Applying computational tools to predict gaze direction in interactive visual environments

ROBERT J. PETERS
Department of Computer Science
University of Southern California
and
LAURENT ITTI
Departments of Computer Science, Neuroscience and Psychology
University of Southern California

Future interactive virtual environments will be "attention-aware," capable of predicting, reacting to, and ultimately influencing the visual attention of their human operators. Before such environments can be realized, it is necessary to operationalize our understanding of the relevant aspects of visual perception, in the form of fully-automated computational heuristics that can efficiently identify locations that would attract human gaze in complex dynamic environments. One promising approach to designing such heuristics draws on ideas from computational neuroscience. We compared several neurobiologically-inspired heuristics with eye movement recordings from five observers playing video games, and found that heuristics which detect outliers from the global distribution of visual features were better predictors of human gaze than were purely local heuristics. Heuristics sensitive to dynamic events performed best overall. Further, heuristic prediction power differed more between games than between different human observers. While other factors clearly also influence eye position, our findings suggest that simple neurally-inspired algorithmic methods can account for a significant portion of human gaze behavior in a naturalistic, interactive setting. These algorithms may be useful in the implementation of interactive virtual environments, both to predict the cognitive state of human operators, as well as to effectively endow virtual agents in the system with human-like visual behavior.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems—*Software Psychology*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Perceptual Reasoning*; I.3.3 [**Computer Graphics**]: Picture/Image Generation—*Viewing Algorithms*

General Terms: Algorithms, Experimentation, Human factors

Additional Key Words and Phrases: active vision, computational modeling, eye movements, immersive environments, video games, visual attention

1. INTRODUCTION

How do we decide where to look? The advent of convenient eye-tracking systems has opened the door to a greater understanding of the relationship between eye

Fig. 1. Four game frames with the eye position of one human (small cyan square, arrows) gazing back and forth between Mario and an adversary in a video game console system. Computational image-processing heuristics drawn from biological vision could help such systems predict and adapt to the behavior of their human operators.

movements and visual perception. At the same time, ever-increasing computer power has allowed our understanding of visual perception to be applied in progressively more powerful computational models of visual processing. Several nascent application domains will rely heavily on such models in the future, including: interactive computer graphics environments ("virtual reality" or video games, as in Figure 1), both in entertainment as well as more sober applications such as flight and driving simulators; visual prosthetic devices that will enhance the abilities for people with normal and impaired vision alike; machine vision systems that will automate mundane tasks, or provide a layer of redundancy for people performing safety-critical visual tasks.

One foundation for such advances will be a thorough understanding of the relationship between visual cognition and eye movements. Recent behavioral studies have shown that this relationship can be quite precise during the execution of a visually-guided behavioral task: the vast majority of fixations are directed to task-relevant locations, and fixations are coupled in a tight temporal relationship with other task-related behaviors such as reaching and grasping [Hayhoe 2004]. Furthermore, eye movements often provide a clear window into the mind of an observer; it is often possible to infer what a subject had in mind as he or she made a particular eye movement. For example, in a "block copying" task, where subjects had to replicate a physical assemblage made of elementary building blocks [Ballard et al. 1995], the observers' algorithm for completing the task was revealed by their pattern of eye movements: first select a target block in the model by fixating it, then find a matching block in the resource pool, then revisit the model to verify the block's position, then fixate the workspace to place the new block in the corresponding position. Other studies have used naturalistic interactive or immersive environments to give high-level accounts of gaze behavior in terms of objects, agents, "gist," and short-term memory [Yarbus 1967; Henderson and Hollingworth 1999; Rensink 2000; Land and Hayhoe 2001; Sodhi et al. 2002; Hayhoe et al. 2003; Bailenson and Yee 2005], to describe, for example, how task-relevant information guides eye movements while subjects make a sandwich [Land and Hayhoe 2001; Hayhoe et al. 2003] or how distractions such as setting the radio or answering a phone affect eye movements while driving [Sodhi et al. 2002].

While such perceptual studies have provided important constraints regarding goal-oriented high-level vision, additional work is needed to translate these descriptive results into fully-automated computational models that can be used in the application domains mentioned above. That is, although the block copying task

	MODEL TYPE	
STIMULUS TYPE	**qualitative**	**quantitative**
static, artificial *(gabor patches, search arrays)*	Treisman & Gelade 1980; Wolfe & Horowitz 2004	Rao et al 2002; Najemnik & Geisler 2005; Navalpakkam & Itti 2006
static, natural *(photographs)*	Yarbus 1967; Parker 1978; Mannan, Ruddock & Wooding 1997; Rayner 1998; Voge 1999	Privitera & Stark 1998; Reinagel & Zador 1999; Parkhurst, Law & Niebur 2002; Torralba 2003; Peters et al 2005; Navalpakkam & Itti 2005; Pomplun 2006
dynamic, natural *(movies, cartoons)*	Tosi, Mecacci & Pasqual 1997; May, Dean & Barnard 2003; Peli, Goldstein & Woods 2005	Carmi & Itti 2004; Itti & Baldi 2005
interactive, natural *(video games, flying or driving simulators, virtual reality, actual reality)*	Land, Mennie & Rusted 1999; Land & Hayhoe 2001; Hayhoe et al 2002; Hayhoe et al 2003	**this study**

Fig. 2. An overview of experimental approaches to gaze behavior. Studies can be roughly categorized according to the type of stimuli (rows), and according to the type of model used to account for observed gaze behavior (columns). The current study represents an entry into the lower-right corner of the table, using quantitative models to predict gaze behavior with naturalistic dynamic stimuli and an interactive task.

reveals observers' algorithm for completing the task, it does so only in the high-level language of "workspace" and "blocks" and "matching." In order for a machine vision system to replicate human observers' ability to understand, locate, and exploit such visual concepts, we need a "compiler" to translate such high-level language into the assembly language of vision—that is, low-level computations on a time-varying array of raw pixels. Unfortunately, a general computational solution to this task is tantamount to the "hard problem" of computer vision.

For that reason, computational models of eye movements to date have relied on simpler stimuli and tasks to demonstrate low-level regularities in eye movement behavior. For example, it has been shown that humans preferentially gaze towards regions with: multiple superimposed orientations (corners or crosses) [Zetzsche et al. 1998; Privitera and Stark 2000]; above-average spatial contrast (variance of pixel intensities) [Reinagel and Zador 1999], entropy [Privitera and Stark 2000], and texture contrast [Parkhurst and Niebur 2004]; and above-average "saliency" [Parkhurst et al. 2002; Peters et al. 2005] as computed by a biologically-inspired model of bottom-up attention [Itti et al. 1998]. Yet these results are typically obtained with static scenes (still images) and address only low-level, bottom-up image properties. Adding a temporal dimension and the realism of natural interactive tasks brings a number of complications in trying to realistically predict gaze targets with a computational model.

Figure 2 shows one way of categorizing existing research on gaze behavior along two dimensions. First, the type of stimuli used can vary from artificial "laboratory" stimuli to naturalistic three-dimensional environments; second, the type of "model" used to explain the behavior can be either qualitative (offering an explanation of

the behavior in high-level terms) or quantitative (explaining the behavior in terms of low-level computations performed on the raw visual input). No relative merit is ascribed to the different categories; rather, research in all of the categories will be needed to reach an eventual complete understanding of visual perception and gaze behavior. To a first approximation, research in areas toward the bottom and toward the right of the table relies more on modern computer power—either for rendering complex virtual environments or for implementing complex computational models—and thus has become practical only relatively recently.

Within the framework of Figure 2, the present study is an attempt to enter new territory for the first time: namely, to use purely computational models to predict gaze behavior in an interactive task with naturalistic, dynamic stimuli. Specifically, we chose to use several contemporary video games with simulated three-dimensional environments to provide the visual input and the interactive task (Figure 3), and we compared the recorded gaze behavior with the predictions of nine different bottom-up computational models ("heuristics") based on low-level image features (such as color, intensity, and motion) and saliency. It might intuitively be expected that volitional top-down influences should overwhelm reflexive bottom-up influences in gaze guidance, particularly in interactive environments with natural tasks, but instead our results demonstrate that in fact all but one of the computational bottom-up heuristics were significantly predictive of eye position. Heuristics sensitive to dynamic events were especially strong gaze predictors. Nevertheless, in this task there are clearly other influences on eye position that may be as strong or stronger than bottom-up effects. Our long-range methodology is to approach the prediction of eye position itself from the bottom up: first determine what can be explained by the simplest bottom-up models, then iteratively refine the models to account for residual behavior unexplained by the previous iteration. Along the way, the computational models will be guided by findings from other research categories in Figure 2.

2. METHODS

Figure 3 gives an overview of the data flow within our experiment, from the recording of eye movements and video frames while observers played video games, through subsequent analysis of the video frames by an array of computational heuristics, to final comparison of the predictions of the heuristics with the observed eye movements. Note that we use the term *heuristic* to refer to the various computational models that predict eye movements, and use the term *metric* to refer to different ways of quantifying the agreement between observed eye movements and heuristic predictions.

2.1 Eye-movement recordings

Five subjects (three male, two female) participated with informed consent, under a protocol approved by the Institutional Review Board of the University of Southern California. Subjects played four or five five-minute segments of off-the-shelf Nintendo GameCube games, including Mario Kart, Wave Race, Super Mario Sunshine, Hulk, and Pac Man World. Stimuli were presented on a 22" computer monitor (LaCie Corp; 640×480 pixels, 75 Hz refresh, mean screen luminance 30 cd/m^2, room 4 cd/m^2); subjects were seated at a viewing distance of 80 cm (28°× 21° usable

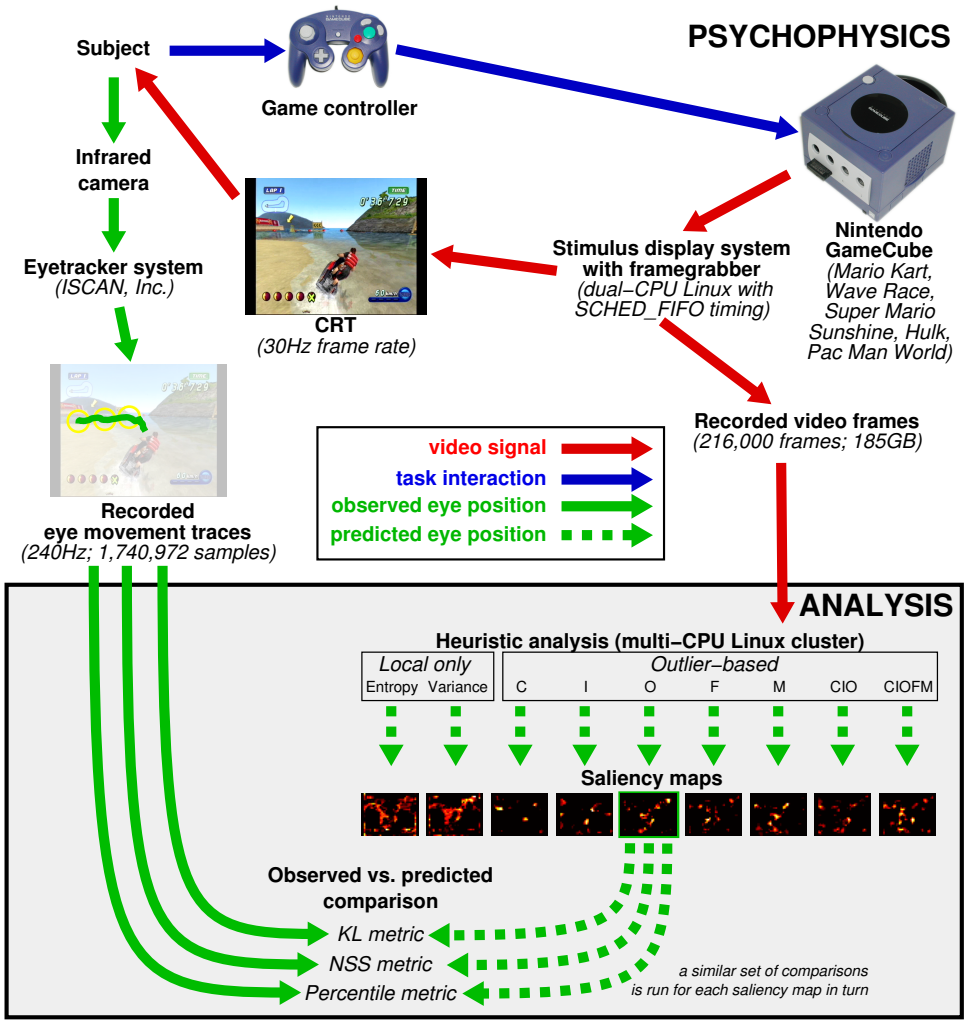

Fig. 3. An illustration of the overall flow of data within the eye-tracking psychophysics experiment and subsequent computational analysis. A commercial video game system (Nintendo GameCube) was used to generate video sequences (red arrows) which subjects viewed and controlled interactively (blue arrows). A separate computer was used to display video frames to the subject and simultaneously record those frames to disk for later analysis. Unlike studies which involve non-interactive tasks, the video sequences were generated "live" and were thus unique for each game-playing session. Subjects' eye position (green arrows) was recorded at 240Hz with an infrared eye tracker (ISCAN, Inc.). Following the psychophysics experiment, the sequence of video frames was passed to several different computational heuristics intended to predict likely eye position candidates (dashed green arrows; see Figure 5 for a detailed view of the operation of these heuristics). Finally, the observed and predicted eye positions were compared with three different metrics.

field-of-view) and rested on a chin-rest. To allow later analysis with our computational heuristics, the video game frames were grabbed, displayed and simultaneously recorded on a dual-CPU Linux computer under `SCHED_FIFO` scheduling to ensure

microsecond-accurate timing [Finney 2001]. Each subject's right eye position was recorded at 240Hz with a hardware-based eye-tracking system (ISCAN, Inc.). Each game segment was preceded by a calibration sequence, in which subjects fixated a series of points on an invisible 3×3 grid, to set up a correspondence between the eye camera and stimulus display coordinate systems [Stampe 1993]. After calibration, subjects fixated a central cross and pressed a key to start the game play. In post-processing, 8,449 saccades of amplitude 2° or more were extracted from 1,740,972 recorded eye position samples over 24 five-minute game-play sessions. The corresponding 216,000 recorded video frames (about 185 GB of raw pixel data) were processed on a cluster with 48 CPUs through the computational heuristics described next.

2.2 Computational heuristics

We tested nine heuristic models for predicting gaze targets. These fall into two broad groups, one in which only local features within small image patches are considered, and another in which global context is used to determine whether a local area is unusual and thus worthy of interest. This second group of heuristics (Figures 4 and 5) can be further subdivided into those that assess static features (which treat each frame individually, with no temporal context), and others that assess dynamic features. Finally, several heuristics can be combined together to form a new model that is different from the sum of its parts; in particular we tested two combined heuristics that correspond to "saliency." Figure 6a shows heuristic responses for several sample video frames.

2.2.1 *Local heuristics.* We evaluated two purely local heuristics that compute local variance and local entropy in 16×16 image patches, which have been previously shown to correlate with eye movements over still images [Reinagel and Zador 1999; Privitera and Stark 2000]. Variance for an image patch k is given by

$$V(k) \;=\; \frac{1}{N-1} \left(\sum_{(x,y) \in P_k} (I(x,y) - \overline{I_k})^2 \right) \tag{1}$$

where (x, y) loop over the $N = 16 \times 16$ set of pixel coordinates P_k that define patch k, $I(x, y)$ is the image intensity at (x, y), and $\overline{I_k}$ is the mean intensity over patch k. Entropy is similarly given by

$$E(k) \;=\; -\sum_{i \in G(k)} F_i(k) \log F_i(k) \tag{2}$$

where $F_i(k)$ is the frequency of occurrence of gray level i within the 16×16 patch k of interest, and $G(k)$ is the set of all gray levels present in the patch.

2.2.2 *Outlier-based heuristics.* We also evaluate heuristics which respond to image outliers in visual feature dimensions known to play a role in human attention [Wolfe and Horowitz 2004]. Our software implementation of these heuristics has been previously described [Itti et al. 1998; Itti and Koch 2001; Itti et al. 2003] and is freely available from our website (http://ilab.usc.edu/toolkit/). Figure 4 gives a detailed block diagram showing the sequence of the major computational steps,

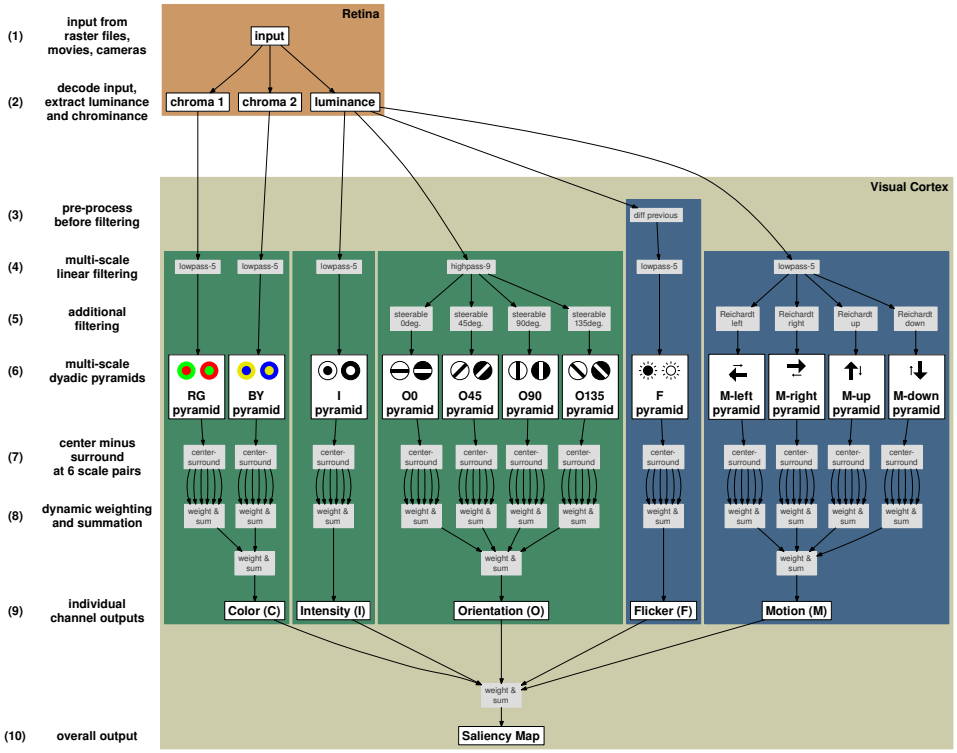

Fig. 4. A block diagram of the computational steps involved in the "outlier-based" bottom-up saliency model. See Figure 5 for an example of the model being run on a sample image. Each input frame is first decomposed by the model "Retina" into luminance and chrominance components. These then pass into the model "Visual Cortex" where the processing subsequently diverges into as many as twelve parallel feature channels grouped into five different feature types, including static features (green background): color (two subchannels), intensity, and orientation (four subchannels); and dynamic features (blue background): flicker, motion (four subchannels). Within each of these streams, the corresponding retinal output image is first decomposed into a multi-scale feature pyramid by an appropriate series of filters (such as for double-opponent color, or for Reichardt motion energy). Next, local spatial competition is used to emphasize feature contrast, by computing the difference maps between pairs of pyramid scales to form a set of center-surround maps. Finally these center-surround maps are dynamically weighted and summed across scales, forming a single conspicuity map for each feature channel. Another round of dynamic weighting and summation across feature types leads to a single saliency map, representing the output of the model. Different versions of the model can be formed by excluding all but one or a few of the the possible channels; for example a model with only static features would be formed by the C, I, and O channels.

and Figure 5 illustrates this sequence for a sample image by showing the intermediate maps generated by each computational step. In brief, this model processes the input through several parallel feature channels (color, intensity, orientation, flicker and motion) to generate feature maps at multiple scales which emphasize spatial outliers—locations with unusual feature values relative to their surroundings. Each map is then scaled by a dynamically determined weight such that a map with just

one or a few strong peaks receives a higher weight than a map with many peaks; in essence, maps with a more focused prediction of which locations are salient earn more votes in producing the final saliency map. The dynamically weighted maps are then summed together. In practice, there are several stages of dynamic weighting and summation, first within features and then across features, leading ultimately to a single saliency map.

We refer to Figure 4 for a more detailed breakdown of this process into 10 steps:

(1) RGB input from some source (camera, movie files, series of raster files) feeds into the model "Retina."

(2) The retina splits the RGB image into one luminance stream and two luminance-normalized opponent-color chrominance streams. These retinal outputs then diverge into 12 feature channels within the model "Visual Cortex," with the two chrominance components feeding two color-opponent channels ("C"; red/green and blue/yellow), and the luminance component feeding the intensity, orientation ("O"; 0°, 45°, 90°, 135°), flicker ("F") and motion ("M"; left, right, up down) channels.

(3) Optionally, each channel may pre-process its retinal output before filtering; this is currently used only by the flicker channel, to compute the absolute difference between the luminance images of the current and previous frames.

(4) Each image stream is decomposed into a multi-scale dyadic feature pyramid by recursive application of a linear filter followed by subsampling across nine scales, from level 0 at the original scale of the image, to level 8 at a 256-fold reduction in width and height. The C, I, F and M channels efficiently compute low-pass pyramids $\boldsymbol{L}$ from their respective inputs, using a separable filter composed of 5-point low-pass kernels in the x direction ($\boldsymbol{k}_x = [1, 4, 6, 4, 1]/16$) and in the y direction ($\boldsymbol{k}_y = \boldsymbol{k}_x^T$). The bottom pyramid level $\boldsymbol{L}_0$ is just the original image, and for $i > 1$ the i-th pyramid level is given by $\boldsymbol{L}_i = ((\boldsymbol{L}_{i-1} * \boldsymbol{k}_x) * \boldsymbol{k}_y) \downarrow_{x,y}$, where $*$ represents convolution and $\downarrow_{x,y}$ represents 2-fold subsampling in the x and y directions.

The O channel computes a high-pass ("Laplacian") pyramid $\boldsymbol{H}$ [Burt and Adelson 1983], essentially by computing the difference between parallel full-band $\boldsymbol{F}$ and low-passed $\boldsymbol{L}$ pyramids. The computation is initialized at level 0 with $\boldsymbol{F}_0$ given by the original luminance image. Then, at each level the low-pass image is computed by convolving the full-band image with a separable filter composed of 9-point low-pass kernels in the x and y directions: $\boldsymbol{L}_i = (\boldsymbol{F}_i * \boldsymbol{k}_x) * \boldsymbol{k}_y$, with $\boldsymbol{k}_x = [1, 8, 28, 56, 70, 56, 28, 8, 1]/256$ and $\boldsymbol{k}_y = \boldsymbol{k}_x^T$. Finally, the high-pass image is the difference of the full-band and low-passed images: $\boldsymbol{H}_i = \boldsymbol{F}_i - \boldsymbol{L}_i$. The process iterates to higher pyramid levels by subsampling the current low-passed image to give the next level's full-band image: $\boldsymbol{F}_{i+1} = \boldsymbol{L}_i \downarrow_{x,y}$.

(5) A post-processing step is used by the O and M channels to extract the features of interest from the pyramid. In the orientation channels, an efficient steerable filter implementation is used to produce 0°, 45°, 90° and 135° orientation-tuned responses from the high-pass pyramid [Greenspan et al. 1994]. This involves modulating each level of the pyramid with a quadrature pair of oriented sine wave gratings offset by 90° in phase, convolving each modulated image with

a 9-point low-pass kernel, and finally computing the complex magnitude of the image pair. In the motion channel, a simple Reichardt motion detector is used to compute motion energy in the left, right, up and down directions. For each time t, pyramid level i, spatial location $\vec{p} = (x, y)$, and motion direction $\vec{d} = (\Delta x, \Delta y)$, the motion energy map is computed from the low-pass pyramid L as

$$\boldsymbol{E}^t_{i,\vec{d}}(\vec{p}) \;=\; \left\lfloor \left(\boldsymbol{L}^t_i(\vec{p}) \cdot \boldsymbol{L}^{t-1}_i(\vec{p}+\vec{d}) \right) - \left(\boldsymbol{L}^{t-1}_i(\vec{p}) \cdot \boldsymbol{L}^t_i(\vec{p}+\vec{d}) \right) \right\rfloor, \qquad (3)$$

where $\lfloor \cdot \rfloor$ represents half-wave rectification to exclude negative responses to motion in the anti-preferred direction.

(6) At this intermediate stage there are now 12 pyramids, each with 9 images, representing local feature content for color, intensity, orientation, flicker and motion features.

(7) A center-surround operator is used to detect outliers, or locations whose feature values are different from those at surrounding locations and surrounding spatial scales. This is implemented by computing point-wise differences between different pairs of pyramid scales, for combinations of three center scales ($c = \{2, 3, 4\}$) and two center-surround scale differences ($s = c + \delta$, $\delta = \{3, 4\}$); thus, six center-surround maps are computed for each of the 12 features, giving 72 maps at this stage. Note that surround image will naturally be smaller than the center image as it is drawn from a higher level in the dyadic pyramid, so when computing each center-surround map the surround image is bilinearly upscaled to the size of the center image in order to allow for the point-wise difference operation.

(8) Each center-surround map is subjected to convolution with a broad difference-of-Gaussians filter, global inhibition and half-wave rectification, in several iterations [Itti and Koch 2001]. This has the effect of enhancing sparse peaks and suppressing cluttered peaks, as well as scaling the overall map peak in proportion to the sparseness of the map. In this way, initially noisy feature maps can be reduced to sparse representations of only outlier locations which strongly stand out from their surroundings. The processed center-surround maps are then summed together to form a single output from each of the 12 individual feature channels.

(9) The overall output maps for each feature type are formed. For the I and F channels, the maps simply pass through from the previous step. Within the C, O, and M channels, each having multiple subchannels, the subchannel outputs are processed with the same dynamic iterative normalization procedure, and summed together to produce the channel output map.

(10) Finally, the channel output maps are processed once more with the iterative normalization procedure and summed together to form the final output saliency map.

Since each of the feature channels operates independently, it is possible to test individual channels or channel subsets in isolation. We hence evaluate five heuristics sensitive to outliers in the general dimensions of color (C), intensity (I), orientation

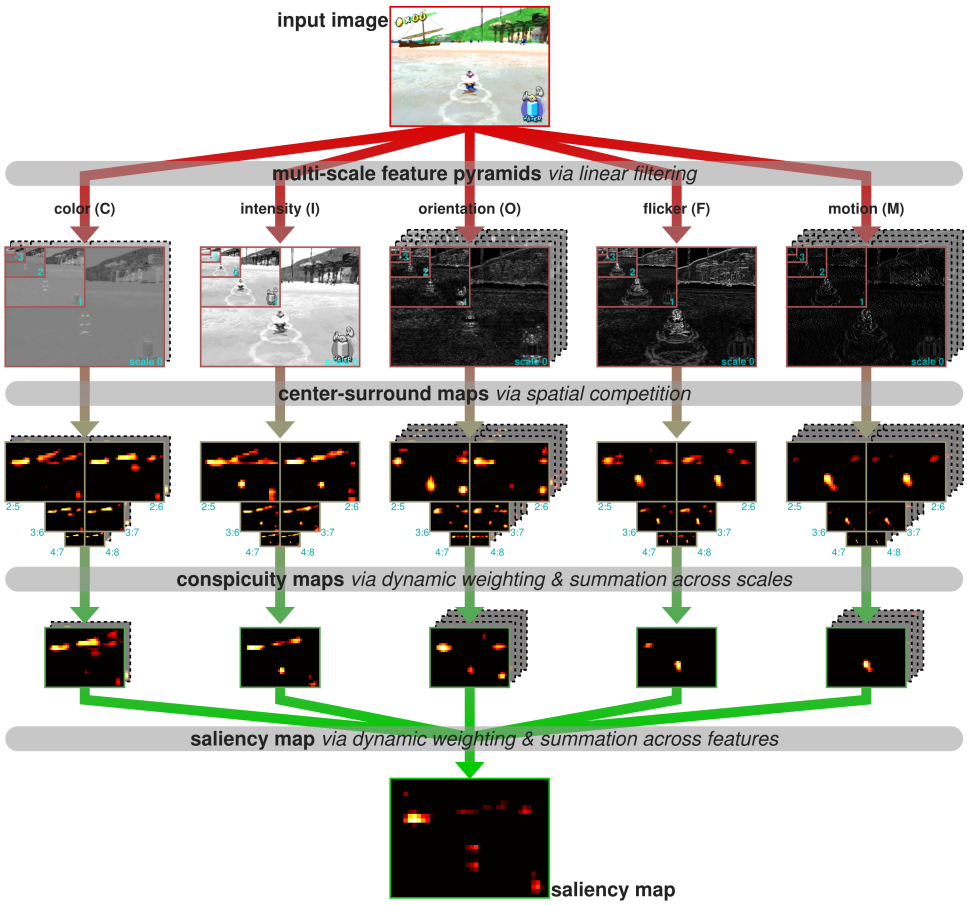

Fig. 5. An example of the intermediate maps involved in processing a sample image with the "outlier-based" bottom-up saliency model. Row 1: **input image** sent through the model. Row 2: **multi-scale feature pyramids** constructed by filtering the input for five feature types (color, intensity, orientation, flicker, motion). For illustration, each pyramid scale is shown nested inside the one beneath it. Row 3: **center-surround maps** constructed as the differences between pairs of pyramid scales; for example, "3:7" indicates that pyramid scale 3 is the center and pyramid scale 7 is the surround. Row 4: **conspicuity maps** constructed by dynamic weighting and summation of the center-surround maps. Row 5: final **saliency map** constructed by dynamic weighting and summation of the conspicuity maps. See Figure 4 for a more detailed diagram of the computational steps involved.

(O), flicker (F), and motion (M), plus one that combines intensity, color, and orientation into a measure of static saliency (C,I,O), and one that adds motion and flicker to yield a measure of full static/dynamic saliency (C,I,O,F,M).

2.2.3 *Computational efficiency of the heuristics.* For producing comparisons with human eye movements in this study, we used heuristic implementations that were designed with a primary goal of being useful research tools, responsible not only for generating the heuristic response maps, but also for comparing these maps

Table I. CPU time required to generated heuristic response maps for a 640×480 color input image, averaged over 1000 frames, running in a single thread on a GNU/Linux system (Fedora Core 6) with a 2.8GHz Intel Xeon processor. Times are shown for two implementations, one based on floating-point arithmetic and one based on integer arithmetic, which produce nearly identical response maps. Note that the combined heuristics CIO and CIOFM can be computed in ∼17% less time than it takes to compute their subcomponents independently, because in the combined models some of the intermediate computations can be shared between subcomponents.

heuristic	**floating-point arithmetic**		**integer arithmetic**		**% time diff.**
	time per frame	frame rate	time per frame	frame rate	(int-fp)/fp
E	14.12ms	70.81Hz	–	–	–
V	4.57ms	218.77Hz	–	–	–
I	4.95ms	202.22Hz	4.33ms	231.21Hz	-12.5%
C	27.64ms	36.17Hz	21.17ms	47.24Hz	-23.4%
O	25.06ms	39.91Hz	19.60ms	51.02Hz	-21.8%
CIO	51.34ms	19.48Hz	40.28ms	24.83Hz	-21.5%
F	9.86ms	101.43Hz	8.32ms	120.25Hz	-15.7%
M	13.11ms	76.27Hz	12.00ms	83.31Hz	-8.5%
CIOFM	66.70ms	14.99Hz	54.37ms	18.39Hz	-18.5%

with human eye traces, and for allowing a high degree of introspection into the heuristic's inner workings. These additional elements tend to dominate the computational time required to run the heuristic models, which is why the results we report here were generated on a cluster of 48 CPUs. Nevertheless, in order for the heuristics to ultimately be used in interactive virtual environments, they must be able to be implemented efficiently, and indeed the core algorithms can run at 15 frames per second or better on a single modern CPU core. We profiled two implementations, one that relies on the same floating-point arithmetic as in our research implementation, and a second one with all computations ported to fixed-point integer arithmetic. Table I shows the average per-frame computation times required to generate each of the various heuristic maps for a stream of 640×480 color input images, on a GNU/Linux system (Fedora Core 6) with a 2.8GHz Intel Xeon processor. These timings provide a rough estimate suggesting that the heuristics could feasibly be used as part of a larger interactive virtual environment using current commodity hardware.

2.3 Metrics for scoring the heuristics

Figure 6a shows human saccade targets superimposed on the response maps from several of the heuristics for a few sample frames. For illustration purposes only, those saccade targets are labeled as "hit," "weak hit," or "miss." Those labels are based on the intuition that a response map represents a good eye movement prediction if: (1) it has a strong peak in the neighborhood of the human saccade target, and (2) it has little activity elsewhere. Conversely, there are two ways for a model to fail: it may either have low response values near the saccade target, or it may have high response values everywhere else, rendering useless its predictions near the target. The first type of failure is exhibited by the misses in the color and flicker heuristics in the figure; the second type is shown by the entropy heuristic which always has widespread activity preventing it from achieving a strong hit.

The remainder of this section describes several approaches to quantifying these intuitions about what characterizes a good eye movement predictor. Each heuristic generates a topographic dynamic master response map $S(\boldsymbol{x};t)$, which is a time-varying two-dimensional array that assigns a response value to each spatial location $\boldsymbol{x} = (x, y)$ at each video frame time t; we will formalize our intuitions by saying that a good heuristic should generate response maps in which the values at locations fixated by observers are in some way statistically discriminable from the values at non-fixated or random locations. To quantify this relationship, we used three different metrics, of the form $M(S(\boldsymbol{x};t))$, which generate a scalar value indicating how well a particular heuristic matched a series of eye positions. The motivation for analyzing the heuristics with more than one metric is to ensure that the main qualitative conclusions are independent of the choice of metric. For example, some metrics may be invariant to reparameterizations—for instance, $M(S) = M(\sqrt{S}) = M(e^S)$—while others are not, so comparing the heuristic scores with these different metrics can help reveal whether any particular nonlinearity is likely to play an important role in the generation of eye movements from the master response maps. Some metrics have well-defined upper or lower bounds. All of the metrics are shift-invariant ($M(S) = M(S + c)$) and scale-invariant ($M(S) = M(cS)$).

2.3.1 *Kullback-Leibler (KL) distance.* At the onset of each human saccade ($t_i = 1 \ldots N$, N = number of saccades in the experimental session), we sample the heuristic's master map activity around the saccade's future endpoint $\boldsymbol{x}_{i,\text{human}}$ and around a uniformly random endpoint $\boldsymbol{x}_{i,\text{random}}$:

$$h(t_i) = \frac{S(\boldsymbol{x}_{i,\text{human}};t_i) - \min_{\boldsymbol{x}} S(\boldsymbol{x};t_i)}{\max_{\boldsymbol{x}} S(\boldsymbol{x};t_i) - \min_{\boldsymbol{x}} S(\boldsymbol{x};t_i)} \tag{4}$$

$$r(t_i) = \frac{S(\boldsymbol{x}_{i,\text{random}};t_i) - \min_{\boldsymbol{x}} S(\boldsymbol{x};t_i)}{\max_{\boldsymbol{x}} S(\boldsymbol{x};t_i) - \min_{\boldsymbol{x}} S(\boldsymbol{x};t_i)} \tag{5}$$

Note that by construction, $h(t_i) \in [0, 1]$ and $r(t_i) \in [0, 1]$ for all t_i. We then form histograms of these values, with 10 bins B_k covering the range $[0, 1]$, across all saccades:

$$B_k = \begin{cases} \left[\frac{k-1}{10}, \frac{k}{10}\right) & \text{if } 1 \le k \le 9 \\ \left[\frac{k-1}{10}, \frac{k}{10}\right] & \text{if } k = 10 \end{cases} \tag{6}$$

$$H_k = \frac{1}{N} \left|\{t_i : h(t_i) \in B_k\}\right| \tag{7}$$

$$R_k = \frac{1}{N} \left|\{t_i : r(t_i) \in B_k\}\right| \tag{8}$$

where $|\cdot|$ indicates set size. Finally we quantify the difference between these histograms with the (symmetric) Kullback-Leibler (KL) distance (also known as relative entropy):

$$KL = \frac{1}{2} \sum_{k=1}^{10} \left(H_k \log \frac{H_k}{R_k} + R_k \log \frac{R_k}{H_k} \right) \tag{9}$$

In practice, we repeat this computation 1200 times, each time comparing the fixated location $\boldsymbol{x}_{i,\text{human}}$ with an $\boldsymbol{x}_{i,\text{random}}$ location at one of the $(640/16)\times(480/16) = 1200$

points in the heuristic response map, and finally we compute the mean and standard deviation of the *KL* distance across those repetitions (Figures 7a and 7d show these mean ± *s.d.* scores).

Heuristics which better predict human scanpaths exhibit higher *KL* distances, since observers typically gaze towards a non-uniform minority of regions with the highest heuristic responses while avoiding the majority of regions with low heuristic responses (see Figure 6b). The *KL* distance offers several advantages over simpler scoring schemes [Reinagel and Zador 1999; Parkhurst et al. 2002]: (1) *KL* is agnostic about the mechanism for selecting a saccade given the instantaneous response map $S(\boldsymbol{x}; t)$—other metrics essentially measure the rightward shift of the H_k histogram relative to the R_k histogram, whereas *KL* is sensitive to any difference between the histograms; and (2) *KL* is invariant to reparameterizations, such that applying any continuous monotonic nonlinearity to master map values does not affect scoring. One disadvantage of the *KL* distance is that it does not have a well-defined upper bound—as the two histograms become completely non-overlapping, the *KL* distance approaches infinity.

2.3.2 *Percentile.* Like the *KL* distance, a simple percentile metric is also invariant to reparameterizations. In contrast to *KL*, the percentile metric has a well-defined upper bound (100%) and "chance" mid-point (50%); the percentile metric also assumes that higher heuristic responses correlate with higher likelihood of fixation, unlike the *KL* distance. Also, whereas the *KL* distance is computed across the entire set of saccades at once (and then repeated and averaged across different random samplings), the percentile metric is computed separately *for each saccade*, and then averaged across all saccades. The metric is defined in terms of the number of locations in the heuristic response map with values smaller than that of the saccade's future endpoint $\boldsymbol{x}_{i,\text{human}}$:

$$P(t_i) \;=\; 100 \cdot \frac{|\{\boldsymbol{x} \in \boldsymbol{X} : S(\boldsymbol{x}; t_i) < S(\boldsymbol{x}_{i,\text{human}}; t_i)\}|}{|\boldsymbol{X}|} \tag{10}$$

where $|\cdot|$ indicates set size and $\boldsymbol{X}$ is the set of all locations in the heuristic response map. In practice, we compute a percentile for each saccade, and then determine the mean and standard error across this set of percentile scores (Figures 7c and 7f show these mean ± *s.e.m.* scores).

2.3.3 *Normalized scanpath saliency (NSS).* The normalized scanpath saliency [Peters et al. 2005] is defined as the response value at a saccade's future endpoint $\boldsymbol{x}_{i,\text{human}}$ in a heuristic response map that has been normalized to have zero mean and unit standard deviation:

$$\mu_S(t_i) \;=\; \frac{1}{|\boldsymbol{X}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}} S(\boldsymbol{x}; t_i) \tag{11}$$

$$\sigma_S(t_i) \;=\; \sqrt{\frac{1}{|\boldsymbol{X}| - 1} \sum_{\boldsymbol{x} \in \boldsymbol{X}} \left(S(\boldsymbol{x}; t_i) - \mu_S(t_i)\right)^2} \tag{12}$$

$$\text{NSS}(t_i) \;=\; \frac{1}{\sigma_S(t_i)} \left(S(\boldsymbol{x}_{i,\text{human}}; t_i) - \mu_S(t_i)\right) \tag{13}$$

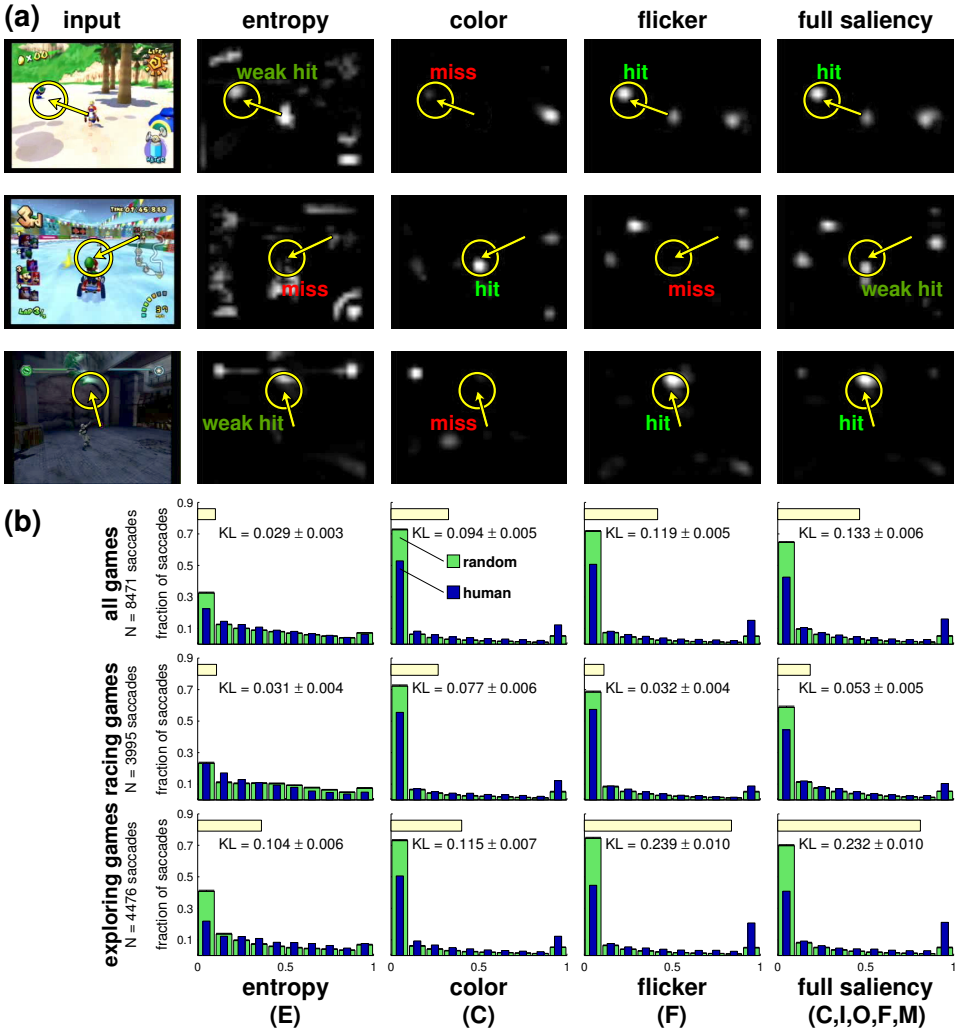

Fig. 6. Scoring different heuristics for their ability to predict human gaze targets during video-game playing. **(a)** Each sample video frame (column 1) was chosen at a time when the subject is initiating a saccade from the base of the arrow to the tip of the arrow surrounded by the circle. We sample the master maps generated by each heuristic (columns 2–5) in the neighborhood of the saccade target and at a number of uniformly random locations. For illustration only (not for quantitative analysis), each target is labeled here as "hit" (strong peak at target with few other peaks in the map), "weak hit" (weak peak within target circle with many other map peaks) or "miss" (no peak at target location). Individual heuristics often have misses (such as color and flicker here), while a combined heuristic such as full saliency is more likely to have at least a weak hit. **(b)** Across our entire data set, these histograms show how frequently locations with particular heuristic values are the targets of human saccades (narrow, dark bars), and of random saccades (light, wide bars). The Kullback-Leibler (KL) distance (light horizontal bars atop each subpanel) quantifies the dissimilarity between these distributions; higher KL values indicate that the heuristic is better able to distinguish human fixation locations from other locations. Top row: heuristic scores across all game types; middle row: scores for racing games only; bottom row: scores for exploration games only. See Figure 7 for a more comprehensive summary of the results.

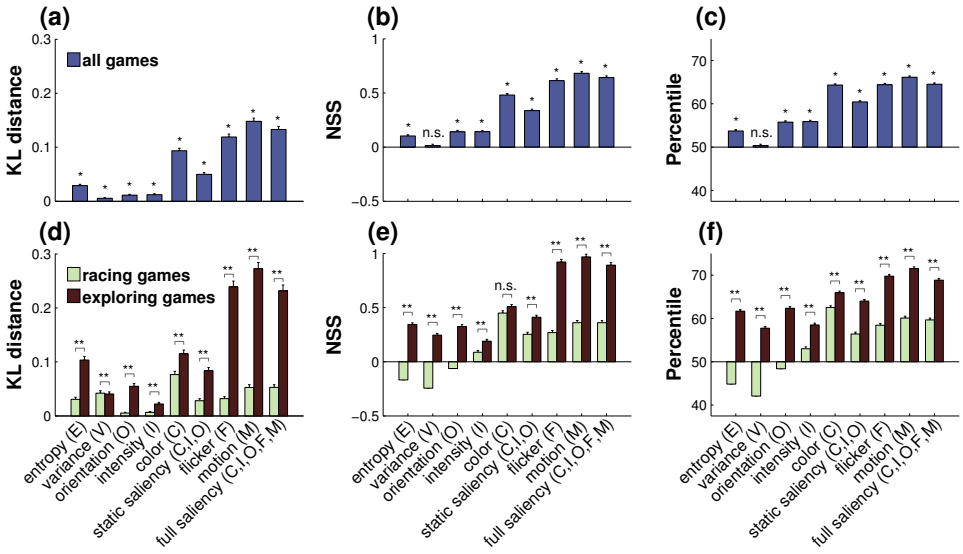

Fig. 7. Nine heuristics for eye position prediction were scored by three metrics — **(a,d)** KL distance (section 2.3.1), **(b,e)** normalized scanpath saliency (NSS), and **(c,f)** percentile — to quantify how well the heuristics discriminate human gaze targets from random locations. For each metric, a higher score means a better fit between predicted and observed eye position. **(a,b,c)** Across all game sessions, the dynamic features motion (M) and flicker (F) scored better than static features color (C), intensity (I) and orientation (O), and local heuristics entropy (E) and variance (V). Bars marked with $(*)$ represent heuristics that scored significantly above the chance level in that metric (1-tailed t-test, $p < 0.0001$ or better), while "n.s." indicates a non-significant result. **(d,e,f)** When the sessions were split by game paradigm, all heuristics (except variance with the KL metric) were better predictors of human gaze in exploration games than an racing games. Bars marked with $(**)$ represent heuristics that showed a significant difference in metric score between the two game types (2-tailed paired t-test, $p < 0.0001$ or better; note that the paired t-test reveals significant differences even in cases where the two scores are very similar between the game types, even with overlapping error bars); "n.s." indicates a non-significant result.

Like the percentile metric, NSS is computed once for each saccade, and subsequently the mean and standard error are computed across the set of NSS scores (Figures 7b and 7e show these mean $\pm$ *s.e.m.* scores). By virtue of the zero-mean normalization, NSS has a well-defined "chance" mid-point (0). Unlike KL and percentile, NSS is not invariant to reparameterizations.

3. RESULTS

Figure 7a–c shows results obtained overall. Each of the three analysis metrics (KL distance, percentile, NSS) revealed similar qualitative patterns of results. All nine heuristics tested performed significantly above the chance level according to all three metrics (t-tests, $p < 0.0001$ or better, for $KL > 0$, NSS > 0, or percentile > 50), with the exception of variance (V) scored by the NSS and percentile metrics.

Under our conditions of interactive game-play and with our video game stimuli, we found that motion alone (M) was the best predictor of human gaze, with flicker alone (F) and the full saliency heuristic (C,I,O,F,M) scoring nearly as well. Inter-

mediate scores were obtained with color alone (C) and static saliency (C,I,O). The poorest scores came from static features orientation alone (O) and intensity alone (I), and from local heuristics computing variance (V) and entropy (E). Notably, color alone (C) scored much better than the other static features. Studies with static scenes have found that the relative gaze-attraction strength of the color feature varies with image type [Parkhurst et al. 2002]; in our case the strength of the color feature is likely explained at least in part by the art design of the video games which includes strong color contrast between main characters and the background.

We found two interesting dissociations within the scores obtained by the different heuristics. A split by game type (Figures 7d–f) suggested an interesting dissociation between racing games (Mario Kart and Wave Race), and exploration games (Super Mario Sunshine, Hulk, Pac Man World). Racing games involve primarily obstacle-avoidance while the player navigates a predetermined course under time pressure. In contrast, exploration games, with their more open-ended storylines, impose fewer restrictions on the player's navigation, but instead implicitly require the player to search the environment for objects or other characters with which to engage. All of the heuristics scored better at predicting observers' gaze during exploration games than during racing games (paired t-tests, $p < 0.0001$ or better), with the two exceptions of variance (V) by the KL distance and color (C) by NSS. In addition to this overall effect, we also found that the top-scoring heuristics were different for racing and exploring games. For the exploration games, the best heuristics were the dynamic outlier-based features motion (M) and flicker (F), followed closely by full saliency (C,I,O,F,M). In contrast, for the racing games, the best heuristic was instead the static feature color (C). In a second split, we found that there was less variability in heuristic performance across subjects than across games (Figure 8), especially so for the overall best heuristics motion (M) and flicker (F).

4. DISCUSSION

In this study we have quantitatively evaluated nine simple computational heuristic models for their ability to predict where people look while playing video games. We found that all heuristics score significantly above chance, hence representing easily computable shortcuts to the targets selected by human gaze. This finding was robust, with the same qualitative pattern revealed by three different metrics for scoring the heuristics. Our study provides direct experimental evidence that bottom-up image analysis can predict a non-negligible fraction of human gaze targets, even in situations where top-down influences are expected to dominate, driving eye movements based on task demands rather than on visual inputs. Still, by one measure, purely bottom-up factors make a relatively smaller contribution to eye movements in the interactive video game stimuli (NSS score for static saliency C,I,O: 0.34) than in static natural scenes (NSS score for "baseline salience model," equivalent to static saliency: 0.69 [Peters et al. 2005]). Part of that gap is bridged with the inclusion of the dynamic bottom-up features motion (M) and flicker (F) in the full saliency heuristic (NSS score: 0.64). In this study, the full saliency heuristic was actually slightly outperformed by the motion channel alone (NSS score: 0.68), which makes sense given the highly dynamic nature of the video game stimuli that we used. Still, we would suggest that the full saliency model, with both static and

dynamic features, is a better choice for a general-purpose bottom-up heuristic, since it can account well for eye movements in both static and dynamic scenes, whereas the motion channel alone would obviously fail completely for static scenes. In the human visual system it may be that the different channels are weighted dynamically according to their relevance to the current visual input, so that in a video game task, for example, the predictions of the motion channel could be emphasized and those of the static channels deemphasized.

The main failure mode of the heuristics that we tested is that they often fail to highlight objects or regions that, to any human observer, would be obviously task-relevant; this is reflected in preliminary unpublished results indicating that the heuristics are better at predicting the gaze of observers passively viewing pre-recorded video game clips than of those interactively playing the games. For example, in racing games, the goal of successful navigation requires game players to focus on the horizon and the focus of expansion of the oncoming track, yet that point is often not the most salient according to the bottom-up heuristics. This may partly explain why the bottom-up heuristics perform more poorly on the racing games than on the exploring games (Figures 7d–f): the racing games may simply involve a stronger set of task-dependent biases on eye position. Similarly, in order to perform well at the game-playing task, observers must often effectively ignore certain regions that are actually quite salient by bottom-up measures, such as a constantly ticking digital clock in one corner of the screen. Such observations, while simple to describe, have yet to be translated into computational terms that could be implemented in a gaze-prediction heuristic. Thus, one direction for future study is to operationalize the top-down, task-related influences that are outside the scope of bottom-up heuristics. Such influences are likely to be found in studies from the interactive-stimulus/qualitative-model category of Figure 2, which suggest behavioral phenomena that might be candidates for future implementation and testing in the quantitative-model category.

One limitation of the heuristics that we tested is that they stop short of making a specific saccade predictions—that is, they don't predict *when* a saccade will occur, nor to which single location a saccade will occur. Rather, they make a continuous prediction of how likely each location would be to be a saccade target, if a saccade were to be triggered at that moment; then, at those instants where human observers in fact generated saccades, we test the predictions of the heuristic. In terms of the bottom-up visual attention system proposed in [Koch and Ullman 1985; Itti et al. 1998] in which a saliency map (SM) interacts with dynamic winner-take-all (WTA) and inhibition-of-return (IOR) mechanisms to select a series of locations, the current study has focused exclusively on the saliency map stage, ignoring WTA and IOR mechanisms. This is largely a practical decision: we want to fine-tune an SM model that does well at predicting eye movements on average, before testing WTA and IOR implementations for predicting specific eye movements; otherwise, if a combined SM/WTA/IOR was found to produce poor results it would be difficult to know whether the SM or WTA/IOR mechanisms were to blame. An important topic for future study is to a develop a WTA/IOR model capable of producing accurate eye movement predictions with high temporal and spatial precision. Such a model will likely rely on significantly more complex visual features including knowl-

edge about objects, since human decisions about when and where to saccade and when to release fixation are influenced not just by low-level bottom-up information such as saliency, but also by top-down influences from the observer's current goals [Navalpakkam and Itti 2005], such as a tendency to maintain fixation until the currently fixated object has been fully analyzed and recognized.

Despite these limitations, we believe that heuristics of the kind we have presented here may nevertheless be useful in a range of potential applications. In particular, many interactive graphics environments are starting to share a paradigm in which one or more human operators participate in a virtual world along with a potentially large number of virtual agents. This paradigm applies to mass-market video games, as well as high-end simulators/trainers for flying or driving, and large-scale combat simulators. As with computer systems in general, the shared environment is often "seen" in different ways by human and virtual agents; the human participants sense the environment through visual and auditory displays, while the virtual agents typically have direct access to some higher-level representation (such as an object-oriented scene graph) that is used to render the final visual and auditory signals. Our finding that the heuristics tested here significantly correlate with actual gaze position provides empirical validation that previously proposed visual systems for virtual agents indeed yield outputs that correlate with human behavior [Terzopoulos and Rabie 1997; Itti et al. 2003; Peters and O'Sullivan 2003]. Furthermore, these results suggest two possible applications for interactive virtual environments. First, the system could better predict the behavior of its human operator(s) by becoming "attention-aware" [Toet 2006], allowing it to tailor the visual display appropriately (for example, by placing important information near the likely focus of attention, or by avoiding distracting displays during important moments). This application is aided by the fact that heuristic performance varied more across games than across subjects (Figure 8), suggesting that heuristics could be usefully tuned to particular virtual environments, without needing to also be tuned to individual human observers. Second, the system could better mimic human behavior, allowing the virtual peers of the human operator to more naturally determine where to look next in any virtual, real, or mixed environment. Although an analysis of a computer-graphics scene graph might allow prediction of potentially interesting gaze targets without requiring processing of fully rendered images, such an approach is limited to environments that are entirely virtual. Accounting for humans interacting with such environments becomes possible with heuristics that do not require knowledge of the scene graph and can instead operate at the pixel level.

Our main contributions have been two-fold. First, we studied *dynamic, interactive* video scenes whereas previous work on computational mechanisms for gaze direction has largely focused on still images or pre-recorded movies. Although the visual stimuli here were artificially generated, we believe they are similar enough to natural scenes to suggest that it is likely that the heuristics described here would also serve well in predicting gaze in truly natural scenes. Second, and perhaps most importantly, each of the heuristics that we tested has a tractable *computational implementation* that takes a time-varying 2-D pixel array as input and produces a time-varying prediction of where a human observer's gaze might be directed. Any

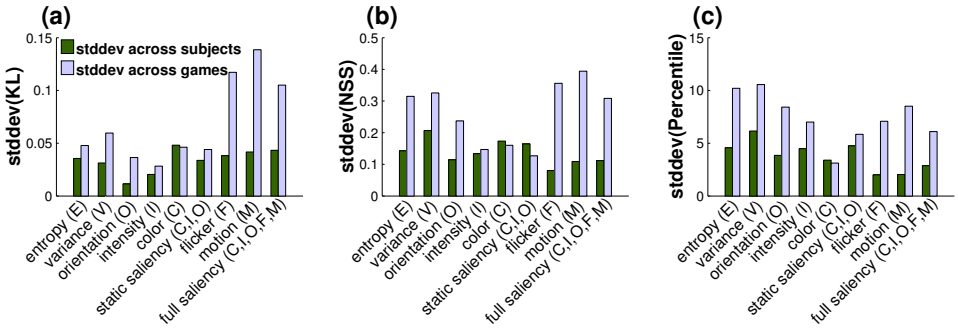

Fig. 8. Sessions were grouped either by subject or by game, then the standard deviation of the heuristic scores was computed within each group, for each of the three metrics: **(a)** *KL* distance, **(b)** normalized scanpath saliency (NSS), and **(c)** percentile. For almost all of the heuristics, there was higher variability due to game than due to the subject.

apparent high-level behavior exhibited by the heuristic is simply attributable to low-level processing of the input; that is, the heuristics do not require the visual input to be accompanied by explicit high-level labels for "objects" or "targets." Whereas behavioral studies have made great strides in understanding observers' gaze behaviors in just such object-based or agent-based terms [Land and Hayhoe 2001; Hayhoe et al. 2002; Hayhoe et al. 2003; Bailenson and Yee 2005], such findings become available to artificial visual systems only when there is an algorithm to extract the high-level information directly from the visual input. Accurate object recognition and semantic interpretation continues to be a "hard problem" in artificial vision, particularly for unconstrained visual input; while our approach has the drawback of lacking high-level scene understanding, it has the virtue of a straightforward computational implementation that can be applied to any visual input.

Acknowledgments. Support for this work was provided by the National Geospatial-Intelligence Agency (NGA) and the Directorate of Central Intelligence (DCI). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

REFERENCES

Bailenson, J. and Yee, N. 2005. Digital chameleons: Automatic assimilation of nonverbal gestures in immersive virtual environments. *Psychological Science 16*, 814–819.

Ballard, D., Hayhoe, M., and Pelz, J. 1995. Memory representations in natural tasks. *Journal of Cognitive Neuroscience 7,* 1 (Winter), 66–80.

Burt, P. and Adelson, E. 1983. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31,* 4, 532–540.

Carmi, R. and Itti, L. 2004. Bottom-up and top-down influences on attentional allocation in natural dyamic scenes. In *Proc. Vision Science Society Annual Meeting (VSS04).* 20.

Finney, S. A. 2001. Real-time data collection in linux: A case study. *Behavior Research Methods, Instruments, and Computers 33*, 167–173.

Greenspan, H., Belongie, S., Goodman, R., Perona, P., Rakshit, S., and Anderson, C. 1994. Overcomplete steerable pyramid filters and rotation invariance. In *Proceedings Computer Vision and Pattern Recognition*. IEEE Computer Society, 222–228.

Hayhoe, M. 2004. Advances in relating eye movements and cognition. *Infancy 6,* 2, 267–274.

Hayhoe, M., Ballard, D., Triesch, J., and Shinoda, H. 2002. Vision in natural and virtual environments. In *Proceedings of the symposium on Eye Tracking Research & Applications (ETRA)*. 7–13.

Hayhoe, M., Shrivastava, A., Mruczek, R., and Pelz, J. 2003. Visual memory and motor planning in a natural task. *Journal of Vision 3,* 1, 49–63.

Henderson, J. M. and Hollingworth, A. 1999. High-level scene perception. *Annual Review of Psychology 50*, 243–271.

Itti, L. and Baldi, P. 2005. A principled approach to detecting surprising events in video. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. San Siego, CA, 631–637.

Itti, L., Dhavale, N., and Pighin, F. 2003. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proceedings of SPIE 48th annual international symposium on optical science and technology*. 64–78.

Itti, L. and Koch, C. 2001. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging 10,* 1 (January), 161–169.

Itti, L., Koch, C., and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20,* 11 (November), 1254–1259.

Koch, C. and Ullman, S. 1985. Shifts in selective visual-attention—towards the underlying neural circuitry. *Human Neurobiology 4,* 4, 219–227.

Land, M. and Hayhoe, M. 2001. In what ways do eye movements contribute to everyday activities? *Vision Research 41,* 25-26, 3559–3565.

Land, M., Mennie, N., and Rusted, J. 1999. The roles of vision and eye movements in the control of activities of daily living. *Perception 28,* 11, 1311–1328.

Mannan, S., Ruddock, K., and Wooding, D. 1997. Fixation patterns made during brief examination of two-dimensional images. *Perception 26,* 8, 1059–1072.

May, J., Dean, M., and Barnard, P. 2003. Using film cutting techniques in interface design. *Human-Computer Interaction 18,* 4, 325–372.

Najemnik, J. and Geisler, W. 2005. Optimal eye movement strategies in visual search. *Nature 434,* 7031 (March), 387–391.

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

Navalpakkam, V. and Itti, L. 2006. Optimal cue selection strategy. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*. MIT Press, Cambridge, MA, 1–8.

Parker, R. 1978. Picture processing during recognition. *Journal of Experimental Psychology—Human Perception and Performance 4,* 2, 284–293.

Parkhurst, D., Law, K., and Niebur, E. 2002. Modeling the role of salience in the allocation of overt visual attention. *Vision Research 42,* 1, 107–123.

Parkhurst, D. and Niebur, E. 2004. Texture contrast attracts overt visual attention in natural scenes. *European Journal of Neuroscience 19,* 3 (February), 783–789.

Peli, V., Goldstein, T., and Woods, R. 2005. Scanpaths of motion sequences: where people look when watching movies. In *Proceedings of the Fourth Starkfest Conference on Vision and Movement in Man and Machines*. School of Optometry, UC Berkeley, Berkeley, CA, 18–21.

Peters, C. and O'Sullivan, C. 2003. Bottom-up visual attention for virtual human animation. In *Computer Animation and Social Agents 2003*. 111–117.

Peters, R., Iyer, A., Itti, L., and Koch, C. 2005. Components of bottom-up gaze allocation in natural images. *Vision Research 45,* 18, 2397–2416.

Pomplun, M. 2005. Saccadic selectivity in complex visual search displays. *Vision Research 46,* 12 (June), 1886–1900.

Privitera, C. and Stark, L. 1998. Evaluating image processing algorithms that predict regions of interest. *Pattern Recognition Letters 19,* 11, 1037–1043.

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

Rao, R., Zelinsky, G., Hayhoe, M., and Ballard, D. 2002. Eye movements in iconic visual search. *Vision Research 42,* 11 (May), 1447–1463.

Rayner, K. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin 124,* 3 (November), 372–422.

Reinagel, P. and Zador, A. 1999. Natural scene statistics at the centre of gaze. *Network-Computation in Neural Systems 10,* 4 (November), 341–350.

Rensink, R. A. 2000. The dynamic representation of scenes. *Visual Cognition 7*, 17–42.

Sodhi, M., Reimer, B., Cohen, J., Vastenburg, E., Kaars, R., and Kirschenbaum, S. 2002. On-road driver eye movement tracking using head-mounted devices. In *Proceedings of the symposium on Eye Tracking Research & Applications (ETRA)*. 61–68.

Stampe, D. M. 1993. Heuristic filtering and reliable calibration methods for video based pupil tracking systems. *Behavior Research Methods, Instruments, and Computers 25,* 2, 137–142.

Terzopoulos, D. and Rabie, T. F. 1997. Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research 1,* 1, 2–19.

Toet, A. 2006. Gaze directed displays as an enabling technology for attention aware systems. *Computers in Human Behavior 22,* 4 (July), 615–647.

Torralba, A. 2003. Modeling global scene factors in attention. *Journal of the Optical Society of America A-Optics Image Science and Vision 20,* 7 (July), 1407–1418.

Tosi, V., Mecacci, L., and Pasquali, E. 1997. Scanning eye movements made when viewing film: Preliminary observations. *International Journal of Neuroscience 92,* 1-2, 47–52.

Treisman, A. and Gelade, G. 1980. A feature-integration theory of attention. *Cognitive Psychology 12*, 97–136.

Voge, S. 1999. Looking at paintings: Patterns of eye movements in artistically naive and sophisticated subjects. *Leonardo 32,* 4, 325–325.

Wolfe, J. and Horowitz, T. 2004. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience 5,* 6 (June), 495–501.

Yarbus, A. 1967. Eye movements during perception of complex objects. In *Eye Movements and Vision*, L. Riggs, Ed. Plenum Press, New York, NY.

Zetzsche, C., Schill, K., Deubel, H., Krieger, G., Umkehrer, E., and Beinlich, S. 1998. Investigation of a sensorimotor system for saccadic scene analysis: an integrated approach. In *From animals to animats, Proceedings of the fifth international conference on the simulation of adaptive behavior*. Vol. 5. 120–126.

Received Month Year; revised Month Year; accepted Month Year