Computational Modeling of Top-down Visual Attention in Interactive Environments

Ali Borji
borji@usc.edu
Dicky N. Sihite
sihite@usc.edu
Laurent Itti
itti@usc.edu

Dapartment of Computer Science
University of Southern California
Los Angeles, CA, USA

Modeling how visual saliency guides the deployment of attention over visual scenes has attracted much interest recently — among both computer vision and experimental/computational researchers — since visual attention is a key function of both machine and biological vision systems. Research efforts in computer vision have mostly been focused on modeling bottom-up saliency. Strong influences on attention and eye movements, however, come from instantaneous task demands. Here, we propose models of top-down visual guidance considering task influences. The new models estimate the state of a human subject performing a task (here, playing video games), and map that state to an eye position. Factors influencing state come from scene gist, physical actions, events, and bottom-up saliency. Proposed models fall into two categories. In the first category, we use classical discriminative classifiers, including Regression, kNN and SVM. In the second category, we use Bayesian Networks to combine all the multi-modal factors in a unified framework. Our approaches significantly outperform 15 competing bottom-up and top-down attention models in predicting future eye fixations.

0.1 Features

Employed features are from vision and action modalities. For description of the scene we use light-weight yet highly discriminant features. For driving games, we have collected action data which we combine with annotated scene events (*e.g.*, stop sign) for state determination.

Mean eye position (*MEP*). MEP (mean of the distribution of all human fixated locations) is an oracle prediction derived from the human data itself (as opposed to computed by an algorithm).

Gist (*G*). Gist (scene context) is a very rough representation of a scene and does not contain much details about individual objects or semantics but can provide sufficient information for coarse scene discrimination (*e.g.*, indoor vs. outdoor or category of the scene). The pyramid-based feature vector (pfx) [3], relies on 34 feature pyramids from the bottom-up saliency model: 6 intensity channels, 12 color channels (first 6 red/green and next 6 blue/yellow color opponency), and 16 orientations. For each feature map, there are 21 values that encompass average values of various spatial pyramids: value 0 is the average value of the entire feature map, values 1 to 4 are the average values of each 2×2 quadrant of the feature map and values 5 to 20 are the average value for each of the 4×4 grids of the feature map leading to overall of $34 \times 21 = 714$ elements. It is possible to reduce dimensionality of this vector while maintaining discriminability.

Bottom-up saliency map (*BU*). This model includes 12 feature channels sensitive to color contrast (red/green and blue/yellow), temporal luminance flicker, luminance contrast, four orientations $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$, and four oriented motion energies (up, down, left, right). After center-surround difference operations and across scale competitions, a unique saliency map is created and subsampled to a 20×15 feature map which is linearized to a vector of 1×300 [1]. We used the original bottom-up saliency map both as a signature of the scene and a saliency predictor.

Physical actions (*A*). In the driving experiment, action is a 22D feature vector containing wheel positions, pedals (brake and gas), left and right signals, mirrors and left and right views, gear change, etc which are wheel buttons that subjects used for driving. Note that in general, physical actions recorded in this way are different than actions that happen in the game but they convey some knowledge about them.

Labeled events (*E*). Each frame of games was manually labeled as belonging to one of different events such as {left turn, right turn, going straight, red light, adjusting left, adjusting right, stop sign, traffic check and error frames due to unexpected events that terminate the games like hitting other cars}. Hence this is only a scalar feature.

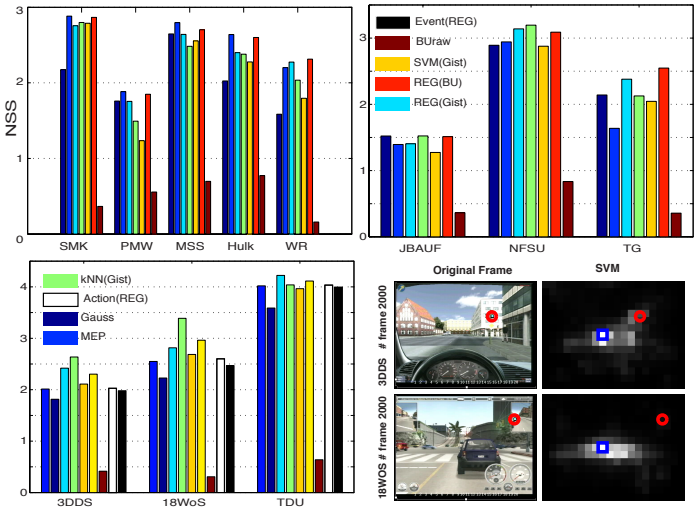

Figure 1: Fixation prediction results over 11 video games. Bottom-right panel shows two sample frames and corresspondings attention maps learned by SVM classifier. Red circle is the fixation

0.2 Classifiers

The classifiers estimate $p(X|S_t) = \frac{p(S_t|X)p(X)}{p(S_t)}$ where S_t is a feature vector (or combination of them) estimating subject state. $P(X)$ is the prior over eye positions (the MEP model computed over other subjects than the one under test) and is biased by likelihood $p(S_t|X)$ (probability of state given eye position). In the case where S_t is only the Gist, our method reduces to the approach in [2].

Regression(REG): Assuming a linear relationship between feature vectors M and eye fixations N, we solve the equation $M \times W = N$. The solution is: $W = M^+ \times N$, where M^+ is the (least-squares) pseudo-inverse of matrix M. When the feature vector is b (a constant scalar), the solution (predicted map) is simply the average of all eye position vectors in N. This classifier is equivalent to the MEP model. We used SVD to find the pseudo inverse of matrix M.

kNN: The idea here is to look into training data and find similar neighborhoods to the current test frame and then make attention maps from the associated eye fixations. This resembles a local MEP model, where we make a map with 1's at fixated locations and zeros elsewhere. Then to generate an attention map, we convolve this map it with a Gaussian filter.

SVM: To use SVM, we first reduced the high-dimensional feature vector using PCA to preserve 95% of variance. Then a linear multi-class SVM was trained from other subjects with 300 output classes. Due to the high number of classes and huge amount of data using SVM is slow. Experimenting over a subset of the data with low-resolution eye fixation maps (4×3 and 8×6 hence number of classes 12 and 48) and with polynomial and RBF kernels did not improve the results.

Fig. 1 shows classification results over 11 video games (1.4M video frames and 11M fixations) using Normalized Scan-path Saliency (NSS score). MEP model is simple the average fixation position calculated from our training data (leave-1-out paradaigm) and BU is the bottom-up

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

[2] Peters R. J. and Itti L. Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *Proc. CVPR*, 2007.

[3] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions PAMI*, 29(2):300–312, 2007.