A Collaborative Benchmark for Region of Interest Detection Algorithms

Tz-Huan Huang Kai-Yin Cheng Yung-Yu Chuang
National Taiwan University
http://www.cmlab.csie.ntu.edu.tw/roi

Abstract

This paper presents a collaborative benchmark for region of interest (ROI) detection in images. ROI detection has many useful applications and many algorithms have been proposed to automatically detect ROIs. Unfortunately, due to the lack of benchmarks, these methods were often tested on small data sets that are not available to others, making fair comparisons of these methods difficult. Examples from many fields have shown that repeatable experiments using published benchmarks are crucial to the fast advancement of the fields. To fill the gap, this paper presents our design for a collaborative game, called Photoshoot, to collect human ROI annotations for constructing an ROI benchmark. Using this game, we have gathered a large number of annotations and fused them into aggregated ROI models. With these models, we are able to evaluate six ROI detection algorithms quantitatively.

1. Introduction

Attention plays an important role in human vision. For example, when we look at an image, our eye movements comprise a succession of *fixations* (repetitive positioning of eyes to parts of the image) and *saccades* (rapid eye jump). Those parts of the image that cause eye fixations and capture primary attention are called *regions of interest* (ROIs). Studies in visual attention and eye movement have shown that humans generally only attend to a few ROIs. Detecting these visually attentive regions in images is challenging but useful in many applications.

Many algorithms have been proposed for automatic ROI detection in images. Unfortunately, these methods were often evaluated only on specific and small data sets that are not publicly available. The lack of published *benchmarks* makes experiments non-repeatable and quantitative evaluation difficult. In many fields, there is a growing trend towards the use of common benchmarks for evaluation and development of algorithms. Notable examples include Caltech-256 for object categorization and Middlebury dataset for stereo vision. These benchmarks allow researchers to compare their method with other algorithms quantitatively and to identify the main factors that affect performance, thus stimulating an even faster performance improvement. Furthermore, the most promising approaches in these fields often rely on machine learning. The success of learning-based approaches, however, is heavily dependent upon the training data. Benchmarks with ground truth or annotations are crucial assets for such approaches.

There are two common ways to obtain benchmarks, *measuring* and *labeling*. Measuring is often used in the situations when there is a true answer and there is apparatus to capture the ground truth. On the other hand, labeling is used when there is no available equipment to obtain the true answer, face detection for example, or when there is potential ambiguity in the answer to the problem, such as semantic concept detection. For ROI detection, although it is possible to employ a measuring approach to construct benchmarks using eye trackers, we choose to use a labeling approach to have participants annotate ROIs by hand. We prefer the labeling approach mainly because we need a large number of annotations. It is not only because we need many images in the benchmark, but also because each image must be annotated by many different people. It has been suggested that ROIs could vary according to personal and application perspectives. However, despite of differences, evidences also indicate that there are lots of similarities among personal ROIs [8, 10]. For example, Wooding found that the difference in regions fixated by two groups is less than 4% on average [17]. Thus, we believe that the aggregated ROI from many people should provide a useful reference. Unfortunately, eye trackers are still too expensive to be widely deployed to collect a large number of annotations.

To collect a large number of human annotations from many people, inspired by recent success of collaborative games [15], we follow a similar approach and develop an online game called *Photoshoot* for our ROI benchmark construction. In Photoshoot, paired online players take turns of “target” and “shoot” roles. The target player places targets on an image displayed at both players’ sites. Without seeing the placed targets, the shoot player attempts to find the targets by shooting at the image, essentially clicking on several locations on the image. To maximize score, players tend to target and shoot at visually prominent parts of the image. By playing this game, people help us annotate ROIs within images. These annotations are fused to form an aggregated ROI model for each image. The benchmark is then used for quantitative comparisons for ROI detection algorithms.

Liu *et al*. also collected a set of images with human-annotated ROIs for developing their learning-based ROI detection algorithm [5]. They also compared their algorithm with two other algorithms. The main differences of our work from theirs are: (1) Their benchmark assumes that there is only a single ROI region in each image while ours does not have such an assumption. Many real-world images do have multiple ROIs and it is important to evaluate ROI algorithms' capability on detecting multiple ROIs. (2) Their benchmark is annotated by only few people while ours are annotated by many. As ROI is subjective, we believe that it is important to be annotated by many people. (3) Our evaluation is more complete, including more algorithms and a more thorough evaluation.

This paper offers to the research community a large collection of manually annotated ROI benchmarks and a uniform procedure for quantitatively evaluating ROI detection algorithms. Specifically, our main contributions are: (1) a publicly available large collection of ROI-annotated images, (2) an evaluation methodology for ROI detection algorithms, (3) the most complete quantitative evaluation, to date, for ROI detection algorithms, and (4) a game design for collecting ROI annotations.

2. Related work

ROI detection algorithms. Bottom-up methods assume that human eyes can rapidly skirt across the entire image and select small areas that are different from the surroundings. Itti's method [4] is probably the most popular example. For an input image, it first computes a saliency map from pyramid maps for color, luminance, and orientation contrasts. A winner-take-all strategy is used to select the most salient locations in order of decreasing saliency. Osberger and Maeder used a weighted combination of a region's low-level features such as contrast, size, shape and location to determine that region's importance value [8]. Ma and Zhang proposed a method to locate ROIs as areas of high color contrasts [6]. Fuzzy growing is used to find the salient areas in the image. Hu *et al*. used generalized principal component analysis (GPCA) method for finding ROIs [3]. The values of intensity, colors and hue are first transformed into the first quadrant of the polar coordinate system. GPCA is then used to identify dominant components as salient regions. Even if bottom-up approaches imitate our eye activities very well, their main defect is that they do not emphasize highly semantic areas, such as human faces, in an image.

Top-down methods, on the other hand, assume that people pay more attention to areas corresponding to semantic objects even if those areas are not salient compared with its surrounding [9]. To emphasize semantic objects, in addition to low-level features, top-down approaches often augment saliency map calculation with semantic object detection such as face detection and text detection. Ma and Zhang's fuzzy growing method optionally includes a face detection module to improve the calculation of attended areas [6]. Chen *et al*. weighted together saliency attention model, face attention model and text attention model to form an aggregated image attention model [1]. A rule-based approach is used to adjust weights. Different rules are applied to images of different categories.

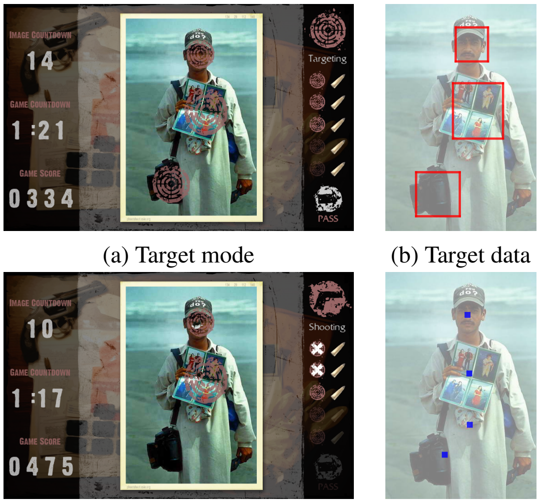
Liu *et al*. proposed the first learning-based ROI detection algorithm [5]. They used conditional random field as the learning framework on three features, multi-scale contrast, center-surround histogram and color spatial-distribution.

Collaborative labeling. Because of demands for training sets, many collaborative labeling tools for different applications have been developed. Russell *et al*. designed a web-based tool called LabelMe for users to extract and annotate objects in images for object categorization [11]. For LabelMe, the incentive to annotate is to obtain the whole annotation data. These systems often only attract a few volunteers or researchers. To harness more human power, von Ahn is the pioneer to propose the idea of using online games to attract more people to help [15]. He has designed ESP game for tagging images [15], Peekaboom for locating objects [16], and some other games.

3. Photoshoot

Game design. Although our framework of using collaborative games is similar to ESP, designing such a game for solving a different problem is actually not easier than designing an algorithm [15]. As pointed out by von Ahn, any good game design to solve a problem must simultaneously satisfy two requirements: it ensures reasonable solution to the problem to be solved and it is fun to play [15].

Photoshoot randomly pairs two online users. The paired players do not know the identity of their partner and they do not have any way of communicating other than an image they can both see. In Photoshoot, players are assigned the roles of "target" and "shoot" in turns. In a round, a same image is presented to both target and shoot players. The target player places targets on the image by drawing rectangles over the image using drag-and-drop (Figure 1(a)). Up to five targets can be placed. Without seeing the targets placed by the target player, the shoot player's role is to guess where those targets are by shooting at them by clicking on the image (Figure 1(c)). Similarly, at most five bullets can be fired. Both players receive a certain number of points for each agreement of a bullet and a target. The targets that



(a) Target mode (b) Target data

(c) Shoot mode (d) Shoot data

Figure 1. Photoshoot. On the left are the screen captures of a target player (a) and a shoot player (c). Note that they were captured during the course of playing. At the moment of capturing, the target player had placed three targets and the shoot player had shot two of them. The shot targets were displayed on the shoot player's screen while un-shot ones were not. One the right are the target data (b) and shoot data (d) we collected at the end of that round.

have been shot are displayed and any further bullet shot on those targets gives no point. For each round, players have at most 15 seconds to either place targets or shoot bullets. The players move on to the next round for a new image if time is up, all targets have been shot or all bullets are used up. Players can also pass on the images that they feel difficult to make decisions. The players switch roles in the next round. They have a total of three minutes to go through as many images as possible.

From game's perspective, the goal is to guess where your partners will place targets or shoot bullets to maximize score. To increase the chance of agreement, since there is no way for players to communicate, the easiest way for both players to agree is to select the more prominent areas in the image. It turns out the regions on which the two players agree are typically the salient regions.

For our game to be successful, it is important to encourage players to continue playing by providing them with points for agreements between targets and bullets. Although the exact number of points is not important, for the game's purpose, the score rules must be able to encourage accurate annotations. Thus, the score gained are inversely proportional to the size of the target to avoid players setting extremely large targets to increase the chance of being shot. To optimize score, players have incentive to choose proper sizes for targets and good locations for bullets. For target placement, larger targets increase chance of being hit, but lead to lower score. On the other hand, smaller targets are less likely to be shot, but could gain more points if they are. Hence, the best strategy for target placement is to make targets just large enough to contain the more prominent areas. For shooting, the score is inversely proportional to the distance between the position of the bullet and the center of the target being shot. Therefore, the optimal shooting strategy is to shoot at the centers of ROIs. By employing these score rules, the game ensures that the optimal player's strategy is the optimal behavior we expect from annotators, leading to more accurate annotations. Furthermore, we encourage players to select the most prominent areas first by adding bonus if the orders of target placement and bullet shooting match better. In addition, points are not subtracted for passing to avoid noisy annotations for difficult cases.

Implementation. There are a total of 3,000 images in the database collected from the web and our colleagues. Currently, we only use photographs. We do not employ a larger database because our goal is to collect a benchmark instead of attempting to annotate all images of the world. Therefore, we want to obtain the aggregated behaviors by having as many annotations as possible for each image.

We carefully choose parameters in the game. For example, each round has a time limit of 15 seconds. It is because experiments suggest that viewers tend to be distracted if the duration extends beyond 15 seconds [7]. The limits on the numbers of targets and bullets are set because there are usually only a small number of ROIs in an image. For example, Privitera *et al*. reported an average of seven ROIs per image according to their experiments [10]. We set a lower number at five so that the game is more enjoyable. This limit is compensated when user annotations are aggregated into ROI models in Section 4.

Similar to ESP, to avoid having the odd-number player wait for too long in the queue, a bot can play with a player with a pre-recorded set of actions from a real player [15]. Human players' actions are still recorded even if they play with the bots. As ESP, cheating prevention is necessary. When players connect to our server, they are not immediately paired to avoid cheating by logging in at the same time. Since players have no way for communication between them, it is unlikely for players to cheat by setting up strategies during the game. However, they could use some pre-defined protocols to cheat. For example, they could decide to place targets and shoot bullets at the center and four corners. We check for several unusual patterns and put cheaters into a blacklist. Overall, we observed very few cheats. It is because there is no fun to cheat. In addition, since we aggregate data from multiple players, we can detect cheats during off-line analysis described in Section 4.

Statistics. During the one-month period after Photoshoot's deployment, there are a total of 1,002 users who have played at least once. Among these people, 71% of them played more than once and 36% played more than four sessions. Furthermore, 20 people played more than 60 times (3 hours of playing) and some have spent more than 10 hours playing this game. The number is not as amazing as ESP's. However, it is good enough for us to construct a useful benchmark since we only intend to annotate 3,000 images. From this one-month period, we have collected a total of 134,646 targets and 168,352 shoots. Figure 1(b) shows an example of target data collected from a player and Figure 1(d) shows an example for shoot data for a round.

4. ROI modeling

For each image I, we have a set of target data, rectangles, and a set of shoot data, points. Unlike ESP, in our case, users' inputs can't be directly used as answers to the problem to be solved. Instead, we have to fuse users' annotations together and convert the result into more popular ROI formats. This is the nature of ROI and not unique to labeling-based approaches. For measuring-based approaches such as eye tracking, post-processing and fusion are also required to convert measurements into more useful formats.

There are several common formats to represent ROIs. Different formats could be useful for different applications. Conversions between formats are possible although there is no definite way for conversions.

An importance map. It is also called a saliency map. In such a map, each pixel records an importance score for the degree of being a part of ROIs. With proper normalization, an importance map can also be interpreted as a probability density distribution of belonging to an ROI for each pixel. An importance map is a more general format since it can be easily converted into other formats. Many algorithms output this form, at least as an intermediate result [4, 6].

A binary ROI mask. Each pixel in the mask records whether it belongs to part of ROIs. A few algorithms can only output this format [3].

A set of focus of attention (FOA) [4]. ROIs are represented as an unordered set or an ordered list of a handful of attended points [6].

The remaining of this section introduces methods for aggregating data from multiple players. We have two types of data, target and shoot. They are converted separately into different formats. Although combined separately, they use each other for verification.

4.1. Target ROI model

Each target t_i is a rectangle with a center (x_i, y_i), a width w_i and a height h_i. Thus, each t_i defines a Gaussian G_i centering at (x_i, y_i) with the standard deviations related to w_i and h_i by a scaler factor s. We set s as $\frac{1}{3}$ in our experiments. Each Gaussian can be regarded as an importance map by itself. In principle, we could form an importance map simply by superimposing all Gaussian functions G_i together. However, as mentioned earlier, the collected data might be polluted by cheats or annotation noises. For constructing a more reliable model, we use Random Sample Consensus (RANSAC) algorithm to remove outliers.

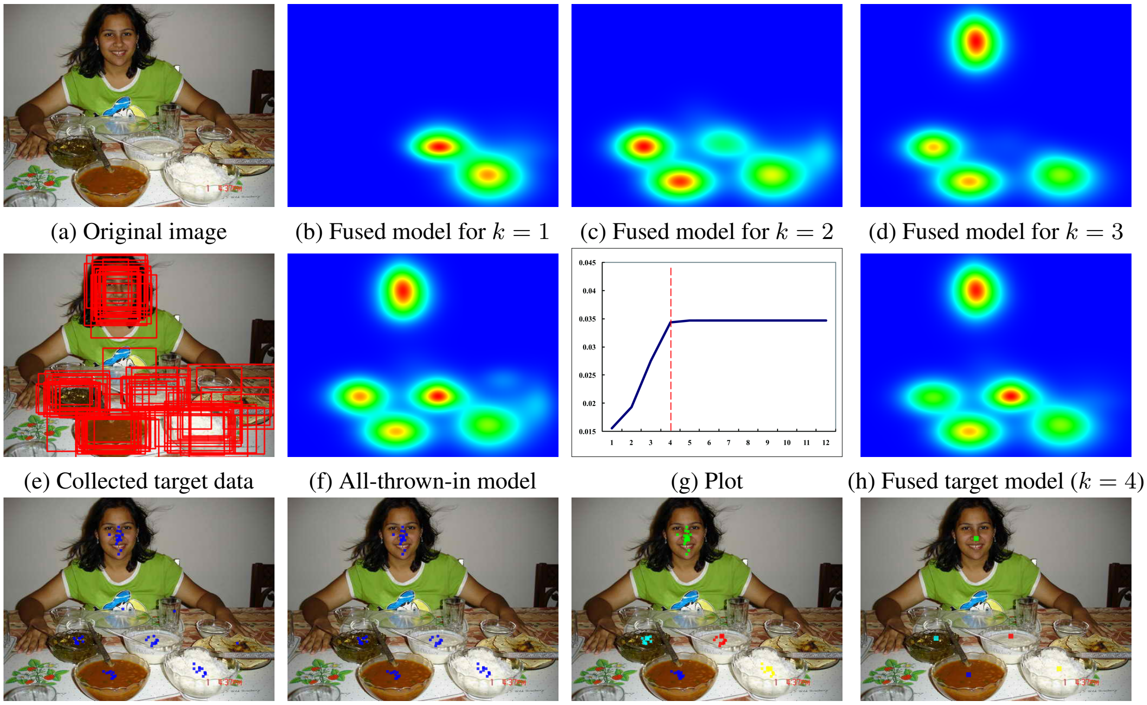
If we already know there are k ROIs in the image, then the important map should have k modes. Thus, we can form a hypothesis by randomly selecting k Gaussians from G_i's as hypothetical inliers. A hypothetical model is then formed by superimposing these k Gaussians. All other Gaussians are tested against the hypothetical model and the number of their agreements with the hypothesis is counted. If there are sufficiently many inliers, then the hypothetical model should be reasonably good. This process is iterated several times and the hypothesis with most agreements is selected. Finally, all targets regarded as inliers to the selected hypothesis are fused together to form the final importance map. We also have to define a metric to evaluate the distance between two importance maps for inlier verification. Since importance maps can be regarded as probability density distributions, methods such as KL divergence and intersection could be used. We found that intersection leads to more stable results and used it in our implementation.

The main question that remains is to determine a proper number of ROIs, k. This is essentially a model selection problem and there are several possible solutions such as minimum description length. We opt for an easier but more effective solution, using hold-out set to select the model with a minimum generalization error. The main drawback of using hold-out set is the waste of precious data. However, we can use shoot data as the hold-out set to avoid the waste. Hence, we enumerate k from 1 to 12, find the best model for each k using RANSAC and select the one that best explains the shoot data. We choose 12 as the upper bound since it is suggested that an image usually has less than 11 ROIs [10].

Figure 2 gives an example for aggregating target data. Figure 2(a) is the original image and Figure 2(e) shows all target data. Figure 2(f) displays the aggregated model by throwing in all target data without outlier removal. Thus, it could have Extra ROIs due to outliers. In this map, warm colors mean higher values and cold color represents smaller ones. Figure 2(b,c,d) shows the fused models for $k = 1, 2, 3$. Figure 2(g) lists the performance for models of different k. Figure 2(h) shows the final aggregated target model, selected when $k = 4$. Note that outliers are removed and the number of ROIs does match our intuition.

4.2. Shoot ROI model

The nature of shoot data is similar to eye fixation measurements from eye trackers. It is possible to fuse shoot data in the same manner as fusing target data by artificially augmenting fixed widths. Here, instead, we choose to follow a more popular approach of clustering to convert shoot



(a) Original image (b) Fused model for $k = 1$ (c) Fused model for $k = 2$ (d) Fused model for $k = 3$

(e) Collected target data (f) All-thrown-in model (g) Plot (h) Fused target model ($k = 4$)

(i) Collected shoot data (j) Shoot data without outliers (k) Clustering (l) Fused shoot model

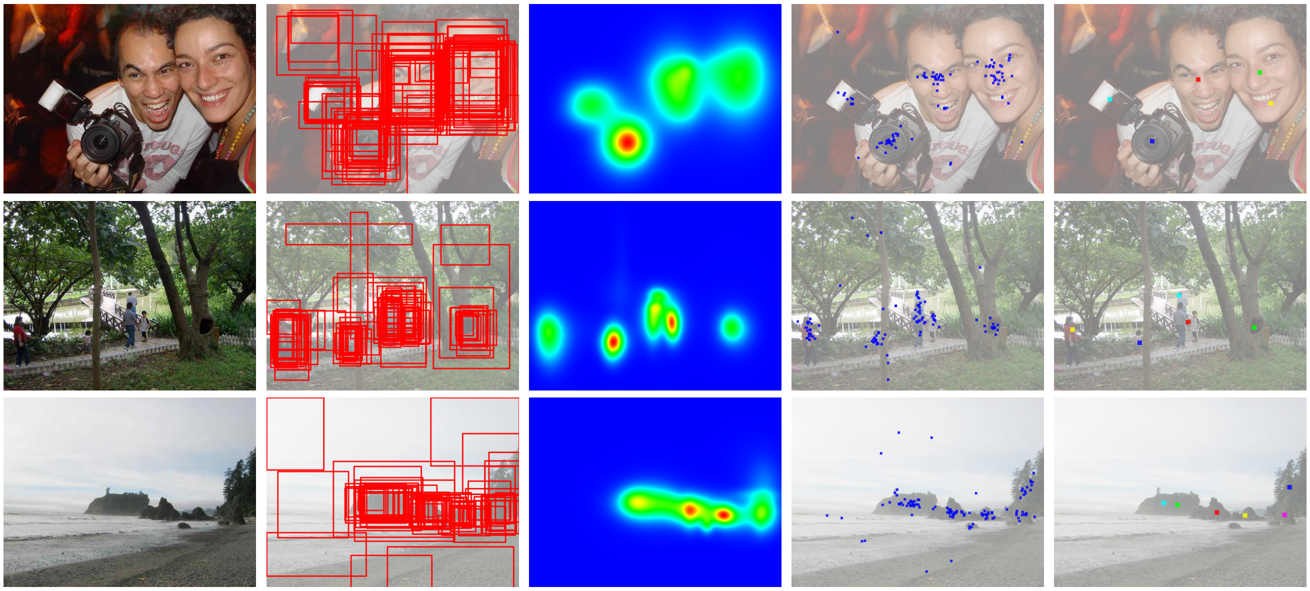
Figure 2. ROI Model fusion. Given the original image (a), and the collected target data (e), the all-thrown-in model (f) could include outliers. We use RANSAC to remove outliers for each specific number of ROIs, k. Models for different k are displayed in (b), (c) and (d). To determine the right number of ROIs, the fitness of models of different k are plotted (g) and the best one (h) when $k = 4$ is selected. For the given shoot data (i), outliers are first removed using the target ROI model (h). The points that remain are then clustered using affinity propagation (k). One attended point is selected for each cluster to form a set of FOA as the shoot ROI model (l).

data into a set of FOA [10, 13, 7]. As specified by Santella and DeCarlo [13], a algorithm for clustering eye movement data should hold the following three desirable characteristics: ability to produce consistent results without regard to initialization, no need to know the number of clusters in advance and robustness to outliers. We use a recently proposed clustering algorithm, affinity propagation [2], for clustering. It satisfies first two requirements. It has some resistance to outliers as it would often create extra clusters to contain outliers. These clusters of outliers can be easily detected and removed. However, in our implementation, we use the fused target ROI model to remove outlier first and then feed the trimmed list into affinity propagation for clustering. This is analogous to only keep the bullets that hit the target set up by the fused target ROI model. The result of affinity propagation assigns each cluster with a representative point. Thus, these representative points together form a set of FOA, our shoot ROI model. Optionally, each cluster can be fitted into a Gaussian to become an importance map.

Figure 2(i) shows all collected shoot data for Figure 2(a). After removing outliers using the model in Figure 2(h), we have points in Figure 2(j). These points are then clustered into Figure 2(k). Figure 2(l) displays the final shoot model of three attended points.

4.3. Results

Figure 3 shows several examples of the collected data and the aggregated ROI models. Figure 4(a-c) shows more examples for fused ROI models. Subjectively, most match with our intuition for ROIs very well. Note that, even if our game sets a maximum of 5 targets/bullets at a time, the resulted ROI models are not necessarily restricted by this number. The aggregated behavior of players tends to recover all ROIs. See the third example in Figure 4 for example. Similarly, although players can only draw rectangles, The last example of Figure 4 shows that the recovered ROI can match the shape of the object very well. On average, 90% of annotations are regarded as inliers by our algorithms. We also validated our benchmark on a small dataset of 30 images by comparing to ROIs labeled by volunteers. The precision and recall of game-collected ROIs against volunteer-labeled ROIs are 0.91 and 0.90 respectively. In addition, we compared the game-collected ROIs to ROIs from eye tracking data. The precision and recall are 0.85 and 0.87 respectively. These show that annotated ROIs are very close to fixation ROIs although there is probably slight variation between them. Furthermore, in some applications, the annotated ROIs are probably sufficient. For ex-



(a) Original image (b) Collected target data (c) Fused target model (d) Collected shoot data (e) Fused shoot model

Figure 3. Examples of the collected data and fused ROI models.

ample, image cropping cares more about the regions which users "consider" important. Thus, there is value in benchmarks built around both kinds of ROIs.

From this large set of human annotations, we have made several observations and some of them confirm observations from others. First, more attention is drawn into semantic objects, especially human faces, even if they are of low contrasts or of small sizes. Other examples include texts, gestures and unusual behaviors. This indicates the needs of including higher-level visual components into ROI detection. Second, center of an image tends to gain more attention. This is caused by both the way photographers take pictures and the way we look at an image. This suggests adding spatial priors for ROI detection. Third, even if they are not distinctive in low-level features, vanishing points often get more attention in landscape photographs. Finally, if a ROI is large in size, then it is often further divided into multiple ROIs. In addition, viewers only fixated on a particular region of an object but not necessarily the whole object. For example, we could only pay attention of the head of a horse. Similar observations have been made by Nguyen *et al.* [7].

5. Evaluation

Our current evaluation implemented six methods, Itti's method [4], Fuzzy growing [6], Osberger's method [8], MSRA's method [5], Robust GPCA [3] and Threshold selection [12]. Note that these methods often output different ROI formats and some even output multiple formats. For example, Itti's method can output importance maps or convert it to FOA set using a winner-take-all (WTA) neural network while robust GPCA can only output binary ROI masks. Fuzzy growing can output any of three ROI formats in Section 4, contrast-based saliency maps as importance maps, attended view as binary ROI masks and attended points as FOA sets [6]. Overall, the evaluation on importance maps gives a better indication. Unfortunately, not all algorithms can output importance maps. Thus, for evaluation, we have selected different metrics for different ROI formats*. Figure 4 shows examples for original images, ROI models in the benchmark and outputs of the evaluated algorithms.

	Itti's method	Fuzzy growing	Osberger's method	MSRA's method
L_2 error	0.0053	0.0053	0.0053	**0.0051**
KL divergence	0.5683	0.5435	0.5641	**0.4161**
Intersection	0.3601	0.3879	0.3696	**0.4921**
Rank correlation	0.1795	0.3583	0.3242	**0.5784**

Table 1. Evaluation of importance maps. (The best ones are bold.)

	Robust GPCA	Threshold selection
Precision	0.0868	0.2214
Recall	0.1544	0.3834
MAP	0.1941	0.4131

Table 2. Evaluation of binary ROI maks.

	Itti's method	Fuzzy growing
Precision	0.4475	0.4506
Recall	0.5515	0.5542

Table 3. Evaluation of FOA sets.

*Note that we do not necessarily evaluate all possible output formats for the algorithms we have implemented.

Importance maps. It is the most general form and could be converted into the other two using methods such as WTA or fuzzy growing. We report the results for L_2 error, KL divergence and intersection [14]. Sometimes, it is the relative rank but not the absolute value that really matters. Thus, we also report results for Spearman's rank correlation.The outputs of Itti's method, Fuzzy growing, Osberger's method and MSRA's method are compared against the target ROI models. Table 1 reports the results aggregated from all images in the benchmark. Note that a better method should have a smaller value for L_2 error and KL divergence, and a larger value for intersection and rank correlation. For all measurements, MSRA's method consistently outperforms other methods by a large margin. This may not be a surprise since that method is learning-based. This indicates that learning-based methods have the potential to give better performance. Our benchmark could also serve as the training set of such methods.

Binary ROI masks. There is no such form in our benchmark. It is possible to convert the important maps of the target ROI models using thresholding or other methods. We select a proper threshold and calculate the precision/recall values for GPCA and Threshold selection. In addition, to relieve the impact of the threshold, we also calculate the mean average precision (MAP) by treating algorithm output as ground truth and using it to evaluate the correspondence importance map in the benchmark. This metric tells us the correlation between the algorithm outputs and the benchmark references. Table 2 reports the aggregated results for Robust GPCA and Threshold selection.

Focus of attention sets. Similar to previous work [10], we use precision and recall values. Shoot ROI models in the benchmark are used as ground truth. Two FOAs are considered coincided if their distance is less than $\frac{1}{12}$ of the smaller of image width and height. Table 3 reports the aggregated results for Itti's method and Fuzzy growing. Importance maps are converted into FOA sets using a WTA network. In general, Fuzzy growing is better. The results are consistent to the results for importance maps.

6. Conclusion and future work

In this paper, we have presented a benchmark for ROI research. Our benchmark is both subjective and objective in nature. It is subjective because it ia annotated by human. It is objective because we aggregate large number of annotations to obtain the average annotation. Because of the nature of ROI, we will not say that our benchmark is precise. However, we believe that it does provide a good reference and is better than other available options. In the future, we plan to develop a taxonomy and categorization scheme for ROI detection algorithms. Such a systematic analysis of existing ROI detection algorithms could lead to better understanding and further improvement to ROI detection.

Acknowledgments

The authors would like to thank reviewers for their helpful suggestions. This work was supported by grants NSC97-2622-E-002-010-CC2 and NTU-98R0062-04.

References

[1] L. Q. Chen, X. Xie, X. Fan, W. Y. Ma, H. J. Zhang, and H. Q. Zhou. A visual attention model for adapting images on small displays. In *ACM Multimedia System Journal*, volume 9, 2003.

[2] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[3] Y. Hu, D. Rajan, and L.-T. Chia. Robust subspace analysis for detecting visual attention regions in images. In *ACM Multimedia 2005*, pages 716–724, 2005.

[4] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.

[5] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *Proceedings of CVPR 2007*, 2007.

[6] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of ACM Multimedia 2003*, pages 374–381, 2003.

[7] A. Nguyen, V. Chandran, and S. Sridharan. Gaze tracking for region of interest coding in JPEG 2000. *Signal Processing: Image Communication*, 21(5):359–377, 2006.

[8] W. Osberger and A. J. Maeder. Automatic identification of perceptually important regions in an image. In *Proceedings of ICPR 1998*, pages 701–704, 1998.

[9] R. Parasuraman. *The Attentive Brain*. The MIT Press.

[10] C. M. Privitera and L. W. Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 22(9):970–982, 2000.

[11] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *MIT AI Lab Memo AIM-2005-025*, 2005.

[12] P. K. Sahoo, D. W. Slaaf, and T. A. Albert. Threshold selection using a minimal histogram entropy difference. *Optical Engineering*, 36:1976–1981, July 1997.

[13] A. Santella and D. DeCarlo. Robust clustering of eye movement recordings for quantification of visual interest. In *Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 27–34, 2004.

[14] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[15] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of ACM SIGCHI 2004*, pages 319–326, 2004.

[16] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of ACM SIGCHI 2006*, pages 55–64, 2006.

[17] D. S. Wooding. Fixation maps: quantifying eye-movement traces. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 31–36, 2002.

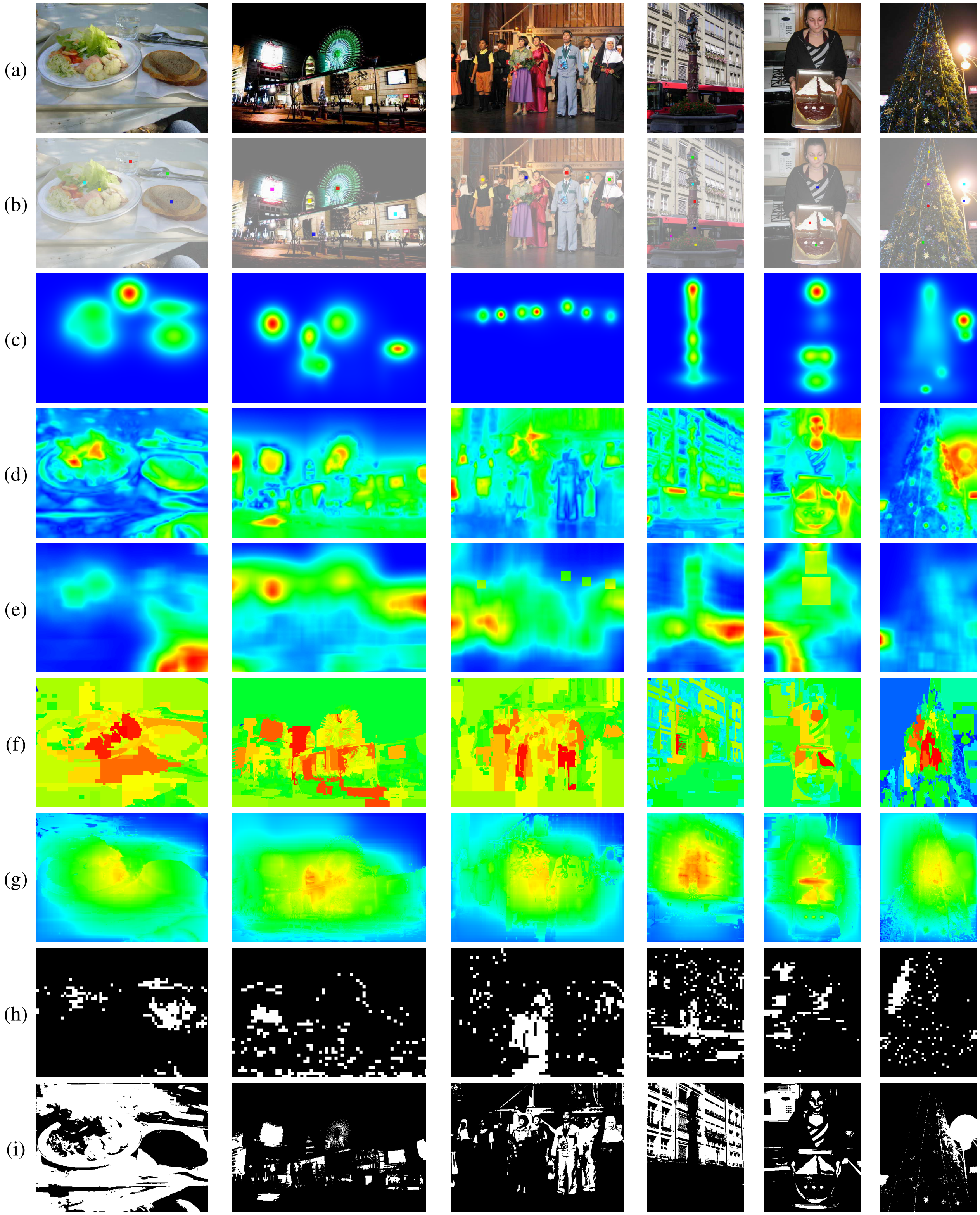


Figure 4. Examples for ROI models and algorithm outputs. (a) Original image. (b) Shoot ROI model. (c) Target ROI model. (d) Importance maps from Itti's method. (e) Importance maps from Fuzzy growing. (f) Importance maps from Osberger's method. (g) Importance maps from MSRA's method. (h) Binary ROI masks from Robust GPCA. (i) Binary ROI masks from threshold selection.