SmartOverlays: A Visual Saliency Driven Label Placement for Intelligent Human-Computer Interfaces

Srinidhi Hegde Jitender Maurya Ramya Hebbalaguppe
TCS Innovation Labs, New Delhi
{sri.hegde, jitender.maurya, ramya.hebbalaguppe}@tcs.com

Figure 1: Representative labels generated by *SmartOverlays*: (a) and (c) are the output of the YOLOv2 [10] object detector; (b) and (d) depict the outcome from our *SmartOverlays* algorithm. Note that labels are closer to the objects of interest, do not overlap each other, and do not occlude salient regions in a scene; (e) depicts a cluttered surveillance scene; (f) and (g) show outdoor and indoor sports applications respectively; (h) depicts a comic style label placement feature.

Abstract

In augmented reality (AR), the computer generated labels assist in understanding a scene by addition of contextual information. However, naive label placement often results in clutter and occlusion impairing the effectiveness of AR visualization. For label placement, the main objectives to be satisfied are, non occlusion to scene of interest, the proximity of labels to the object, and, temporally coherent labels in a video/live feed. We present a novel method for the placement of labels corresponding to objects of interest in a video/live feed that satisfies the aforementioned objectives. Our proposed framework, ***SmartOverlays*** [1]*, first identifies the objects and generates corresponding labels using a YOLOv2 [10] in a video frame; at the same time, Saliency Attention Model (SAM) [3] learns eye fixation points that aid in predicting saliency maps for label placement; finally, computes Voronoi partitions of the video frame, choosing the centroids of objects as seed points, to place labels for satisfying the proximity constraints with the object of interest. In addition, our approach incorporates tracking the detected objects in a frame to facilitate temporal coherence between frames that enhances readability of labels. We measure the effectiveness of SmartOverlays framework using two objective metrics: (a)* Label Occlusion over Saliency (LOS)*, and, (b) temporal jitter metric to quantify jitter in the label placement.*

1. Introduction

Augmented Reality applications fuse contextual synthetic data with the real visual data to enrich perception and efficiency of the user performing a targeted task. Such contextual data superimposed on live video feed is referred as overlays. The overlays can take the forms of, but not limited to, text, audio, 3D objects and GPS coordinates. In this work, we propose *SmartOverlays*, a generic label placement framework that enhances the effectiveness of situated visualization across most of the hand-held/head mounted AR applications. SmartOverlays can be a great addition to low field of view devices like head mounts, where any obstruc-

[1] Refer https://ilab-ar.github.io/SmartOverlays/ for more details and demo video.

tion in an already constrained environment can greatly hamper the immersive experience.

There are certain key challenges in placing labels in AR. Overlays could take a variety of geometric shapes/sizes as per application specifications. Furthermore, the number of possible label positions grows exponentially with the number of items to be labelled, making the problem NP-Hard [1].

Previous works employed either sensor information such as GPS information, or used cues from the overlay features such as configuration of fixed anchor points [5, 12] as a criteria to place labels. Eye tracking for AR can be utilised for label placement to know a suitable position of the label. In this work, we consider eye gazes as the visually salient part of the scene as it is an inherent feature of the image aiding in understanding the dynamics of the scene. Thus, we incorporate Saliency Attention Model [3] as a part of label placement model in combination with image aesthetics to reduce the strain on labels overlayed. Our work does not focus on generating meaningful caption which is an open problem in computer vision community, neither do we address label content based on scene understanding of an image. However, these aspects can be combined with the proposed *SmartOverlays* algorithm, to deliver promising artificial intelligence applications. Key contributions of our work are: (a) We propose *SmartOverlays*, a multi-label placement framework on video frames/live feed utilizing object cues from object detector and visual saliency. (b) We ensure temporal coherence in placed labels using tracking based methods. (c) We introduce two metrics, *Label Occlusion over Saliency score (LOS)* and *Temporal Jitter* metric, for measuring the effectiveness of overlay placement spatially and temporally.

2. Proposed Method

We formulate the label placement problem as follows: the input for our pipeline is an RGB video $V < f_1 , f_2 , \dots , f_n >$ with frame sequence of length n and each frame of dimension $F_w \times F_h$. Our proposed model outputs an image coordinate $P = (x^i, y^i)$, for the i^{th} label in a frame, where $1 \leq x^i \leq F_w$ and $1 \leq y^i \leq F_h$. P represents the most suitable coordinate in the frame space for placing the i^{th} overlay. This point corresponds to the top left corner of the overlay or overlay bounding box if the overlay is non rectangular. In case of unconstrained overlays, we consider the tightest bounding box surrounding the overlay. Figure 2 shows an overview of our *SmartOverlay* algorithm.

2.1. Object Detection and Saliency Map Computation

For multiple label placement, it is necessary to have correspondences of labels with object of interest; the labels as placed as close to the relevant objects as possible. Hence, we detect and label objects in the input video frames using a YOLOv2[10] object detector.

We then use Saliency Attention Model (SAM) proposed by Cornia et al.[3], for computing saliency maps. SAM is trained on video frames along with its saliency ground truth in the form of both saliency density map and eye-fixation points. Cornia et al.[3] propose a loss function that is a combination of different scoring metrics for saliency maps, given by:

$$L(\hat{y}, y, y^{fix}) = \alpha NSS(\hat{y}, y^{fix}) + \beta CC(\hat{y}, y) + \gamma KL(\hat{y}, y) \quad (1)$$

where $\hat{y}$, y and y^{fix} are the predicted saliency maps, ground truth saliency maps and eye-fixation points respectively and α, β and γ are three scalars which balance the three loss functions. NSS, CC and KL are saliency evaluation metrics (NSS is the Normalized Scanpath Saliency, CC is the Linear Correlation Coefficient and the KL, Kullback-Leibler Divergence). These metrics help learn saliency maps for video frames as it is helpful in estimating similarity and dissimilarity between two saliency maps [7].

2.2. Overlay/Label Placement

We consider each detected object sequentially, in the decreasing order of saliency occlusion, for placing labels on the frame. Once we place an overlay we mark the region occupied by the overlay as highly salient. We see that this region is unsuitable for placing other overlays as it will increase occlusion with salient region. Thus, we rank the object label pairs as per the label occlusion over saliency (LOS) score of bounding box of object of interest which is a metric to decide the saliency occlusion by label.

$$LOS(N, G) = \frac{\sum_{(x,y)\in N} G(x, y)}{|N|} \quad (2)$$

where N is the set of pixels (x, y) that is occluded by overlay and G is the ground truth saliency map generated from the SAM.

Apart from minimizing the occlusion with highly salient region, there are three additional objectives that we consider while placing an overlay - (a) *The overlay must be as close to the corresponding object as possible*, (b) *Connector or leader lines, connecting objects and overlays, should not intersect with each other*, (c) *The overlay must satisfy diagonal heuristic and central bias*.

2.2.1 Proximity to Objects of Interest

We propose an approach based on a strict Voronoi partioning [15] (boundary excluded) of the image space. We choose the seed points of the Voronoi partitions as the centroids of the bounding boxes that are generated from the YOLOv2 detector. The Voronoi formulation ensures that

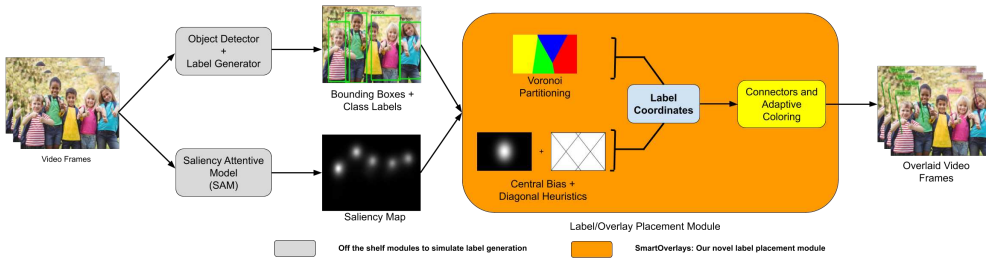

Figure 2: Overview of *SmartOverlays* label placement framework: We take video frames as input and pass it to object detector (that also generates labels) and Saliency Attention Module for saliency estimation. Thus, the object detector and label generator module also creates object-label correspondences. SAM [3] computes the saliency maps for each of the video frames. In the final module, we compute the overlay position for each label in a frame based on the object-label correspondences, saliency maps and overlay placement objectives.

the label's top left corner, P, is placed close to the corresponding objects due to the defining property of the Voronoi partitions which states that each of such partitions ensure proximity to its corresponding seed point.

2.2.2 Avoiding Intersection of Connectors

For clarity of object-label pair correspondences, we use *connectors* or *leader lines* - a path of line segments that connects the center of the object bounding box to one of the corners of the corresponding label. We choose the label corner that (i) has the least Euclidean distance to the object bounding box centroid and, (ii) is strictly within the same Voronoi partition as that of the object bounding box centroid. In our work, we use only a single line segment, $C(s_1, s_2)$ where s_1 and s_2 are the endpoints of the line segment, as connectors. As per the widely accepted aesthetic rule, a connector should not intersect with any other connector [11]. The proposed connector placement method prevents such intersections because Voronoi partitions are convex polygons [15]. Also, since the endpoints of the connectors lie in the Voronoi partition, all the points on the connector lie in the same region. Due to mutual exclusivity of the partitions, the different connectors do not intersect.

2.2.3 Diagonal Heuristics and Central Bias

Malu and Indurkhya [9] show that placing labels on the diagonal angle bisectors tends to increase the user experience in viewing. Furthermore, studies have shown that eye-fixation points tend to cluster towards the centre of the scene [13, 14]. Thus, to improve user experience, we add central bias and diagonal heuristics to the saliency map. For improving the legibility of text in labels, we use an adaptive color scheme where the text color adapts to the texture present in the label's background using Maximum HSV Complement[4].

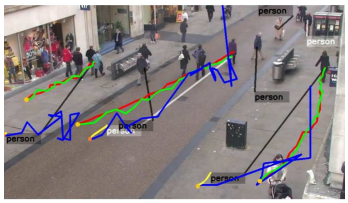

Figure 3: Visualization of robust tracking of detected objects with optical flow. **Legend**: **Green**-Detected centroid and **Red**-Corrected centroid, **Blue**-Jittery label location, **Yellow**-Corrected Label Location from method proposed in Section 2.3.

2.3. Temporal Coherence in Label Placement

Real-time label placements can be jittery due to dynamic scene changes and drastic object movements in the input videos. To address this, we employ two schemes for maintaining temporal coherence in the label locations - *Fixed Label* and *Tracking by Optical Flow* methods (refer to Figure 3).

In *Fixed Label* method, we assume the motion of moving objects to be small. Therefore, for a small time interval within Δk frames, we update the anchor points of the connectors without changing the label location. We track the locations of the corresponding objects by taking bounding box locations with the maximum IoU among all the bounding box pairs (with IoU greater than 0.75).

For *Tracking by Optical Flow* method, we extend the *Fixed Label* method. Along with centroid, here, we also update the location of centroid of object bounding box, C_i^n, along with label location, L_i^n, in frame f_i and $n \in \{1, .., N_O\}$ (N_O being the number of tracked objects). We update C_i^n to UC_i^n using exponential weighted average as follows

$$UC_i^n = (1 - \beta)C_i^n + \beta T_{(i-1)}^n, \tag{3}$$

with β being the weight. Exponential weighted average ensures a smooth flow of label resulting in minimum jitter.

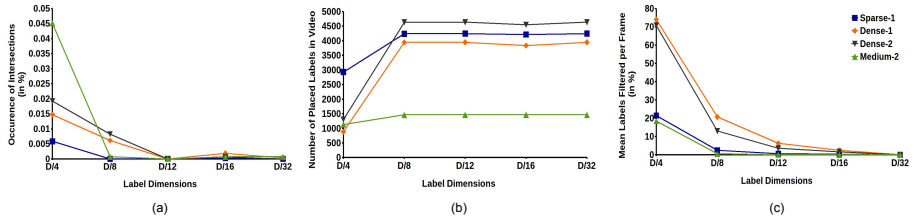

Figure 4: Label leader lines intersections. (a) shows the percentage of objects intersecting leader lines in the entire video. Label dimensions are directly proportional to label-leader line intersections owing to the increase in inter label distances. (b) shows the total number of labels placed in the entire video sequence. In (c) we see the effect of label filtering for larger label dimensions due to unavailability of space for placement (D is the image dimension in all the analysis).

We choose centroid and 8 points around centroid for robust object centroid tracking. We select multiple points as there might be an error in tracking only a few points. More than 9 points could be used but empirically we found that using 9 was enough to maintain symmetry and reduce computation. We track these 9 points in subsequent frames, using optical flow tracking [2]. In the corresponding object voronoi partition a location, L_i, is computed with minimum saliency. Now the updated label location, UL_i^n, is calculated by taking exponential weighted average again, as

$$UL_i^n = (1-\gamma)L_i^n + \gamma UL_{(i-1)}^n, \tag{4}$$

with γ being the weight. We give more weightage to the information obtained from previous frames and, thus, set β and γ to 0.9.

3. Experiments and Results

We performed all the experiments on an Linux platform system with an Intel Xeon CPU E5-2697 v3 2.60GHz with an Nvidia Tesla K40c GPU with 12GB RAM. For label generation and object detection, we use an YOLOv2 pretrained on COCO [8] dataset that has 80 classes. We resize input video frames to 608×608 before passing it to YOLOv2. For computing saliency maps, we use a SAM that is pretrained on SALICON dataset [6] containing eye fixation ground truths for images. For evaluating our placement We use the values of the hyperparameters $\alpha = -1$ = , $\beta = -2$ and $\gamma = 10$, in loss function as specified in [3].

We also investigated a few user defined specifications and their effects in the framework. The user can define the shape and size of the labels that we overlay. In cases where the size of a label is larger than that of the corresponding Voronoi partition the search space will be empty. As a result the placement module misses out on such labels and this helps in filtering labels in a cluttered environment (see Figure 4).

For evaluating the jitters from the methods, to ensure temporal coherence in labels, we define a metric, *temporal jitter* metric, M_j, as $M_j = \frac{d_l}{d_o+\epsilon}$ where d_o and d_l are the distance traveled by an object and its corresponding label throughout the duration of the video and ϵ is a small

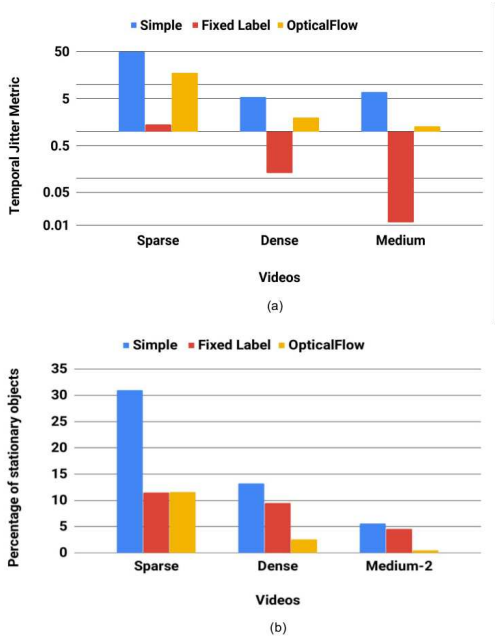

Figure 5: Comparison of temporal coherence in label placement by *SmartOverlays* without temporal coherence, fixed label method and optical flow tracking method. (a) Comparison of changes to temporal jitter metric on different types of videos (y-axis in log-scale), and (b) the number of stationary object vary for different techniques throughout the video sequence.

constant added to avoid division by zero. M_j captures the relative motion of label with respect to the its corresponding object due to the d_o in the denominator. Figure 5 shows the effectiveness of the proposed algorithms in terms of low M_j with varying number of stationary and moving objects. Figure 5(a) depicts an important observation that jitter in *Fixed Label* method is lesser than *Optical Flow Tracking* method. This could be because the latter method takes into account the motion of the object which could be jittery due to the inaccuracies in object detection step. Figure 5(b) shows that lower number of stationary objects can also result in jittery label placement highlighting the effectiveness of tracking based methods for label placement in videos.

References

[1] Ronald Azuma and Chris Furmanski. Evaluating label placement for augmented reality view management. In *Proceedings of the 2nd IEEE/ACM international Symposium on Mixed and Augmented Reality*, page 66. IEEE Computer Society, 2003.

[2] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[3] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, Oct 2018.

[4] Joseph L Gabbard, J Edward Swan, Deborah Hix, Robert S Schulman, John Lucas, and Divya Gupta. An empirical user-based study of text drawing styles and outdoor background textures for augmented reality. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 11–18. IEEE, 2005.

[5] Raphael Grasset, Tobias Langlotz, Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. Image-driven view management for augmented reality browsers. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 177–186. IEEE, 2012.

[6] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1072–1080. IEEE, 2015.

[7] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[9] Gautam Malu and Bipin Indurkhya. An approach to optimal text placement on images. In Don Harris, editor, *Engineering Psychology and Cognitive Ergonomics. Understanding Human Cognition*, pages 68–74, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[10] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. IEEE, 2017.

[11] Dieter Schmalstieg and Tobias Hollerer. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016.

[12] Thierry Stein and Xavier Décoret. Dynamic label placement for improved interactive exploration. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 15–21. ACM, 2008.

[13] Benjamin W Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of vision*, 7(14):4–4, 2007.

[14] Po-He Tseng, Ran Carmi, Ian GM Cameron, Douglas P Munoz, and Laurent Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of vision*, 9(7):4–4, 2009.

[15] Georges Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.