Region of Interest Determination Using Human Computation

Flávio Ribeiro [1], Dinei Florêncio [2]

[1] *Electronic Systems Engineering Department, Universidade de São Paulo, Brazil*
[1] fr@lps.usp.br

[2] *Microsoft Research, One Microsoft Way, Redmond, WA, 98052*
[2] dinei@microsoft.com

***Abstract*—The ability to identify and track visually interesting regions has many practical applications – for example, in image and video compression, visual marketing and foveal machine vision. Due to challenges in modeling the peculiarities of human physiological and psychological responses, automatic detection of fixation points is an open problem. Indeed, no objective methods are currently capable of fully modeling the human perception of regions of interest (ROIs). Thus, research often relies on user studies with eye tracking systems. In this paper we propose a cost-effective and convenient alternative, obtained by having internet workers annotate videos with ROI coordinates. The workers use an interactive video player with a simulated mouse-driven fovea, which models the fall-off in resolution of the human visual system. Since this approach is not supervised, we implement methods for identifying inaccurate or malicious results. Using this proposal, one can collect ROI data in an automated fashion, and at a much lower cost than laboratory studies.**

***Index Terms*—region of interest, foveation, point of gaze, fixation selection, human visual system, crowdsourcing, mechanical turk.**

I. Introduction

Even though the human visual system (HVS) is characterized by a large field of view, its resolution declines rapidly from the point of gaze [1]. To compensate for this, the HVS samples its environment by linking periods of fixation with fast and sudden movements called saccades. Since vision is suppressed during saccades [1], nearly all information comes from the fixation points, which we call regions of interest (ROIs).

ROI identification has many practical applications. Foveated image and video compression can provide major improvements in subjective quality by minimizing distortion on the neighborhood of the ROIs [2]–[4] at the expense of additional distortion at peripheral regions. ROIs can be used to measure advertising effectiveness by tracking attention duration across ad elements [5]. Active machine vision benefits from using context-aware cues and incorporating some pattern-matching capabilities of the HVS (e.g., [6]).

In applications where eye tracking hardware is not available, saliency models can be used to predict ROIs. Objective gaze selection methods can be broadly classified as top-down (using the cognitive interpretation of a scene) or bottom-up (exploring low-level features such as luminance, contrast, texture and motion). In practice, the HVS relies on a combination of both approaches. Nevertheless, since current state-of-the-art object and scene identification algorithms are still very domain-specific, most objective saliency models are bottom-up [7], [8]. To remove the requirement for scene identification, recent work models context-independent features with machine learning approaches, trained using annotated databases [9], [10]. However, training and validating these methods requires large datasets which are expensive and time consuming to obtain. Thus, objective fixation selection is still a very challenging open problem.

In this paper, we propose a method for facilitating the collection of subjective ROIs and high-level saliency maps. Instead of running laboratory user studies, we outsource ROI tracking to workers from an internet crowd. Tracking is performed with conventional pointing devices such as mice or trackpads, using a video player specifically designed for this purpose. The mouse pointer is replaced by fine crosshairs, around which the video player applies a real-time, radially dependent blur. The level of blur increases monotonically with the radius, simulating the HVS resolution map [11]. This motivates users to track interesting regions with the mouse, in order to minimize the perceived amount of blur.

The task of tracking ROIs with this video player is crowdsourced using Amazon Mechanical Turk. Workers are typically non-experts drawn from a pool of hundreds of thousands of individuals distributed around the world. Using this approach, one can obtain ROI data with a larger and more diverse pool than with laboratory studies, and with costs which are at least one order of magnitude smaller. To address the workers' lack of supervision and uncontrolled environment, we process all submitted scores to remove inaccurate results.

Even though crowdsourcing has become quite popular for user studies, its full potential as a generic DSP tool is only starting to be explored. Huang et al. [12] developed a web game for extracting image ROIs, where players earn points by agreeing on their choices. In contrast, we extract ROIs for full-motion video, and do not organize the task as a game. While games with a purpose [13] can potentially bring together large crowds seeking online entertainment, most such games do not become popular enough to be used as research tools. Thus, crowdsourcing marketplaces are still a more dependable method for recruiting workers in a scalable manner.

978-1-4577-1434-4/11/$26.00 ©2011 IEEE

Carlier et al. [14] have recently proposed a method for crowdsourcing the task of retargeting video for low-resolution devices. They use a video player with pan and zoom controls, which is operated by workers with no prior video editing experience. The results are then post-processed to reframe and stabilize each shot, ensuring spatial continuity and delivering the retargeted video. While this method can produce results whose quality approaches that of professionally edited video, it requires a level of cognition and attention to detail which is atypical for most crowd workers, who aim to deliver hundreds of micro-tasks per day. Thus, we use a simpler ROI collection method, suitable for producing short and simple tasks.

A related crowdsourced application is MoodSwings [15], which was designed for identifying and tracking emotion in music. While music is played, workers evaluate its emotional content by moving the mouse over a Cartesian plane. Horizontal displacements measure valence (happy vs. sad), and vertical displacements measure arousal (energetic vs. calm). While mood classification is by no means an easy task, ROI offers additional challenges. For example, there is typically more than one fixation point per frame, and ROIs move faster, making them more difficult to track.

The remainder of this paper is organized as follows. Section 2 describes our experiment design and post-processing methods. Section 3 shows experimental estimates for accuracy and latency, and also presents an example for ROI tracking with full-motion video. Section 4 has our conclusions and final comments.

II. Crowdsourcing ROI tracking

A. Experiment design

Amazon Mechanical Turk (MTurk) is a service designed for crowdsourcing large quantities of small tasks using a web interface and an open API. Jobs are known as human intelligence tasks (HITs), and are typically designed to be very simple and to require little specialized training. Most HITs can be completed in a few minutes, and workers are rewarded per HIT using a micropayment scheme. Submitted HITs can be rejected, in which case the worker does not get paid. Since MTurk accepts workers from all over the world, the typical pay is below minimum wage in the United States.

Our ROI tracking experiment relies on an interactive video player written for the Adobe Flash platform. It is designed to run as a browser plug-in, and is capable of streaming H.264 video. To make HITs as simple as possible, the application used for ROI tracking features no controls other than a large play button. Unlike a conventional video player, the user is shown a blurred version of the clip. The mouse pointer is replaced by thin crosshairs, and the level of blur increases monotonically with the distance to the cursor.

Let $b(\mathbf{x})$ be a 2-dimensional blur map with values between 0.0 and 1.0. This map defines the level of blur applied to each pixel of a frame. A blur map modeling the HVS should have values which increase with the distance to the fixation point. To this effect, we use the exponential blur map given by

$$b(\mathbf{x}) = 1 - e^{-\max\left(\|\mathbf{x}-\mathbf{x}_0\|_2 - r_0, 0\right)/r_0}, \qquad (1)$$

where $\mathbf{x} = (x_1, x_2)$ represents pixel coordinates, $\mathbf{x}_0$ is the cursor position and r_0 controls the radius of the region with no blur, as well as the rate at which the blur increases with radius.

Given a video frame $f(\mathbf{x})$, let $\tilde{f}(\mathbf{x})$ be produced by filtering $f(\mathbf{x})$ with a 10×10 box blur. The frame shown by the ROI video player is given by

$$g(\mathbf{x}) = [1 - b(\mathbf{x})]\, f(\mathbf{x}) + b(\mathbf{x})\, \tilde{f}(\mathbf{x}). \qquad (2)$$

The use of an exponential blur map was inspired by the exponential resolution maps used in [11], which fit psychological experiment data. However, [11] can implement any arbitrary level of blur by using a multiresolution pyramid of low-pass filtered frames. With our approach, the maximum level of blur is limited to the box blur used to produce $\tilde{f}(\mathbf{x})$. Furthermore, the use of a single level and a box blur (as opposed to a Gaussian blur) imply that $1-b(\mathbf{x})$ only approximates the resolution map proposed in [11]. Nevertheless, our implementation produces a convincing simulation of the HVS. Furthermore, these approximations are required to implement our proposal with the Flash Player, which was not designed for real-time video processing.

We implement this functionality as follows. The blur map $b(\mathbf{x})$ is initialized programmatically using Adobe Pixel Bender, which allows the description of 2D filter kernels using a C-like language. Decoded frames $f(\mathbf{x})$ are grabbed at 24 frames per second, copied to an auxiliary buffer and filtered using a built-in implementation of the box blur, producing $\tilde{f}(\mathbf{x})$. The foveated image $g(\mathbf{x})$ is then obtained by alpha blending $f(\mathbf{x})$ and $\tilde{f}(\mathbf{x})$, using $b(\mathbf{x})$ as the alpha mask. Using this approach, we can foveate 640x360 H.264 videos encoded at 500 kbps using approximately 60% CPU time on a 2.5 GHz Intel Core 2 Duo T9400 processor, using Flash Player 10.1 on Windows or Linux. In contrast, the conventional (non-foveated) video player requires approximately 25% of the CPU time. Fig. 1 shows an example.

While 60% CPU utilization is not trivial, our experiments have shown that less than 10% of MTurk users have experienced dropped frames due to slow processors. If more than 5% of the frames are dropped, the video player displays an error message and notifies the worker that his computer is too slow to participate in this experiment. To reduce the probability that the video will stop due to buffering issues, the video player preloads the clip until it can play it to the end without stopping, even if the bandwidth is reduced by 33%.

The MTurk HIT creation process is automated using a modified version of the open-source CrowdMOS tools [16]–[18], which were originally developed by the authors to crowdsource subjective quality user studies. Each HIT consists of an HTML page with brief instructions, followed by the interactive video player.

To test and develop this ROI tracking methodology, we used trailers from action and animation movies. They are

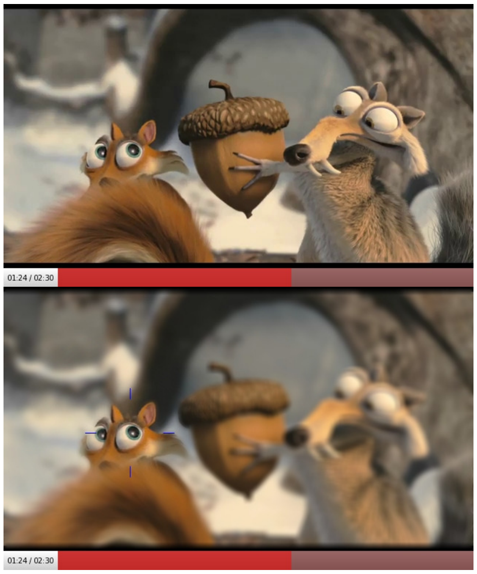

Figure 1. Top: original video; bottom: foveated video, as produced by the interactive video player. In the bottom frame, the mouse cursor is represented with blue crosshairs, and coincides with the modeled fixation point. A radially increasing blur is applied to the frame, simulating the resolution roll-off of the human visual system.

approximately 2 minutes long, and are typically rich in diversity, movement and transitions. Our studies offered a reward of \$.25/HIT. Considering the time required to work on each HIT and buffering times, each worker gets paid approximately \$5/hour. Due to the international worker pool, this amount is sufficient to run experiments featuring 20-40 workers in only a few hours.

B. Result screening

Since MTurk workers are unsupervised, results must be screened for accuracy. Workers have little incentive to submit intentionally inaccurate results, since rejected HITs are not rewarded, and each worker's acceptance rate is used as a qualification requirement by most MTurk requesters. Nevertheless, our experience with CrowdMOS showed that it is not unusual for studies featuring 20-40 workers to have 1 or 2 workers who submit obviously inaccurate results [17]. Even though workers can opt out of submitting their results at the end of the video, sometimes workers submit results which are only partially accurate (for example, because they were distracted during part of the experiment).

In general, video clips have multiple ROIs per frame. Thus, estimating mean cursor coordinates and discarding values with excessive distances to the mean produces unacceptable results. A clustering algorithm would be an obvious generalization, and one might consider using the distance to the closest cluster center as a quality measure. However, with the exception of very simple scenes, ROI maps are usually better represented as distributions (as there is no concept of cluster center). Thus, our proposed screening method does not implement clustering.

The video player is designed to sample the cursor coordinates at 24 Hz. At the end of the clip, the cursor coordinate history and its associated timestamps are submitted to the MTurk servers. The associated timestamps must be known because Flash Player timers contain a significant amount of jitter, which must be compensated for. Thus, the first processing step consists of using linear interpolation to obtain an approximation of what the mouse coordinates would be with jitterless sampling at 24 Hz.

This is followed by the screening procedure, which has two steps. The first step is designed to identify very inaccurate HITs, which are completely discarded. The second step compares each HIT with respect to the others, and decides whether results are accurate on a frame by frame basis.

Let N be the number of workers and T be the duration of the video. Let $(x_n(t), y_n(t))$ be the mouse coordinates (in pixels) for worker n at timestamp t, with $0 \leq n < N$ and $0 \leq t < T$. Define the tracking error for worker n as

$$e_n = \sum_t \left[\min_{i \neq n} \left\| (x_n(t), y_n(t)) - (x_i(t), y_i(t)) \right\|_2 \right],$$

where t iterates over all collected timestamps. The first screening step consists of discarding the HITs with the 20% largest tracking errors. From our experience, obviously bad results do not exceed 10% of the submitted HITs.

Even though this procedure is capable of eliminating submissions which are globally inaccurate, workers will always have ROI histories with brief local inaccuracies. We first scan over all cursor data for one worker at a time, and discard all contiguous sections in which the cursor moved less than 2 pixels on every frame for 3 seconds or more. This addresses common cases when workers stop moving the mouse.

Next we measure worker accuracy on a frame by frame basis, with respect to the pool of results. Consider the cursor position produced at timestamp t_0 by worker n, with $0 \leq t_0 < T$ and $0 \leq n < N$. Define

$$D_n(t_0) = \Big\{ \left\| (x_n(t_0), y_n(t_0)) - (x_i(t), y_i(t)) \right\|_2 \\ : |t_0 - t| \leq 0.2,\, 0 \leq i < N,\, i \neq n \Big\},$$

which contains the distances to the cursors from other workers, for a 400 ms window centered at t_0. As we show in the next section, this window is sufficient to model the different reaction times from other workers. If $\min D_n(t_0) > 50$, then we ignore the cursor data from worker n for a 1 s window centered at t_0. As we also show next, 50 pixels is well above the typical positioning error which can be obtained with this method. Ignoring results over a 1 s window addresses the fact that positioning inaccuracies require some time for the worker to detect and correct.

III. Experiments

A. Accuracy and latency measurements

To measure worker accuracy and latency, we developed two experiments using a variation of the video player described above. Instead of playing an H.264 stream, it draws a red circle with a 10 pixel radius, which the worker is asked to

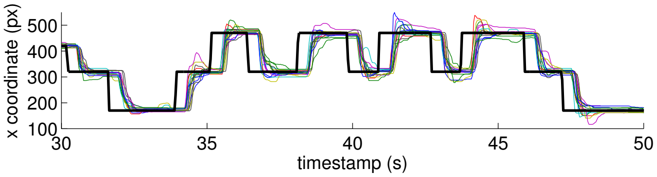

Figure 2. Worker latency detail (experiment J). —: ground truth.

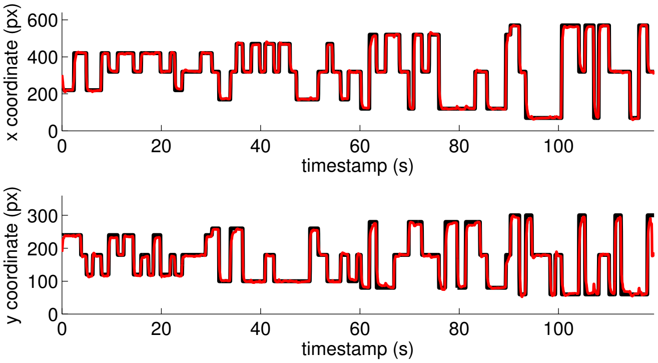

Figure 3. Worker latency results (experiment J). —: time-aligned ground truth, —: averaged answers.

track. The circle assumes random positions over a 3x3 grid, which expands to fill the player area.

We designed two variations of this experiment. In experiment J, the circle jumps between grid positions approximately every $1 + r$ seconds, where r is a pseudorandom value drawn uniformly between 0 and 1. In experiment T, it translates between grid positions at uniform speed. Experiment J was designed to estimate user latency, and experiment T was designed to measure accuracy. Each experiment lasted 120 seconds and was completed by 20 workers. Results were screened as prescribed in the previous section.

Fig. 2 shows a representative section of experiment J. Using the maximum cross-correlation between mouse tracking data and the ground truth position for the circle, we estimated the mean tracking latency to be approximately 500 ms. In contrast, primary saccades in humans have latencies between 100 and 150 ms [19]. Fig. 3 compares the delay-adjusted mean cursor position and the ground truth.

Fig. 4 shows results for experiment T, comparing the ground truth with individual answers. Since the red circle follows a continuous path, users can predict its movement and the mean tracking delay is only 70 ms. Mean horizontal and vertical tracking standard deviations are $\sigma_x \approx 21$ and $\sigma_y \approx 16$ pixels, respectively. Assuming the horizontal and vertical errors are independent such that variances are additive, we have a positioning standard deviation of $\sigma \approx 26$ pixels. Thus, the 50 pixel threshold used in the outlier detection corresponds to approximately twice this value. Fig. 5 compares the ground truth with the average over all workers.

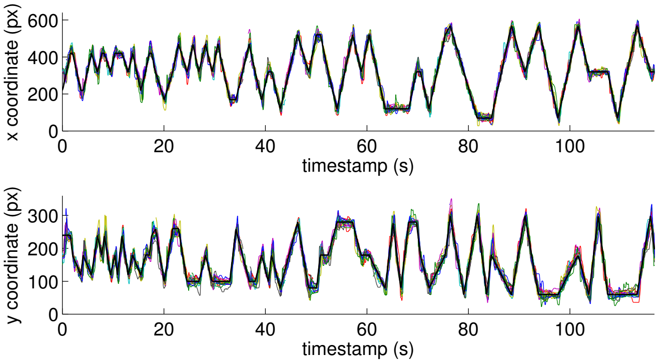

Figure 4. Tracking accuracy results (experiment T). —: ground truth.

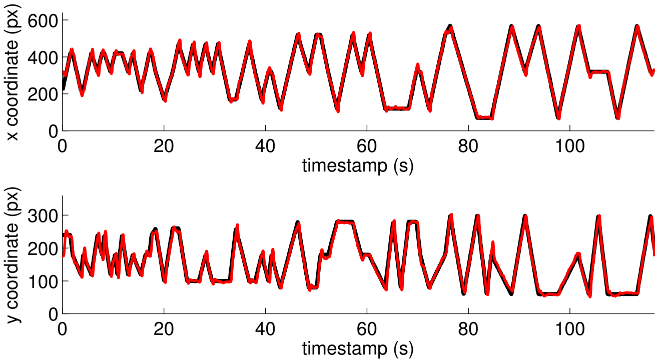

Figure 5. Tracking accuracy results (experiment T). —: ground truth, —: averaged answers.

B. ROI tracking for movie trailers

In this section we show results collected for the 2.5 minute trailer for the movie Ice Age 3. We compare results from 40 MTurk workers with those from a laboratory experiment with 12 volunteers using a Tobii x50 eye tracker. The MTurk data was post-processed as prescribed in Section 2. The eye tracking data was only processed to eliminate glitches due to off-screen glances.

Fig. 6 shows the cursor/eye history after screening. To facilitate the visualization, the ROI coordinate vector for each timestamp was convolved with a 1D Gaussian kernel with $\sigma = 5$ pixels. Detailed results for this experiment (overlaying ROIs on top of the video) can be seen at http://www.crowdmos.org/results/ROI/.

Like most modern animation trailers, this example combines a generous amount of movement and many sudden transitions between camera angles and scenes. Also, many scenes are characterized by more than one ROI. Fig. 7 shows one frame which would be extremely challenging to classify using an objective method. Due to the storyline, the attention is shifted to the female squirrel (shown on the left), despite the fact that she occupies a relatively small portion of the frame. Whenever the ROI was large with respect to the cursor, workers always produced enough positioning variety to cover entire ROIs.

The eye tracker has the definite advantage of following very fast movements, which would be difficult to reproduce using a

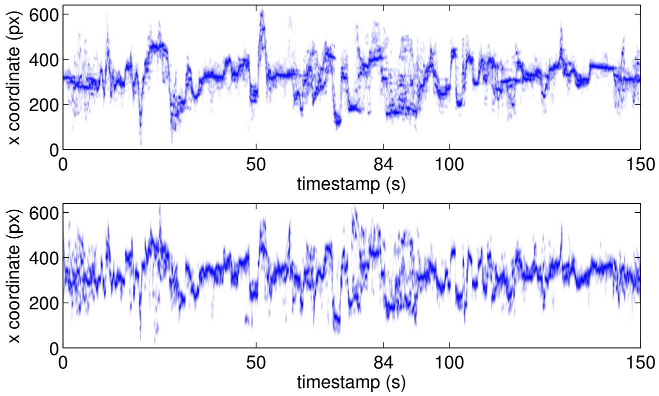

Figure 6. ROI tracking history for the Ice Age 3 trailer. Top: MTurk results; bottom: eye tracker results.

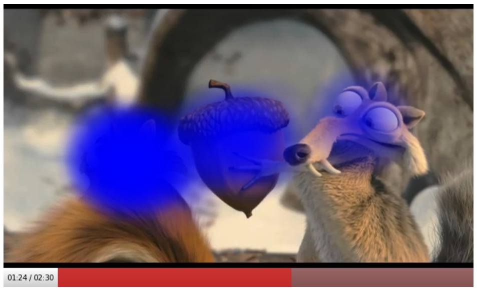

Figure 7. ROIs for the Ice Age 3 trailer, at the 84 second mark.

mouse. On the other hand, our methodology provides cleaner data, since the decision to move the mouse to relevant regions involves a higher degree of cognition. Thus, our proposal tends to filter out low-level instinctive reflexes and emphasize regions and events which the user considers to be important given a real time scene analysis.

IV. Conclusion

This paper describes a method of crowdsourcing subjective region of interest detection and tracking. It was designed to automate the determination of high-level saliency maps, where context and cognitive interpretation are fundamental to produce accurate and reproducible results.

At the time of this writing, objective ROI methods are dominated by bottom-up saliency models, which ignore cognitive aspects and thus cannot fully model human perception. By combining crowdsourcing with a screening algorithm, our proposal offers a method of automating subjective ROI studies. We can thus deliver top-down saliency data at costs which are at least one order of magnitude smaller than those for laboratory studies with eye tracking hardware. Indeed, for our experiments, the crowdsourcing cost was approximately $6 per hour of annotated video.

By providing a very practical means of acquiring subjective data, we hope to facilitate the development and evaluation of top-down models, opening new possibilities for research in foveated video compression, visual recognition and scene interpretation.

References

[1] B.A. Wandell, *Foundations of vision*, Sinauer Associates, 1995.

[2] W.S. Geisler and J.S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communication," in *Proc. SPIE*, 1998.

[3] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, 2002.

[4] S. Lee and A.C. Bovik, "Fast algorithms for foveated video processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 149–162, 2003.

[5] A.T. Duchowski, *Eye tracking methodology: Theory and practice*, Springer-Verlag, 2007.

[6] X.W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their colour and shape features extracted using human vision models," *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 675–685, 2006.

[7] U. Rajashekar, I. van der Linde, A.C. Bovik, and L.K. Cormack, "GAFFE: A gaze-attentive fixation finding engine," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 564–573, 2008.

[8] T. Avraham and M. Lindenbaum, "Esaliency (extended saliency): Meaningful attention using stochastic image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 693–708, 2010.

[9] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proc. ICCV*, 2009, pp. 2106–2113.

[10] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.Y. Shum, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, 2011.

[11] J.S. Perry and W.S. Geisler, "Gaze-contingent real-time simulation of arbitrary visual fields," in *Proc. SPIE*, 2002, vol. 4662, pp. 57–69.

[12] T.H. Huang, K.Y. Cheng, and Y.Y. Chuang, "A collaborative benchmark for region of interest detection algorithms," in *Proc. CVPR*, 2009.

[13] Luis von Ahn and Laura Dabbish, "Designing games with a purpose," *Commun. ACM*, vol. 51, pp. 58–67, August 2008.

[14] A. Carlier, V. Charvillat, W.T. Ooi, R. Grigoras, and G. Morin, "Crowdsourced automatic zoom and scroll for video retargeting," in *Proc. ACM MM*, 2010.

[15] Y.E. Kim, E. Schmidt, and L. Emelle, "Moodswings: A collaborative game for music mood label collection," in *Proc. ISMIR*, 2008.

[16] F. Ribeiro, D. Florencio, C. Zhang, and M. Seltzer, "crowdMOS Standalone Tools," available at http://research.microsoft.com/crowdmos/.

[17] F. Ribeiro, D. Florencio, C. Zhang, and M. Seltzer, "crowdMOS: An Approach for Crowdsourcing Mean Opinion Score Studies," in *Proc. of ICASSP*, 2011.

[18] F. Ribeiro, D. Florencio, and V. Nascimento, "crowdMOS: Crowdsourcing Subjective Image Quality Evaluation," in *Proc. of ICIP*, 2011.

[19] B. Fischer and E. Ramsperger, "Human express saccades: extremely short reaction times of goal directed eye movements," *Experimental Brain Research*, vol. 57, no. 1, pp. 191–195, 1984.