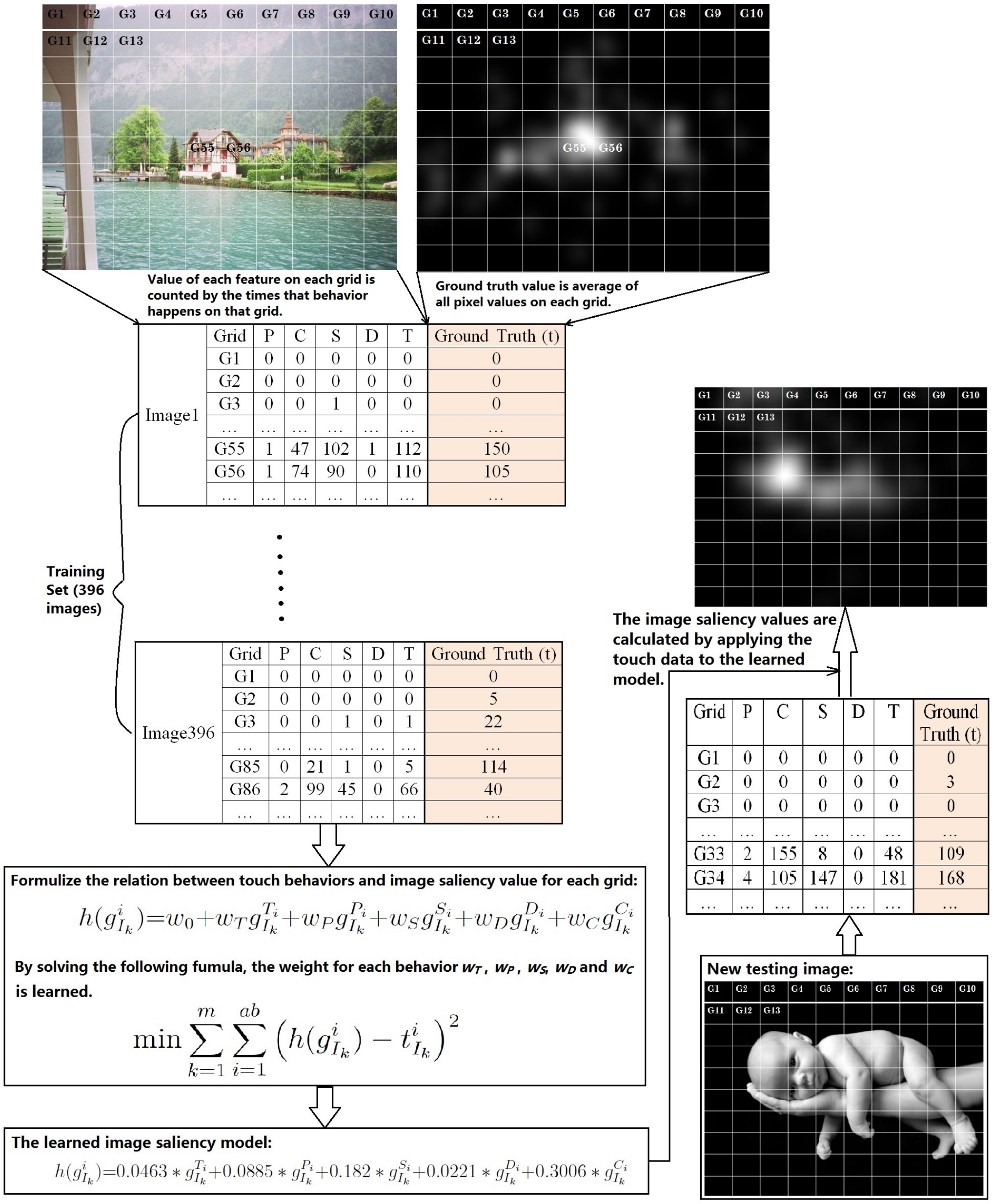
$$\min \sum_{k=1}^{m} \sum_{i=1}^{ab} \left(h(g_{I_k}^i) - t_{I_k}^i \right)^2 \tag{4.2}$$

$t_{I_k}^i$ is the target value. During the training stage, the weight of each behavior can be learned by solving this minimization function, and it indicates how many contributions each touch behavior makes to the touch saliency value. In the testing stage, given collected touch behavior data of a new image, its touch saliency map can be predicted with the learned weights based on Function (4.1). Above all, the proposed learning based approach can successfully explore the correlation between each touch behavior feature and human attention. This thus leads to a good saliency map from those touch behaviors.

Figure 4.3 shows an example of architecture of grid size 10x10 by using TSMB-LR method. There are main two steps: training and testing. At the training step, the five features are extracted from the human touch behaviors, and then the value of each feature on each grid is counted by the times that behavior happens in that grid. With the ground truth pixel value of each grid (average of all the pixel values on the grid) from eye tracking saliency maps, the weight for each grid is learned from training set (touch features data for 396 images) by using the linear regression method. At the testing step, given a new image with human touch behavior data, the value of each feature on each grid also counted. With the weight for each feature learned from training step, the predicted image saliency value is calculate by Function (4.1). At the end, all the predicted values of 100 grids are normalized to 0-255 in order to generate the grayscale image saliency map.

4.4.2 Learning Image Saliency from Multiple Touch Behaviors Using Support Vector Regression

The linear regression based image saliency learning approach only can learn a linear function that represents the relationship between human touch behaviors and image saliency. Therefore, in our study, we propose to use Touch Saliency from Multiple touch Behaviors by Support Vector Regression (TSMB-SVR) approach to learn a non-linear relationship between human touch behaviors and image saliency as the TSMB-SVR can learn a kernel function to fit the data. In this work, polynomial


G1 G2 G3 G4 G5 G6 G7 G8 G9 G10
G11 G12 G13
G55 G56
G1 G2 G3 G4 G5 G6 G7 G8 G9 G10
G11 G12 G13
G55 G56
G1 G2 G3 G4 G5 G6 G7 G8 G9 G10
G11 G12 G13
G1 G2 G3 G4 G5 G6 G7 G8 G9 G10
G11 G12 G13
Training Set (396 images)

Value of each feature on each grid is counted by the times that behavior happens on that grid.

Ground truth value is average of all pixel values on each grid.

Image1

Grid	P	C	S	D	T	Ground Truth (t)
G1	0	0	0	0	0	0
G2	0	0	0	0	0	0
G3	0	0	1	0	0	0
…	…	…	…	…	…	…
G55	1	47	102	1	112	150
G56	1	74	90	0	110	105
…	…	…	…	…	…	…

Image396

Grid	P	C	S	D	T	Ground Truth (t)
G1	0	0	0	0	0	0
G2	0	0	0	0	0	5
G3	0	0	1	0	1	22
…	…	…	…	…	…	…
G85	0	21	1	0	5	114
G86	2	99	45	0	66	40
…	…	…	…	…	…	…

Formulize the relation between touch behaviors and image saliency value for each grid:

$$h(g^i_{I_k}) = w_0 + w_T g^{T_i}_{I_k} + w_P g^{P_i}_{I_k} + w_S g^{S_i}_{I_k} + w_D g^{D_i}_{I_k} + w_C g^{C_i}_{I_k}$$

By solving the following fumula, the weight for each behavior w_T, w_P, w_S, w_D and w_C is learned.

$$\min \sum_{k=1}^{m} \sum_{i=1}^{ab} \left(h(g^i_{I_k}) - t^i_{I_k} \right)^2$$

The learned image saliency model:

$$h(g^i_{I_k}) = 0.0463 * g^{T_i}_{I_k} + 0.0885 * g^{P_i}_{I_k} + 0.182 * g^{S_i}_{I_k} + 0.0221 * g^{D_i}_{I_k} + 0.3006 * g^{C_i}_{I_k}$$

The image saliency values are calculated by applying the touch data to the learned model.

Grid	P	C	S	D	T	Ground Truth (t)
G1	0	0	0	0	0	0
G2	0	0	0	0	0	3
G3	0	0	0	0	0	0
…	…	…	…	…	…	…
G33	2	155	8	0	48	109
G34	4	105	147	0	181	168
…	…	…	…	…	…	…

New testing image:

Figure 4.3: Architecture of grid size 10x10 by using TSMB-LR.

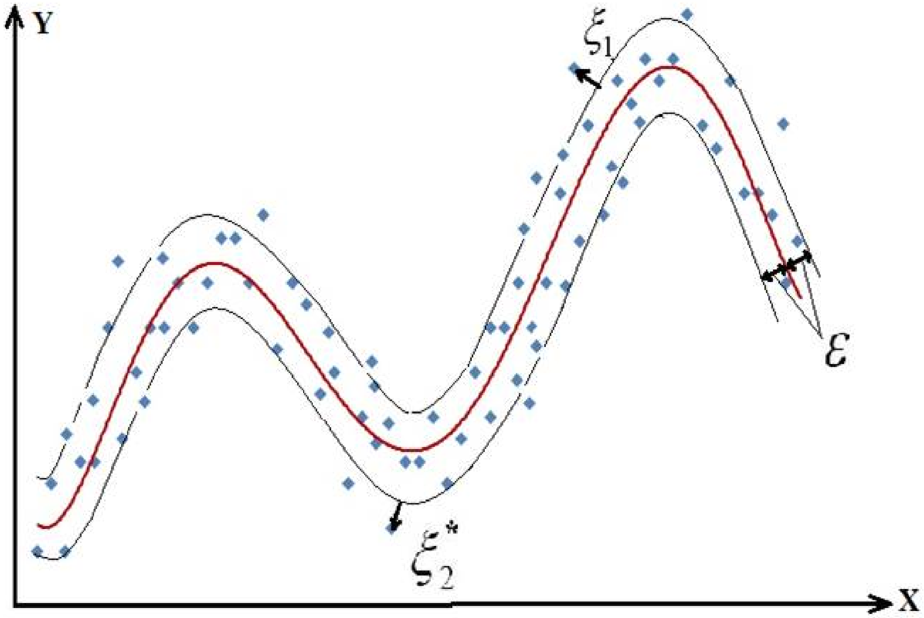


Figure 4.4: Polynomial Kernel Function for Support Vector Regression.

kernel function is used to learn the image saliency model from multiple touch behaviors as shown in Figure 4.4.

The training data used for TSMB-SVR is the same as the one for TSMB-LR. Both algorithms train the data to find the weight for all five touch behavior features. TSMB-LR tries to minimize the difference between the prediction value and the target value and meanwhile solves the weight function. TSMB-SVR finds support vectors with the pre-set parameter epsilon and tolerance parameter so that data fits the SVR prediction model.

Figure 4.5 [Smola and Schölkopf, 2004] gives an overview over the different steps in the regression training stage.

In this image saliency learning study, the Sequential Minimal Optimization (SMOreg) [Boser et al., 1992] from Weka is used. Weka is a collection of machine

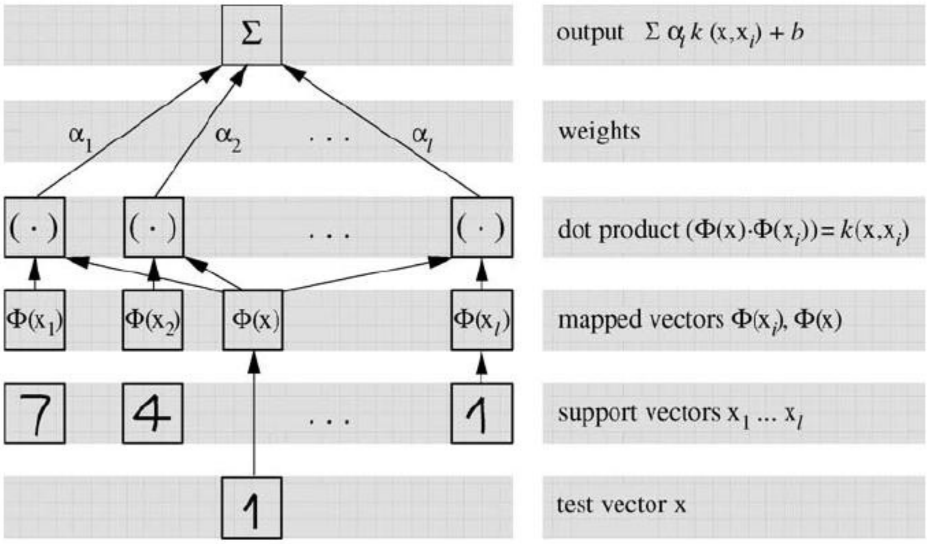

Σ
output $\Sigma\, \alpha_i k(x, x_i) + b$
α_1
α_2
. . .
α_l
weights
(·)
(·)
. . .
(·)
dot product $(\Phi(x) \cdot \Phi(x_i)) = k(x, x_i)$
$\Phi(x_1)$
$\Phi(x_2)$
$\Phi(x)$
$\Phi(x_l)$
mapped vectors $\Phi(x_i)$, $\Phi(x)$
7
4
. . .
1
support vectors $x_1 \ldots x_l$
1
test vector x

Figure 4.5: Architecture of regression machine constructed by the SVR algorithm.

learning algorithms for solving real-world machine learning and data mining problems. It is written in Java. The algorithms can be applied directly to a dataset. Therefore, the primary job is to optimize the parameters to have both good accuracy and low execution time.

The touch saliency learning problem is formulated as a support ventor regression algorithm, which learns the weight of each behavior by solving the following minimization function:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \tag{4.3}$$

Under constraints:

$$t_{I_k}^i - (\mathbf{w} \cdot \mathbf{g}_{\mathbf{I_k}}^{\mathbf{F_i}}) - b \leq \epsilon + \xi_i$$

$$(\mathbf{w} \cdot \mathbf{g}_{\mathbf{I_k}}^{\mathbf{F_i}}) + b - t_{I_k}^i \leq \epsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0, i = 1, ..., m$$

where $\mathbf{w}{=}\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i$ is in the dual formulation, and $\mathbf{g}_{\mathbf{I_k}}^{\mathbf{F_i}}$ is a vector of the input features which are w_T, w_P, w_S, w_D, and w_C. The prediction value of the j-th grid in image I_k is:

$$h(g_{I_k}^{j}){=}\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, \mathbf{g}_{\mathbf{I_k}}^{\mathbf{F_i}}) + b \tag{4.4}$$

As the kernel function does not calculate the mapping features explicitly, the resulting regression model is also expressed as a linear function with weight for each input feature.

CHAPTER V

EXPERIMENT AND RESULT

5.1 Experiment Design

In order to evaluate the touch saliency methods, the comparison of our approach with other five state-of-the-arts on the NUSEF data set was conducted. These five state-of-the-arts include four visual saliency map generation methods (Itti Model (Itti) [Itti et al., 1998], Graph Based Visual Saliency (GBVS) [Harel et al., 2007], Attention via Information Maximization (AIM) [Tsotsos and Bruce, 2006], Image Signature model (Sign.) [Hou et al., 2012]. which derive saliency maps based on image visual content information and one touch saliency generation approach (center-based touch saliency map (Center) [Xu et al., 2012]).

In this thesis, different number of grids (aXb) are also tested, which range from $10X10, 14X14, 18X18, 22X22, 26X26, 30X30, 40X40, 50X50, 60X60$ to image_width X image_height. When the number of grids is set as image_width X image_height (WXH), every recorded pixel in the image is chosen as one grid. In that case, the mild outliers are removed using the quartile method (lower quartile = 0th percentile, higher quartile = 75th percentile) for scrolling and tap features, since most users tend to continiously scroll and accidentally tap the image on the screen.

The data set used for experiments is NUSEF dataset [Ramanathan et al., 2010]. The major reason we choose it is that has eye tracking fixations from a pool of 75 subjects free-viewing 758 images, which are manually collected from Flickr, Photo.net, Google Images and IAPS, containing semantically affective objects or scenes such as expressive faces (human and mammal), nudes, unpleasant concepts, and interactive actions (look, read and shoot). In our experiments, the NUSEF dataset is divided into a training set (396 images) and a testing set (50 images). The evaluation of our approaches, other state-of-the-arts cumputational models, and center-based touch saliency method is only performed on the testing data set.

5.2 Evaluation Metrics

In order to evaluate the performance of our touch saliency learning from multiple touch behaviors algorithm (TSMB), two popular saliency performance evaluation metrics – AUC (Area under ROC Curve) and CC (Correlation Coefficient) are ultilized in this thesis.

AUC calculates the area under the ROC curve (true positives (sensitivity) vs. false positives (1 - specificity)). The AUC value is always between 0 and 1.0. A value of 1 for AUC represents a perfect test. Random guessing has an area of 0.5. Realistic classifier should have an AUC greater than 0.5. AUC is for a binary classifier system as its discrimination threshold varies. For this evaluation, the threshold to define a binary classifier (Salient or Non-salient) on a grid is the pixel value 14. If the average pixel value on the grid is bigger than or equal to 14, it's treated as salient grid, otherwise, it's non-salient grid. It is easy to understand that the different thresholds would produce different results. The experiments can be conducted to determine the best threshold in the future.

CC is a measure of the strength of a linear association between two saliency maps. It can range from -1 to +1. A value of +1 represents a perfect positive correlation while a value of -1 represents a perfect negative correlation. A value of 0 indicates that there is no relationship between the saliency maps being tested.

A good saliency map should have both high AUC score (maximum value is 1) and CC score (maximum value is 1). In our experiments, AUC and CC calculation are programmed by using Matlab.

5.3 Experiment Results

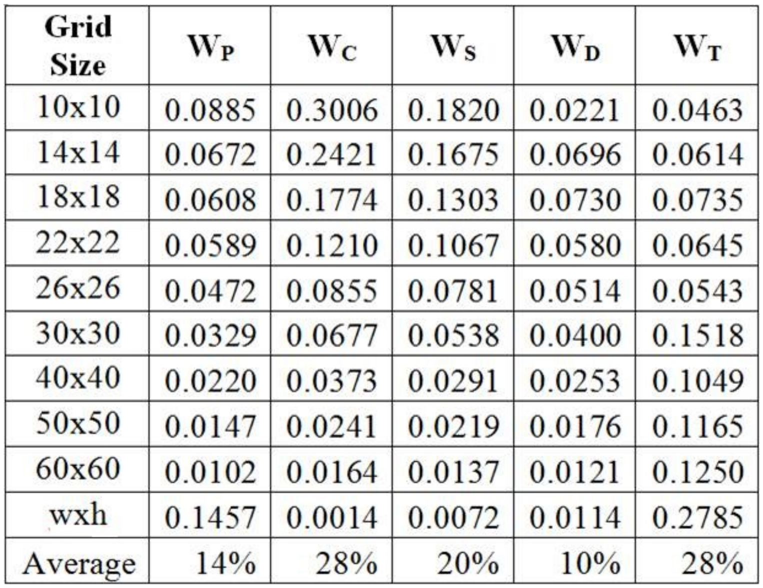
5.3.1 Linear Regression Result

After traning the model on the collected training data using linear regression, the weights of features w_T, w_P, w_S, w_D and w_C are learned, as shown in Table 5.1, whose average value are 28%, 10%, 20%, 10%, and 28% respectively. These learned weights show that all the features contribute to the touch saliency, but in the different degree. Center point of screen and the tap behavior are the most important ones. Scrolling is the third important touch behavior. Pinch-zoom-in and Double-tap-zoom-in make less contribution to the visual attention information. The weights of Pinch-zoom-in and Double-tap-zoom-in are similar, which does make sense as both behaviors are used to zoom in images.

Figure 5.1 shows the generated saliency maps of these methods.

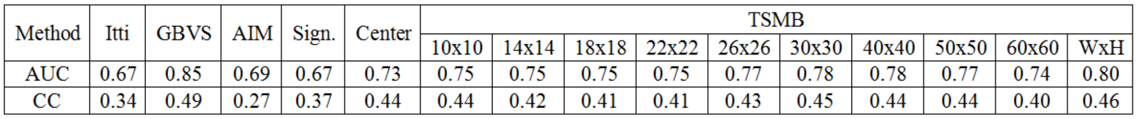
The comparison result based on both AUC and CC is listed in Table 5.2. From the results, it can be observed that:

Table 5.1: The weight of each feature learned by Linear Regression.



Grid Size	W_P	W_C	W_S	W_D	W_T
10x10	0.0885	0.3006	0.1820	0.0221	0.0463
14x14	0.0672	0.2421	0.1675	0.0696	0.0614
18x18	0.0608	0.1774	0.1303	0.0730	0.0735
22x22	0.0589	0.1210	0.1067	0.0580	0.0645
26x26	0.0472	0.0855	0.0781	0.0514	0.0543
30x30	0.0329	0.0677	0.0538	0.0400	0.1518
40x40	0.0220	0.0373	0.0291	0.0253	0.1049
50x50	0.0147	0.0241	0.0219	0.0176	0.1165
60x60	0.0102	0.0164	0.0137	0.0121	0.1250
wxh	0.1457	0.0014	0.0072	0.0114	0.2785
Average	14%	28%	20%	10%	28%

Table 5.2: AUC and CC comprision results for Linear Regression.



Method	Itti	GBVS	AIM	Sign.	Center	TSMB									
						10x10	14x14	18x18	22x22	26x26	30x30	40x40	50x50	60x60	WxH
AUC	0.67	0.85	0.69	0.67	0.73	0.75	0.75	0.75	0.75	0.77	0.78	0.78	0.77	0.74	0.80
CC	0.34	0.49	0.27	0.37	0.44	0.44	0.42	0.41	0.41	0.43	0.45	0.44	0.44	0.40	0.46

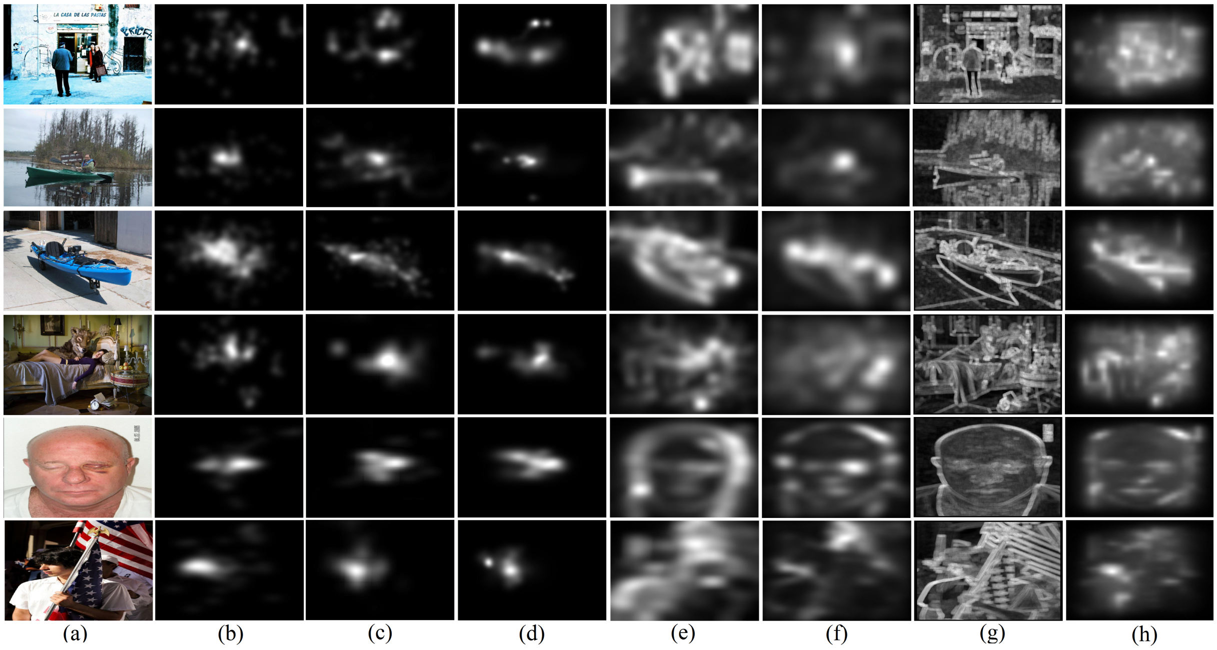


Figure 5.1: Saliency Maps. From left to right: a) original image, b) NUSEF eye fixation map, c) our touch saliency map using Linear Regression (gird size: image_width x image_height), d) Center saliency map, e) Itti saliency map, f) Signature saliency map, g) AIM saliency map, h) GBVS saliency map.

- Our touch saliency learning algorithm TSMB-Linear Regression (TSMB-LR) outperforms the center-based touch saliency learning method (Center). The AUC value has been improved from 0.73 to 0.80 and CC value is also close to 0.44. On average, the WxH TSMB-LR approach improved the prediction accuracy by 7.25% over the center-based method. The major reason is that the center-based method only considers room-in/out behavior. Actually, it is found in our study that all the touch behaviors contribute to the human attention. Tap and scrolling behaviors even make more contributions than zoom does.

- The touch saliency map generated by our algorithms has better quality than the saliency map derived by many complex and expensive visual-based approaches. This observation is exciting. Although multiple touch behaviors may involve noise, the generated touch saliency map still has high quality and the touch saliency learning approach is much cheaper, faster, and more efficient than visual-based approaches. It is surprised that our saliency map quality is also very close to the map generated by GBVS, which is well-known for excellent performance but high complexity and computational cost. This observation future validates the effectiveness and efficiency of our approach.

- As the number of grids increases (the grid size decreases), the accuracy of the learned saliency map also increases. Even if the image is roughly divided into 10x10 grids, the performance is still very good. Therefore, the users can freely choose the best number of grids based on their applications' need. If the application has high requirement on the execution time, 10x10 is a good choice. If the accuracy is the first priority of the application, WxH should be chosen.

5.3.2 Support Vector Regression Result

SMOreg [Boser et al., 1992] implements the support vector machine for regression. The parameters can be learned using various algorithms. The algorithm is selected by setting the RegOptimizer. In this study, RegSMOImproved (the most popular algorithm) is selected to optimize the parameters. Polynomial kernel function used in SMOreg is:

$$K(x, x^{'}) = (x^T x^{'})^d$$

or

$$K(x, x^{'}) = (x^T x^{'} + 1)^d$$

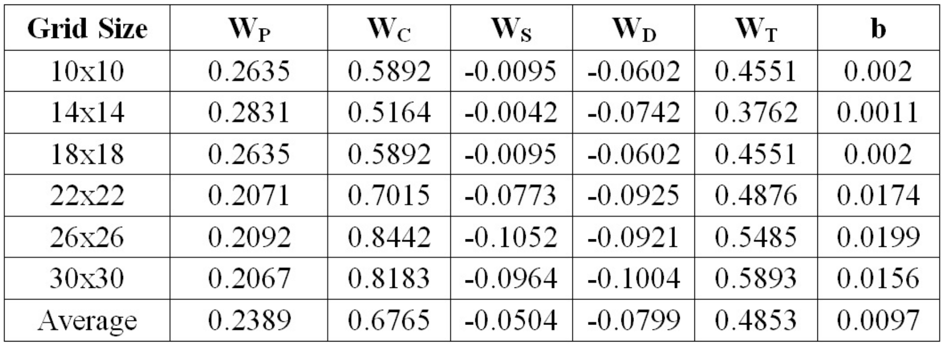
The feature space for this kernel consists of all monomials up to degree *d*. In our experiments, degree *d* is set to 1 for all the grid sizes of images.

The SMOreg implementation globally replaces all missing values and transforms nominal attributes into binary ones. Before learning the model, all the attributes are normalized. The coefficients (weight factors) in the output are based on the normalized data, not the original data.

Table 5.3 shows the weights learned from Weka using SMOreg algorithm.

Once the regression model is learned by SMOreg, the testing data is applied to the model to calculate the prediction salient values for each grid. The prediction

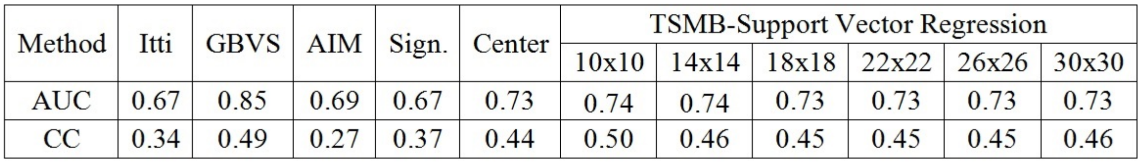
Table 5.3: The weight of each feature learned by SMOreg.



Grid Size	$\mathbf{W_P}$	$\mathbf{W_C}$	$\mathbf{W_S}$	$\mathbf{W_D}$	$\mathbf{W_T}$	**b**
10x10	0.2635	0.5892	-0.0095	-0.0602	0.4551	0.002
14x14	0.2831	0.5164	-0.0042	-0.0742	0.3762	0.0011
18x18	0.2635	0.5892	-0.0095	-0.0602	0.4551	0.002
22x22	0.2071	0.7015	-0.0773	-0.0925	0.4876	0.0174
26x26	0.2092	0.8442	-0.1052	-0.0921	0.5485	0.0199
30x30	0.2067	0.8183	-0.0964	-0.1004	0.5893	0.0156
Average	0.2389	0.6765	-0.0504	-0.0799	0.4853	0.0097

Table 5.4: AUC and CC comparison results for

Support Vector Regression



Method	Itti	GBVS	AIM	Sign.	Center	TSMB-Support Vector Regression					
						10x10	14x14	18x18	22x22	26x26	30x30
AUC	0.67	0.85	0.69	0.67	0.73	0.74	0.74	0.73	0.73	0.73	0.73
CC	0.34	0.49	0.27	0.37	0.44	0.50	0.46	0.45	0.45	0.45	0.46

values therefore are normalized to the scale 0-255, which is the greyscale pixel values. Based on these values, the touch saliency maps are generated using guassian filter to make the maps smooth.

Table 5.4 shows the evaluation results for SVR. Two evaluation metrics AUC and CC are ultilized.

From the results, it can be observed that:

- Our touch saliency learning algorithm TSMB-Support Vector Regression (TSMB-SVR) outperforms the center-based touch saliency learning method (Center). The CC value has been improved from 0.44 to 0.50 and AUC value is also slightly better (from 0.73 to 0.74). On average, the WxH TSMB-SVR approach improved the prediction accuracy by 7.65% over the center-based method. The major reason is that the center-based method only considers room-in/out behavior. Actually, it is found in our study that all the touch behaviors contribute to the human attention.

- The TSMB-SVR also outperforms most of the state-of-the-art prediction methods. The touch saliency map generated by TSMB-SVR has better quality than the saliency map derived by many complex and expensive visual content-based approaches. Although multiple touch behaviors may involve noise, the generated touch saliency map still has high quality and the touch saliency learning approach is much cheaper, faster, and more efficient than visual content-based approaches.

- For TSMB-SVR, it shows that different grid sizes do not make big deference towards the results. Bigger grid sizes do not necessarily mean that the touch saliency maps are closer to the ground truth maps acquired from eye tracking device. Even if the image is roughly divided into 10x10 grids, the performance is very good, and the computational cost is very low. Therefore, 10x10 is a good choice for the TSMB-SVR method.

5.4 Complexity Ananlysis

As the grid size increases, the training samples become very large, the time taken to build model is significantly increased. For example, in the case of Support Vector Regression, the number of samples of 396 images is 39,600 for 10x10 grid size and it

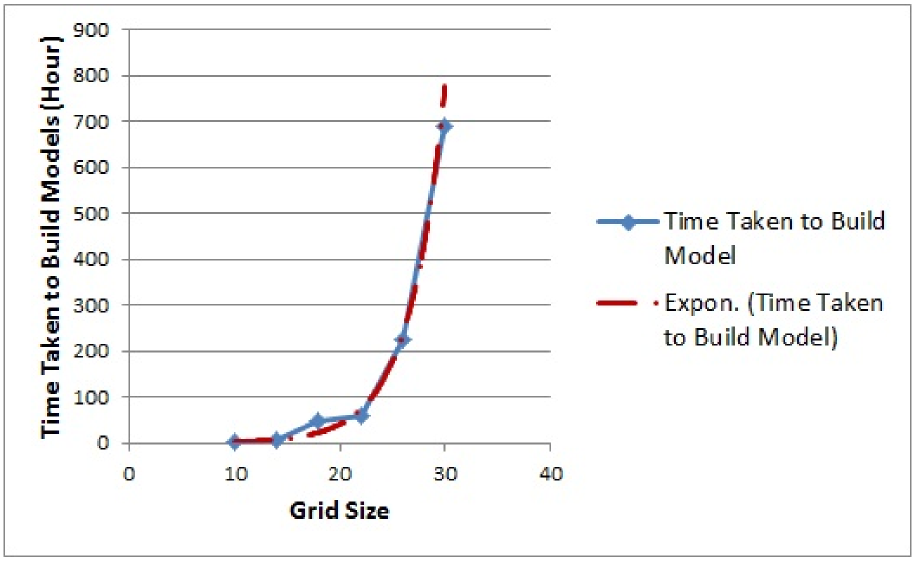


Figure 5.2: Time taken to build the TSMB-SVR models.

will take 1.5 hours to build the model. For 30x30 grid size, the number of samples of 396 images is 1,425,600, time taken to build the model is 223.4 hours. Figure 5.2 shows the time taken to build the model for 396 images of different grid size using SVR. It can be seen that the trend is exponential.

However, the training is offline. Once the model is built, the time to predict the saliency map is trivial (0.0074 second per image for grid size 10x10). Since models are linear functions, the complexity is only O(number of grids). Firgure 5.3 shows the linear relation between the time taken to test and grid size by using SVR. We can see that the trend is linear. Even though the grid size 30x30, it only takes less than 0.06 seconds to predict the saliency values of an image.

On the other hand, the state-of-the-art visual content based methods are based on contrasts of intrinsic image features such as color, texture, semantics, context, orientation, intensity and so on [Cerf et al., 2007]. The models have to

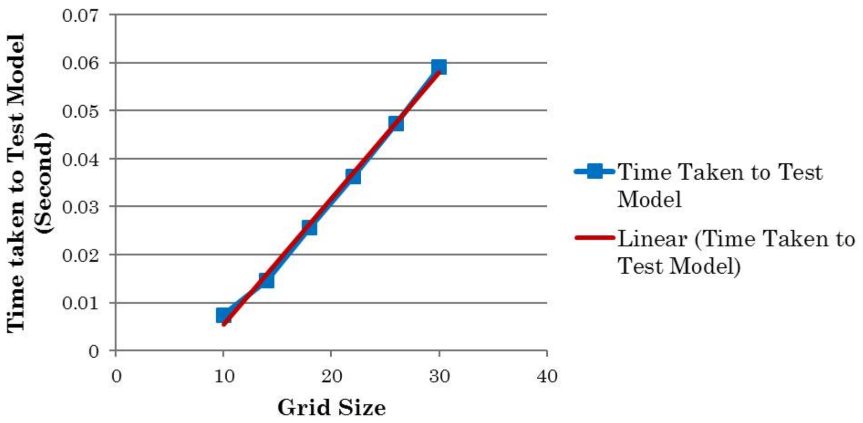


Figure 5.3: Time taken to test an image by using TSMB-SVR models.

extract multiple features first, and then use some algorithm to put them together to estimate the saliency. Therefore, the computational complexity would be very expensive comparing with our regression methods.

CHAPTER VI

CONCLUSION

Visual attention has been a highly popular research area for decades due to its wide applications. There has been a growing interest in the mechanisms of visual attention. Many researchers have developed different approaches to automatically estimate the regions of interest on images by eye tracking or by image visual content. Recently, with the popularity of the touch screen devices (such as touch screen phone, touch pad and so on), users touch behaviors may implicitly express the image saliency when users freely browse the images on limited size touch screen. Therefore, how to learn the image saliency from user touch behaviors becomes a hot topic.

In this thesis work, a quantitative and qualitative study of touch saliency learning from a variety of human touch behaviors such as tap, double tap zoom in/out, scroll (after zoom-in), pinch zoom in/out is conducted. It is learned that different touch behaviors make different contributions to human visual attentions, it is also learned that considering more touch behaviors usually leads to a better touch saliency map.

The experimental results demonstrate the proposed touch saliency learning approach can automatically generate a good saliency map from multiple human touch behaviors. Therefore, our approaches will have wide application potentials where eye tracking is utilized.

In the future, further improvement of the touch saliency performance may be done by applying different learning algorithms such as classification algorithms. Meanwhile, we believe that conducting extensive usability study such as personalized image saliency learning is also promising.

BIBLIOGRAPHY

Bamidele, A., Stentiford, F. W. M., and Morphett, J. (2004). An attention-based approach to content-based image retrieval. *BT Technology Journal*, 22(3):151–160.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Cerf, M., Harel, J., EinhÃďuser, W., and Koch, C. (2007). Predicting human gaze using low-level saliency combined with face detection. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *NIPS*. Curran Associates, Inc.

Foulsham, T. (2008). Saliency and eye movements in the perception of natural scenes.

Goodrich, B. and Arel, I. (2012). Reinforcement learning based visual attention with application to face detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 19–24.

Gunn, S. R. (1998). Support vector machines for classification and regression.

Harel, J., Koch, C., and Perona, P. (2007). Graph-based visual saliency. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 19*, pages 545–552. MIT Press.

Hou, X., Harel, J., and Koch, C. (2012). Image signature: Highlighting sparse salient regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):194–201.

Hua, Z., Li, Y., and Li, J. (2010). Image segmentation algorithm based on improved visual attention model and region growing. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259.

Jian Li, Martin Levine, X. A. and He, H. (2011). Saliency detection based on frequency and spatial domain analyses. In *Proceedings of the British Machine Vision Conference*, pages 86.1–86.11. BMVA Press. http://dx.doi.org/10.5244/C.25.86.

Judd, T., Durand, F., and Torralba, A. (2012). A benchmark of computational models of saliency to predict human fixations.

Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2106–2113.

LEAD (2010). Sr research eyelink 1000.

Marques, O., Mayron, L. M., Borba, G. B., and Gamba, H. R. (2006). Using visual attention to extract regions of interest in the context of image retrieval. In *Proceedings of the 44th annual Southeast regional conference*, ACM-SE 44, pages 638–643, New York, NY, USA. ACM.

Oliva, A., Torralba, A., Castelhano, M., and Henderson, J. (2003). Top-down control of visual attention in object detection. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–253–6 vol.1.

Ouerhani, N., von Wartburg, R., Hugli, H., and Muri, R. (2003). Empirical validation of the saliency-based model of visual attention. *Electronic Letters on Computer Vision and Image Analysis*, 3(1).

Paisitkriangkrai, P. (2012). Linear regression and support vector regression. University Lecture.

Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., and Chua, T.-S. (2010). An eye fixation database for saliency detection in images. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 30–43, Berlin, Heidelberg. Springer-Verlag.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel Methods in Computational Biology*. Computational Molecular Biology. MIT Press, Cambridge, MA, USA.

Setlur, V., Takagi, S., Raskar, R., Gleicher, M., and Gooch, B. (2005). Automatic image retargeting. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, MUM '05, pages 59–68, New York, NY, USA. ACM.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.

Tsotsos, J. K. and Bruce, N. D. B. (2006). Saliency based on information maximization. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 155–162, MIT Press. MIT Press.

Vapnik, V. (1997). The support vector method. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, ICANN '97, pages 263–271, London, UK, UK. Springer-Verlag.

Xie, X., Liu, H., Goumaz, S., and Ma, W.-Y. (2005). Learning user interest for image browsing on small-form-factor devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 671–680, New York, NY, USA. ACM.

Xu, M., Ni, B., Dong, J., Huang, Z., Wang, M., and Yan, S. (2012). Touch saliency. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1041–1044, New York, NY, USA. ACM.

Yang, K.-C., Guest, C., and Das, P. (2006). Human visual attention map for compressed video. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pages 525–532.

VITA

Shaomin Fang was born in Yunnan Province, China, on October 3rd, 1985, the daughter of Chengchun Fang and Zefei Wen. She worked in Electronic Field for 3 years after she graduated from Tianjin Polytechnic University in China in 2005, with a bachelor's degree in Electrical Engineering. Then she moved to United States. In fall 2011, she entered Texas State University-San Marcos to pursue a master's degree in Computer Science.

Permanent Address: 2159 Sid Allens Dr.
Buda, TX 78610

This thesis was typed by Shaomin Fang.