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

Abstract

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 on 18,000 and 75,00 video frames and eye movement samples from a driving and a flight combat video game, respectively. We further test and validate our approaches on 1.4M video frames and 11M fixations samples and in all cases obtain higher prediction scores that reference models.

1 Introduction

The human visual system is highly efficient in dealing with huge amounts of visual information. This is due to a mechanism called visual attention that guides eye gaze toward objects/locations of interest in the scene. Two different types of attention processing are: bottom-up mechanisms (involuntary and very sensitive to salient stimuli) and top-down mechanisms (voluntary, knowledge- and goal-oriented) [10] [20].

Bottom-up saliency mechanisms are based on within-image competitions in which some items stand out from their surrounding regions. They correlate best with fixations during free viewing [11] [34]. Example applications of bottom-up saliency modeling are: object/person detection, segmentation and recognition [28], robotics localization [37], image re-targeting [33], thumbnailing [22], image and video compression [15], non-photo-realistic rendering [5] and seam carving [32].

BMVC 2011 http://dx.doi.org/10.5244/C.25.85

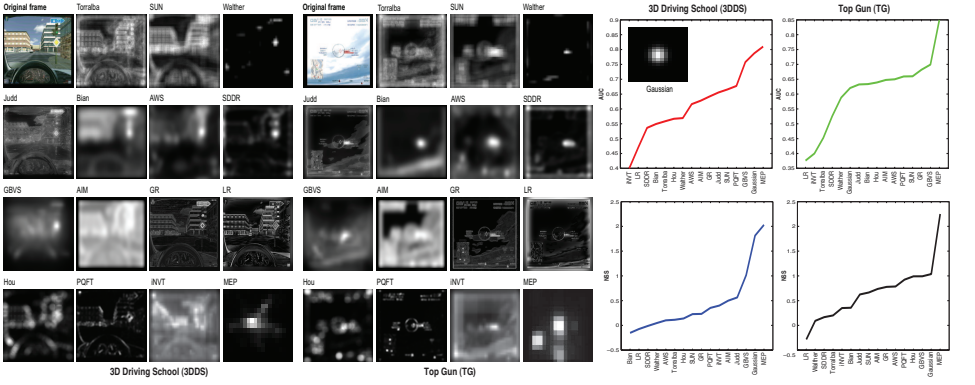


Figure 1: Left: Sample frames along with corresponding saliency maps of models. Right: AUC scores (chance level is 0.5, higher scores indicate better models) and NSS scores (chance level is 0.0, higher is better; see Sec. 3.2) of 14 saliency models over 3D Driving School and Top Gun games. Some models are able to detect the traffic light sign as salient, which happens to be task-related in the sample shown image. Overall performance of models is very poor compared to the inter-observer (MEP) model.

In complex real world tasks, top-down factors often predominate bottom-up factors. In Fig. 1, some major bottom-up saliency models were applied for saliency prediction in two tasks: urban driving and a target shooting game. As results show, performance of these bottom-up models was poor compared to simple predictors, which are the mean eye position map of other subjects (called MEP model, cf. Sec. 2.1) and a Gaussian blob at the center of the image [39]. The best bottom-up model over these data (GBVS) achieved 1.01 NSS score (i.e., saliency at human fixated locations was 1.1 standard deviations above the mean at all image locations) for the driving game (3DDS), and 0.99 over the flight combat game (TG). In contrast, the simple MEP model scored NSS of 2.03 and 2.3 over 3DDS and TG, respectively. These results highlight the poor prediction power of saliency models, when humans are actively engaged in a task and thus strongly top-down driven [1].

How do humans decide where to look or what to attend to in different situations when performing a complex task? This is a hard question since top-down attention engages many different high level brain and body structures and functions, which have been long studied but not yet fully elucidated by cognitive science and AI researchers. In the lack of a general answer, for some tasks, however, mechanisms have been discovered in controlled laboratory setups (*e.g.*, 'block copying' [4], 'making tea' [17], 'driving' [18], and 'reading' [29]). Despite task-based differences, some task-independent top-down mechanisms have been enumerated. For instance, Land and Hayhoe [17], classified eye fixations into four categories: Locating (searching for) a needed object (*e.g.*, milk in the fridge), Directing the hand (grabbing something from shelf), Guiding (lid onto kettle), and Checking (water depth, spout). Then, they proposed a schema for how to compose these so-called *object-related actions* (ORA) to perform a task. In a behavior-based realm, this corresponds to breaking down a complex task into a series of basis functions (micro behaviors, *e.g.*, grasping), and using arbitration on top to choose one of these behaviors at a time and reach a macro behavior (see

[1] To compare bottom-up saliency models over our data, we asked their authors for the implementation code, including: Torralba et. al [40], SUN [44], Walther [42], Judd *et al.* [38], Bian et al, [27], AWS [3], SDSR [35], GBVS [12], AIM [25], Global Rarity (GR) [21], Local Rarity (LR) [21], Hou [43], PQFT [9], and iNVT [11].

[23] for an application of this approach in attention modeling).

While most modeling studies have had limited scope and been focused on a specific task, in this paper, we elaborate on general influences of multi-modal information onto top-down spatial attention. We learn models that generate a likelihood over locations to be fixated in each situation. Eye movements of human subjects were gathered while they played different types of video games. Our models output an attention guidance map, similar to bottom-up saliency maps, but with the difference that top-down influences determine interesting hotspots in our maps (regions of predicted high probability of being fixated) as opposed to bottom-up saliency cues. Modeling top-down attention, besides helping interpret experimental studies, has applications including interactive computer graphics environments (video game playing and virtual reality), flight and driving simulators, and visual prosthetic devices.

Related Work: The conventional features used to extract bottom-up saliency include intensity, orientation, color and motion information [11] [10]. In addition, saliency models have been proposed based on following concepts: Self-similarity in visual information [35], Rarity [21], Surprise [16], Information maximization (AIM) [25], Symmetry [7], Bayesian [44], Spectral residual saliency (Fourier) [43], and many others. Some models train a classifier to distinguish fixated patches from random patches. When facing a scene, they assign to each patch the probability of that patch to be fixated [41] [38] [6]. The concept of saliency detectors operating in spatiotemporal neighborhoods has recently begun to be used for spatiotemporal analysis with emerging applications to video classification, event detection and activity recognition [14]. Examples are the extension of the Harris corner detector to 3D by Laptev [19], spatiotemporal extension of the salient point detector of Kadir and Brady by Oikonomopoulos *et al.* [26]. Willems *et al*. proposed a computationally efficient space-time detector based on the determinant of the 3D Hessian matrix [8]. Some saliency models have incorporated these ideas (*e.g*., [14]).

Some architectures for modeling top-down attention have been introduced. Peters and Itti [13] introduced a model that maps a signature of a scene ("Gist" using pyramid features of basic saliency model [11] or Fourier features) to the eye position using a regression classifier. A combined map of the pointwise product of the learned top-down map and bottom-up saliency map scored higher prediction accuracy. Proposed models here are in-line with this study, with the contributions that we use stronger classifiers and richer information indicative of state at each time. Navalpakkam and Itti [24] proposed a cognitive model of task-driven attention but it has not been fully implemented to generate top-down maps. Sprague and Ballard [23] defined some basic visual behaviors (routines) such as litter collection, obstacle avoidance, sideway walking, for an avatar and proposed a reinforcement learning approach for how to coordinate these behaviors to perform a simple task in a virtual environment.

2 Top-down Attention Modeling

To fulfill task demands, humans have to perform actions while attending to different items based on an internal model that changes state over time. This state transition is influenced by environmental variables and subjective factors. Since there is a high correlation among subjects in performing the same task, we estimate the state from data of other subjects in a similar situation. Formally, we calculate the probability of image location X to be attended in state S_t ($p(X|S_t)$). Since we don't have direct access to S_t, we estimate it from observable variables. In the first class of proposed models, we follow a discriminative approach, where we directly calculate the above probability from data. In section 3.3, we propose a generative

model using Bayesian Networks to model interaction of important variables in a task.

2.1 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). One difference between MEP map in dynamic environments and static images (also called inter-observer model) is that MEP in static images outperforms all other models. The same statement applies over movies when fixations on a frame could be used to build an inter-observer map. However, in dynamic environments used in this paper, since frames are generated dynamically under each player's control, aligning frames across subjects is not possible. Therefore if a method could dynamically predict eye movements on a frame by frame basis then achieving a higher accuracy than MEP is possible.

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) [36], 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 [11]. 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.

2.2 Classifiers

The protocol for making classifiers is as follows. Over n subjects $H_i, i = 1 \cdots n$, in a leave-one-out approach, a model is learned from the data of other subjects $H_i, i = 1 \cdots n, i \neq j$ and

tested over the remaining j-th subject. The final result is the average over all these j-th subjects. To learn a model, features are mapped to 2D eye positions. 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 [13].

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. An important point here is that we set eigenvalues smaller than half of the biggest eigenvalue to zero to avoid numerical instability. Vector P which is the eye position over the 640×480 image is downsampled to 20×15 and transformed into a 1×300 vector with a 1 at the actual eye position and zeros elsewhere. In testing, to predict eye positions for new test frames, feature vectors (as above) are first extracted, and attention maps are generated by applying the learned mapping: Let U be the feature vector matrix for a test subject and V its corresponding predicted eye fixations; then $V = U \times W$. The maximum of the attention map indicates the model's predicted most likely fixated location.

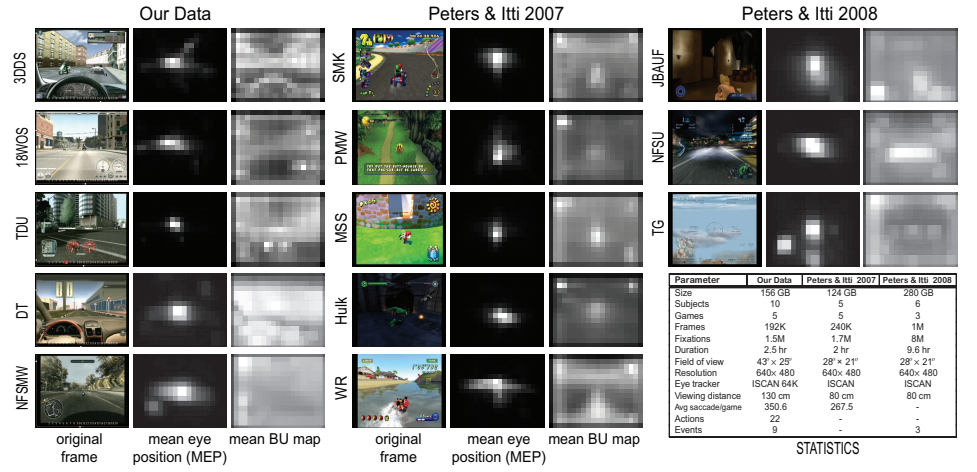
kNN: The idea here is to look into training data and find similar neighborhoods to the current test frame and then make attention maps from the associated eye fixations. This resembles a local MEP model, where we make a map with 1's at fixated locations and zeros elsewhere. Then to generate an attention map, we convolve this map it with a Gaussian filter. For fast testing, we did as follows. Let matrix Q denote similarities (dot product) of all test frames of one subject to all training frames. Then $Q = U \times M^{'}$ where matrix U is of size $|U| \times |M^{'}|$ with $|U|$ as the number of frames for a subject. Let Z be a matrix of size $|Q| \times 300$ of zeros. For $j = 1 \cdots k$, (k number of neighbors in kNN, here 10) maxima of all rows in Q are calculated, which indicate the $j-th$ most similar training frame to each test frame. Then, Z is convolved with a linearized Gauss kernel (1×300) and updated over j. Each time after updating, the value at the $j-1$ th location is set to a large negative value to not be chosen in the next round (next j). Note that with performing operations on matrices in this fashion, there is no need to loop through test frames.

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.

3 Experiments

3.1 Data Gathering

Video games are suitable stimuli for studying task-driven attention because they are interactive, have near-natural renderings and statistics and are easy to control and work with in the lab compared with real-world setups. We chose driving, since it is a demanding task requiring coordinated action and active attention for an experienced driver. We also evaluated our



Parameter	Our Data	Peters & Itti 2007	Peters & Itti 2008
Size	156 GB	124 GB	280 GB
Subjects	10	5	6
Games	5	5	3
Frames	192K	240K	1M
Fixations	1.5M	1.7M	8M
Duration	2.5 hr	2 hr	9.6 hr
Field of view	43°× 25°	28°× 21°	28°× 21°
Resolution	640× 480	640× 480	640× 480
Eye tracker	ISCAN 64K	ISCAN	ISCAN
Viewing distance	130 cm	80 cm	80 cm
Avg saccade/game	350.6	267.5	-
Actions	22	-	-
Events	9	-	3

STATISTICS

Figure 2: Sample frames, mean eye positions and mean bottom-up saliency maps of used datasets with their statistics summarized in the bottom-right table.

approaches over some other already available datasets.

Participants were 10 subjects 18-25 years old with valid driving license and at least 2 years of driving experience. Experimental protocol was approved by our University's Institutional Review Board. Subjects were compensated for their participation. Each subject played each of the 3 games: 3D Driving School (3DDS), 18 Wheels of Steel (18 WOS), and Test Drive Unlimited (TDU). We have also recorded data over two other games: Driver Test (DT) and Need for Speed Most Wanted (NFSMW) (Fig. 2). Due to huge amount of data, we limit our analysis to the first three games. There was a 5-min training session for each game in which subjects were introduced to the goal of the game, rules, buttons, etc. After training, subjects played the game for another 5 minutes. At the beginning of the test session, the eye tracker was calibrated using 9-point calibration. Training and testing phases were from the same game but different situations. Subject's distance from screen was 130 cm yielding field of view of $43^\circ \times 25^\circ$. The overall recording (over 3 games) resulted in 2.5 hours or 156 GB video, 192,000 frames, 1,536,000 fixations, and 10,518 saccades.

Subjects played driving games on PC1 with Windows XP running the games. An array of wheel, pedal and other actions (signal, mirror, etc) was logged with frequency of 62Hz. Frames were recorded on PC2 running Linux Mandriva OS. Game stimuli were shown to the subject at 30Hz. This machine sent a copy of each frame to the LCD monitor and saved one copy to the hard disk. PC2 also instructed the eye tracker (PC3) to record eye positions. PC2 had a dual-CPU processor and used `SCHED-FIFO` scheduling to ensure microsecond-accurate timing. Each subject's right eye position was recorded at 240 Hz with a hardware-based eye-tracking system (ISCAN Inc. RK-464). Subjects drove using the Logitech Driving Force GT steering wheel, automatic transmission, brake and gas pedals, 11-inch rubber-overmold rim, 900 degrees rotation (only 360 degrees; 180 left, 180 right; were used in experiments), Force Feedback, connected via USB to the PC1.

Peters and Itti 2007: Contains 5-minute segments of game playing of Nintendo games (Super Mario Kart (SMK), Pac Man World (PMW), Mario Sunshine (MSS), Hulk, and Wave Race (WR). Subjects played overall 24 sessions (unequal number of sessions) [13].

Peters and Itti 2008: Six subjects played 3 GameCube games: A first person shooting

game (fps) called James Bond Agent Under Fire (JBAUF), a racing game called Need For Speed Underground (NFSU) and a flight combat game called Top Gun (TG). None of the subjects had prior experience with these games. For each game, subjects first practiced the game for several one-hour sessions on different days until reaching a success criterion, and then returned for a one-hour eye tracking session with that game. Within each game, subjects played 3 game levels, and during eye tracking, each subject played each game level twice. Thus, in total, recorded data set consists of video frames and eye tracking data from 108 clips (6 subjects $\times$ 3 games per subject $\times$ 3 levels per game $\times$ 2 clips per level) [31].

Sample frames with the mean eye position (MEP), average bottom-up maps as well as statistics of all are shown in Fig. 2. There are also some other eye movements datasets that have mainly been collected for studying top-down attention. Some could be found here [2].

3.2 Evaluation Metrics

To quantify how well a model can predict the actual human eye focusing positions, we used two metrics: 1) Normalized scan-path saliency (NSS) and 2) AUC score.

NSS: NSS is the response value at the human eye position, (x_h, y_h), in a model's predicted gaze density map that has been normalized to have zero mean and unit standard deviation $NSS = \frac{1}{\sigma_s}(S(x_h, y_h) - \mu_S)$, $NSS = 1$ indicates that the subject's eye positions fall in a region whose predicted density is one standard deviation above average while $NSS = 0$ indicates that the model performs no better than picking a random position on the map.

One issue when evaluating saliency models is center-bias which means a majority of eye data happens to be in the center [39]. Over video games, game designers often put the interesting and task-relevant items at the center (*e.g.*, main actor, road, commands). Therefore, a trivial model like MEP or Gaussian Blob usually scores high. Center-bias is tightly related to another problem which is observer agreement that shows a strong peak in the eye data. This peak generates many true positives for the MEP model, and hence high scores (any type of score) over many frames. Since the chance of making false positives is thus small for MEP (because of less data at the tails of distribution), there is less opportunity for models to show their superiority over MEP or Gaussian. One remedy is to compare models over data with uniform overall distributions, which is hard to control. The other possibility is to design new scores or evaluation approaches. To stretch the differences between sophisticated computational models and brute-force models, each time we discarded those fixations that were in top $\alpha\%, \alpha \in \{0, 10, 20, ...90\}$ of the MEP map (note, this is different than percentile). This gives an idea of how well models predicted "non-trivial" fixations, i.e., away from the central peak of MEP data. Then to summarize these values we chose the mean statistic.

AUC: Using the AUC metric, a model's saliency map is treated as a binary classifier on every pixel in the image; pixels with larger saliency values than a threshold are classified as fixated while the rest of the pixels are classified as non-fixated. Human fixations are used as ground truth. By varying the threshold, the ROC curve is drawn as the false positive rate vs. true positive rate, and the area under this curve (AUC) indicates how well the saliency map predicts actual human eye fixations. Perfect prediction corresponds to a score of 1.

3.3 Results

Results of task-based saliency detection are summarized in Fig. 3. All models performed higher than chance. Over our data (all 3 games, 3rd row in **a** and **c**), kNN classifier achieved the best score followed by Regression and SVM classifiers, all with Gist features, for both

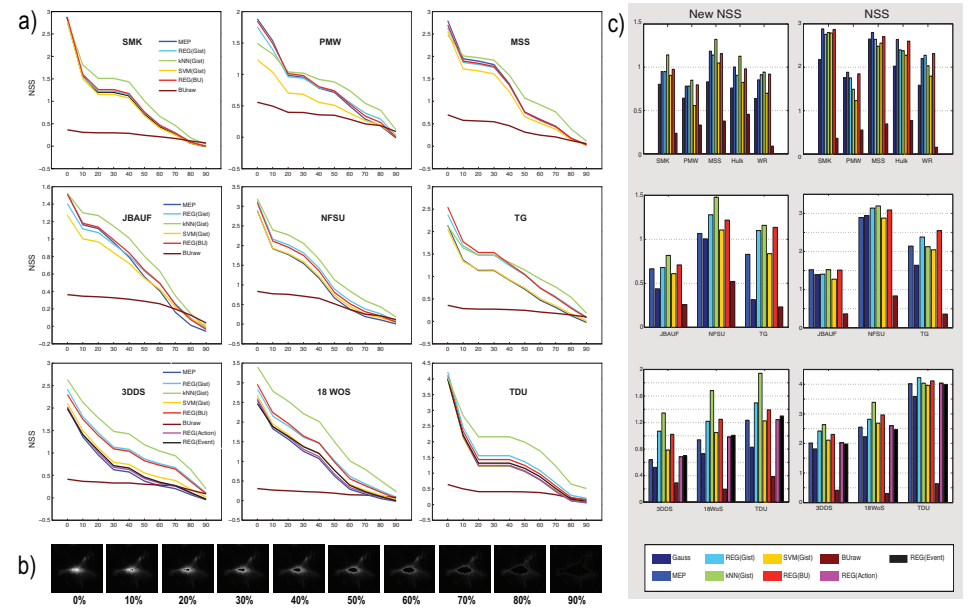


Figure 3: a) NSS scores over three video games for different amounts of data, b) Fixation maps with α% of data discarded and c) Average NSS over saliency levels (left) and NSS score over all fixations (i.e 0% case) for classifiers.

new NSS and traditional NSS scores. Regression classifier with Event and Action features performed higher than MEP and Gaussian models. The pure bottom-up saliency model performed the worst again, highlighting that BU saliency does not account for top-down attention (This is the case across all games). Over other two datasets, due to higher center-bias (verify from Fig. 2), models achieved almost similar scores at the 0% level of MEP. Over the Peters & Itti 2007 data (1st row), the MEP model achieved higher score over games (except WR when Regression classifier outperformed). However with new NSS score, kNN classifier with Gist features showed a big improvement. Over the Peters & Itti 2008 data (2nd row), results are consistent with results over our data when kNN classifier showed the best performance over both scores (except TG when Regression with bottom-up features won). Overall, kNN classifier seems to be a better for eye fixation prediction over these data.

In another experiment, we used the proposed models for prediction of the next action. As shown in Fig. 4.a, using employed features (here we also used 2D eye position as a feature), a Regression classifier was able to predict actions (22D vector of action) better than a a model that is the average of actions (similar to MEP for eye positions) in terms of NSS score. BU map and Gist scene descriptors performed better than other features. Fig. 4.b shows an upper bound on NSS score when fixations of previous frames were considered as predictors for the current frame (averaged across subjects for each game). This is the score of an optimal model that could consider subjectivity, noise and task demands, and it provides an interesting comparison point for our computational models.

Bayesian Networks: In this part, we propose a generative model based on Bayesian Networks to systematically learn relationships between variables and eye position. To accommodate features for use of Bayes Net, we clustered the high-dimensional Gist vector using k-means to r clusters (here $r = 20$). Continuous wheel and pedal positions were dis-

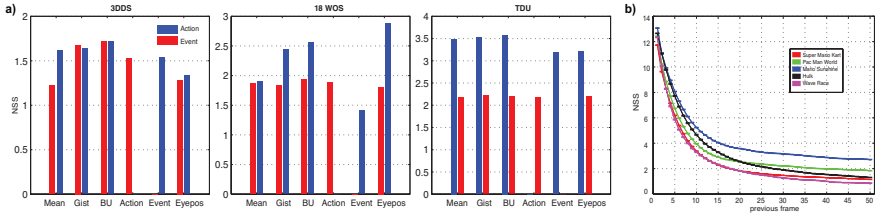


Figure 4: a) Action and Event prediction over driving games b) Upper-bound in NSS score.

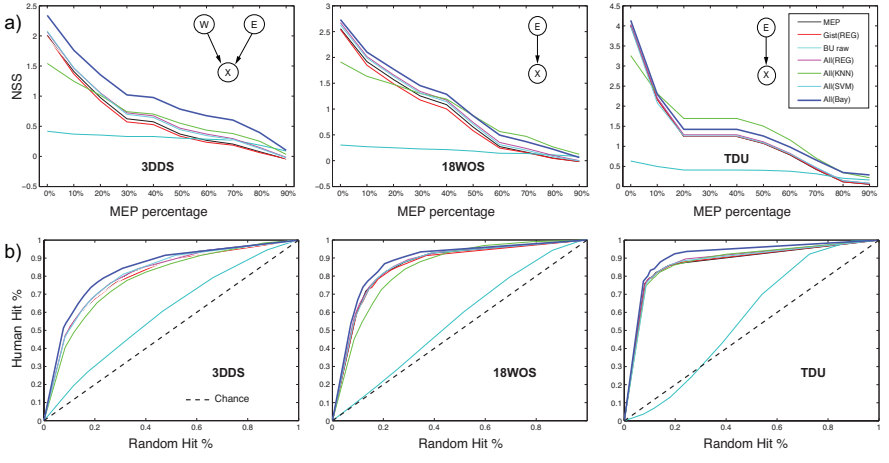


Figure 5: a) NSS and b) ROC curves over driving games with the best learned Bayes Net.

cretized to 8 values. Number of events were 9. Due to high complexity of these games a manually-designed Bayes Net is less likely to produce good results (We systematically experimented with several network topologies). Thus, we used a variant of Markov Chain Monte Carlo (MCMC) algorithm called Metropolis-Hastings (MH) to search the space of all DAGs in a network that has all variables (Gist (G), BU map (B), Wheel(W), Pedal(P) and Event (E)) connected to eye position (X). Learned network structures are shown in Fig. 5. The Bayesian Network approach resulted in higher NSS and ROC scores over all three driving games, compared to the other approaches, when using the same features. In the Bayesian Network model, MEP it is a prior distribution of data over eye position variable so by default such a network is going to perform better than MEP. For implementation of Bayesian Networks we used a Bayes Net toolbox freely available [1]. Sample frames of driving games and their corresponding top-down attention maps generated by models are shown in Fig. 6.

4 Discussion and Conclusion

In this paper, we proposed frameworks for learning task-based top-down spatial attention. Our models outperform previous approaches and simple heuristic models. The slightly higher performance of classic classifiers over the Bayes Net model is because of the lower-dimensional features used in the Bayes Net; yet, when compared using the same features, the Bayes Net outperformed all other approaches. Despite their higher computational cost which may restrict the dimensionality of features that can be used, Bayesian Networks and their variants (Dynamic Bayesian Networks) give us the capability to reason over scene content at

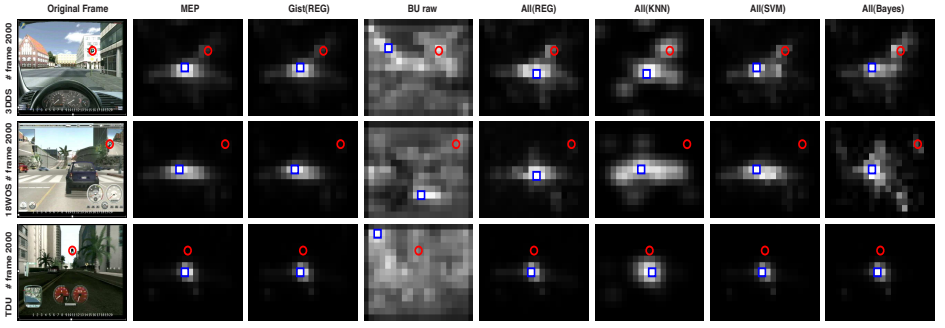


Figure 6: Model prediction maps. Each red circle indicates the observer's actual eye position superimposed with each map's peak location (blue squares).

the object level, which is subject to our future work. Similar approaches have been followed in the past for modeling reading tasks [29] and other cognitive tasks (*e.g.*, arranging items on a table [30]). This study demonstrates that it is possible to develop computational models which are capable of estimating state and predicting task-dependent future eye movements and actions of humans engaged in complex interactive tasks.

References

[1] http://code.google.com/p/bnt/.

[2] http://www.cis.rit.edu/pelz/scanpaths/scanpaths.htm.

[3] Garcia-Diaz A., Fdez-Vidal X. R., Pardo X. M., and Dosil R. Decorrelation and distinctiveness providewith human-like saliency. In *Proc. ACIVS (LNCS)*, 2009.

[4] D. Ballard, M. Hayhoe, and J. Pelz. Memory representations in natural tasks. *Journal of Cog. Neurosci.*, 7(1):66–80, 1995.

[5] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics*, 21(3):769–776, 2002.

[6] Vig E., Dorr M., Martinetz T., and Barth E. A learned saliency predictor for dynamic natural scenes. In *Proc. ICANN*, 2010.

[7] Kootstra G., Nederveen A., and de Boer B. Paying attention to symmetry. In *Proc. BMVC*, 2008.

[8] Willems G., Tuytelaars T., and Gool. V. G. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. LNCS*, 2008.

[9] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, 19(1): 185–198, 2010.

[10] L. Itti and C. Koch. Computational modeling of visual attention. *Nat. Rev. Neurosci.*, 2(3): 194–203, 2001.

[11] 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.

[12] Harel J., Koch C., and Perona P. Graph-based visual saliency. In *Proc. NIPS*, 2006.

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

[14] Rapantzikos K., Avrithis Y., and Kollias S. Dense saliency-based spatiotemporal feature points for action recognition. In *Proc. CVPR*, 2009.

[15] Itti L. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Process*, 13(10), 2004.

[16] Itti L. and Baldi P. A principled approach to detecting surprising events in video. In *Proc. CVPR*, 2005.

[17] M. Land and M. Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41(25):3559–3565, 2001.

[18] M. F. Land and D. N. Lee. Where we look when we steer. *Nature*, 369:742–744, 1994.

[19] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[20] M M. Corbetta and G. L. Shulman. Control of goalâĂŘdirected and stimulusâĂŘdriven attention in the brain. *Nature Review Neuroscience*, 3(3):201–215, 2002.

[21] M. Mancas. Computational attention: Modelisation and application to audio and image processing, 2007. PhD. thesis.

[22] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau. A coherent computational approach to model bottom-up visual attention. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(5):802–817, 2006.

[23] Sprague N. and Ballard D. H. Eye movements for reward maximization. In *Proc. NIPS*, 2003.

[24] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45(2): 205–231, 2005.

[25] Bruce N.D.B. and Tsotsos J.K. Saliency based on information maximization. In *Proc. NIPS*, 2005.

[26] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(3):710–719, 2006.

[27] Bian P. and Zhang L. Biological plausibility of spectral domain approach for spatiotemporal visual saliency. In *Proc. LNCS*, 2009.

[28] Achanta R., Hemami S., Estrada F., and Susstrunk S. Frequency-tuned salient region detection. In *Proc. CVPR*, 2009.

[29] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 85:618–660, 1998.

[30] R. D. Rimey and C. M. Brown. Controlling eye movements with hidden markov models. *International Journal of Computer Vision*, 7(1):47–65, 1991.

[31] Peters R.J. and Itti L. Congruence between model and human attention reveals unique signatures of critical visual events. In *Proc. NIPS*, 2008.

[32] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM Transactions on Graphics (SIGGRAPH)*, 2008.

[33] Goferman S., Zelnik-Manor L., and Tal A. Context-aware saliency detection. In *Proc. CVPR*, 2010.

[34] D. S. Wooding S. Mannan, K. H. Ruddock. Fixation patterns made during brief examination of 2-d images. *Perception*, 27:1059–1072, 1997.

[35] H.J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, 9(12):1–27, 2009.

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

[37] C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *IEEE Transactions on Robotics*, 25(4):861–873, 2009.

[38] Judd T., Ehinger K., Durand F., and Torralba A. Learning to predict where humans look. In *Proc. ICCV*, 2009.

[39] B. 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(4):1–17, 2007.

[40] A. Torralba. Modeling global scene factors in attention. *Journal of Optical Society of America*, 20(7):1407–1418, 2003.

[41] Kienzle W., Wichmann A. F.and Scholkopf B., and Franz M. O. A nonparametric approach to bottom-up visual saliency. In *Proc. NIPS*, 2007.

[42] D. Walther and C. Koch. Modeling attention to salient protoobjects. *Neural Networks*, 19(9): 1395–1407, 2006.

[43] Hou X. and Zhang L. Saliency detection: A spectral residual approach. In *Proc. CVPR*, 2007.

[44] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(32):1–20, 2008.