Interactive Images

December 2003
Microsoft Research Technical Report
MSR-TR-2003-64

Kentaro Toyama
Microsoft Research
One Microsoft Way
Redmond, WA U.S.A.
kentoy@microsoft.com

Bernhard Schoelkopf
Max Planck Institute
Spemannstrasse 38
72076 Tuebingen, Germany
bernhard.schoelkopf@tuebingen.mpg.de

ABSTRACT

Interactive Images are a natural extension of three recent developments: digital photography, interactive web pages, and browsable video. An interactive image is a multi-dimensional image, displayed two dimensions at a time (like a standard digital image), but with which a user can interact to browse through the other dimensions. One might consider a standard video sequence viewed with a video player as a simple interactive image with time as the third dimension.

Interactive images are a generalization of this idea, in which the third (and greater) dimensions may be focus, exposure, white balance, saturation, and other parameters. Interaction is handled via a variety of *modes* including those we call ordinal, pixel-indexed, cumulative, and comprehensive.

Through exploration of three novel forms of interactive images based on color, exposure, and focus, we will demonstrate the compelling nature of interactive images.

1. INTRODUCTION

Technological progress in digital photography appears to be measured by how well a digital photograph compares against its analog counterpart. Digital cameras are marketed as being more convenient and less expensive in the long term than analog cameras, but little else. The end goal is still the same – to shoot a still photograph.

Recently, some efforts have been made to do things with digital photography that are difficult or impossible with analog photography. Many digital cameras now come with a capacity to do a "sports shot" or to shoot short video clips. Some digital camera software comes equipped with image-stitching capabilities that allow one to create larger panoramas sewn together from smaller, overlapping images of the same scene.

In this paper, we consider a generalization of these trends that results in a novel form of media we call the *Interactive Image*. An interactive image goes beyond the standard media types of static imagery and sequential video. Instead of capturing a series of images in which time or pan/tilt parameters are varied (resulting, respectively, in standard video and 360 panoramas), we capture sequences in which other camera parameters, such as focus or exposure, are varied. Such a sequence gives us a correspondingly richer representation of the scene captured, and as a result, invites the possibility of richer interaction. Instead of browsing a video by manipulating "forward" and "backward" buttons, we can browse an interactive image by pointing to different objects in the image and watching them brighten with color or come into focus. Other forms of interaction are also possible and discussed in the following sections.

Lastly, we mention that a certain class of graphics-intensive web pages implement effects similar to those of the interactive images described here. For example, some sites implement "discoverable" links as a mouseover effect: when the cursor passes over a linked icon, the icon displays itself differently, thus popping out at the user. While these are undoubtedly images with which one can interact, we distinguish our work in two ways: First, the images we handle are photographs, not graphical icons or text. Second and more important, our work is concerned with the *automatic* construction of interactive images from image sequences. Automatic construction requires application of techniques from image processing and computer vision that are not necessary for handcrafted interactive web pages.

2. GENERAL APPROACH

The key concepts of an interactive image are simple and can be understood easily by construction:

1. Collect one or more digital images likely, but not necessarily restricted, to be of the same static scene, in which d^* camera parameters are varied. Let these images be labeled I_i^*, for $1 \leq i \leq n^*$.

2. Choose a *mode of interaction* – we list several possibilities below.

3. Use graphics and image processing techniques on the input images, I^*, to construct n *image representatives*. Label the representative images I_i, for $1 \leq i \leq n$.

4. Use image processing techniques to construct an *index image*, J, which specifies one image representative for each pixel.

5. During interaction, display the image representatives or processed combinations of image representatives based on user input, the index image, and the chosen mode of interaction.

Note that with sufficient processing power, it is possible to make a time-space trade-off in which the image representatives and index image are constructed online.

We now consider some possible modes of interaction. This list is not meant to be an exhaustive list, but it suggests the kinds of interaction that are possible. All of the examples in this paper will be implemented with each of the following modes of interaction.

1. **Ordinal:** Use sliders, joysticks, and so forth to directly control the indices of the representative to be displayed. A slider to browse a video sequence is an example: Moving the slider to the right increases the value of the displayed image's time index.

2. **Pixel-Index:** Using any means to select a pixel in the image, display the representative image which corresponds to that pixel in the index image. Implementable as a mouseover effect.

3. **Cumulative:** Allow a mechanism that freezes the image as displayed (by one of the above means, for example), and allow further interaction to have a cumulative effect.

4. **Comprehensive:** Construct an image that displays some combination of all of the image representatives in a single view.

Although the construction procedure is easy to understand in this general form, the interesting aspects of interactive images reside in the algorithms required to (1) generate image representatives, (2) generate index images, and (3) implement a mode of interaction. In the following sections, we discuss details for three types of interactive image.

The accompanying CD-ROM contains Java applets viewable with a web browser which implement the pixel-index mode of interaction for all three examples. We hope the reader will have a chance to try them out to feel the full effect of interactive images.

3. DECENT EXPOSURE

The first interactive image, we call *Decent Exposure*. These interactive images are constructed from multiple images of the same scene taken with different exposure. The dimensionality of the interaction will be $d = 1$, and we will begin with a handful of original images with varied exposure settings. We will then construct an array of image representatives and a single index image.

Figure 3: Decent Exposure index image.

Figures 1(a)-(c) show three images of an office scene taken at three exposure settings. We note that outdoor objects seen through the window are best viewed in one image while indoor objects are better seen in another. These images are the only images which compose the sequence $\{I^*\}$, and so we will generate a larger set of representative images.

(a)

(b)

Figure 4: Decent Exposure interactive images: (a) cumulative image that is the sum of Figures 2 (b) and (c); (b) comprehensive image which is a scaled version of S (see text).

The principle behind Decent Exposure's representative images is simple: We first construct a high-dynamic-range image from the originals and then pass them through a transfer function that emphasizes certain intervals of the total range. Construction of high dynamic-range images is a well-studied area [2, 7, 10], and we will take inspiration from previous work but use a novel algorithm that is better suited to our needs. In particular, our aim here is not to reconstruct accurate radiance maps [2], to specify a hardware rig to snap a high-range image [7], or necessarily to construct a single perceptually high-range image [10].

Work with dynamic-range images often begins by performing sums of the differently exposed originals, and we begin similarly. In fact, we take the most straightforward sum possible, where each new pixel $S(x, y)$ is simply the channel-wise sum of the RGB components of corresponding pixels $I_i^*(x, y)$, $1 \leq i \leq n^*$.

Representative images I_i are then constructed by passing S through sigmoid transfer functions with two parameters, μ and σ. The first parameter controls the center value of the range to be emphasized and the second controls the extent of expansion or contraction of values near the center value. We use the following sigmoid func-

(a) (b) (c) (d)

Figure 1: (a)-(c) Three images taken with different exposure settings; (d) examples of possible transfer functions (see text).

(a) (b) (c) (d)

Figure 2: 4 of 20 representative images constructed by passing the summed image through the transfer functions in Figure 1(d).

tion:

$$T_{\mu,\sigma}(v) = \frac{1}{1+\exp(-a_{\mu,\sigma}(v))}, \text{ where} \quad (1)$$

$$a_{\mu,\sigma}(v) = \frac{\sigma(v-\mu-k_{max})}{k_{max}}, \quad (2)$$

where k_{max} is the maximum value over all pixels/channels of S and v is the input pixel value. $T(\cdot)$ is additionally scaled such that its minimum value corresponds to 0 and its maximum value is 255.

To generate representative images, we fix σ ($\sigma = 4$ works well), and create equispaced values μ_i, such that $1 \leq i \leq n$, $\mu_1 = 0$, and $\mu_n = k_{max}$. To construct image i, we pass S through the transfer function by computing $T_{\mu_i,\sigma}(S(x,y))$ for every pixel. Some representative images constructed in this way are shown in Figure 2 – these images correspond to the output generated when the summed image is passed through the four transfer functions in Figure 1(d). Note that the representative images span a perceptual range even greater than that of the original images I^*, though, of course, no new information is generated.

The final set of images that Decent Exposure handles are these constructed representative images only. The original I^* are ignored, since they are likely to exhibit characteristics different from any of the constructed images (that is, they are unlikely to be generated from S and our sigmoid, no matter what values of μ and σ are chosen).

To compute the index image, we once again wish to maximize local contrast, but this time, contrast will be defined to be over a larger area than that required to compute second derivatives.

In particular, for each pixel, we compute J as follows,

$$J(x,y) = \arg\max_i C_i(x,y), \quad (3)$$

with $C(x,y)$ defined as the variance of the intensity values of pixels in an $N \times N$ window centered on (x,y) (clipped near image boundaries; we use $N = 15$ pixels). An example of the resulting index image is shown in Figure 3.

Finally, we implement the modes of interaction:

1. **Ordinal:** Implemented as a GUI slider that allows users to move back and forth between image indices. Any of the representative images can be viewed (Figure 2).

2. **Pixel-Index:** Implemented as a mouseover effect (see example contained in accompanying CD-ROM). When the cursor is at location (x,y) in the image, display the representative image, $I_{J_{final}(x,y)}$. Figures 2(b) and (c) show possible output ((a) and (d) are not referred to by the index image in practice). One difficulty with high-range images is that it is unclear how to display them on limited-range hardware [2]. This is one possible solution.

3. **Cumulative:** Implemented via mouse clicks. On Click 1, set the cumulative image, H, to the current displayed image, I_{i_1}. On subsequent Click m, do a pixel/channel-wise weighted sum: $H \leftarrow \frac{1}{m} I_{i_m} + \frac{m-1}{m} H$. One possible result is shown in Figure 4(a).

4. **Comprehensive:** One simple solution is to compress the summed image into the displayable range by scaling RGB values (Figure 4(b)). A more sophisticated option is to maximize contrast in each subregion and smoothly blend the results [10].

This example gives the flavor of the interactive image concept. We now continue with two other examples.

4. COLOR SATURA

Unlike Decent Exposure, *Color Satura* images are created from a single color image (see Figure 5(a)), and interaction allows us to explore the three-dimensional RGB color space. Derived from the original image, a representative image will look like a largely desaturated version of the original image, but with certain pixels colored in. Depending on the mode of interaction, the user can browse through the RGB space and see different parts of the image "light up" with color, while other parts fade back into gray. Here, $d^* = 1$, but representative images live in a space of dimensionality $d = 3$.

To create the representative images, we run the following operation on every pixel, $[R(x,y) \;\; G(x,y) \;\; B(x,y)]^T$, of the original

(a) (b)

Figure 5: Images for Color Satura example: (a) original (and comprehensive) image; (b) index image with $N = 2$.

(a) (b)

Figure 6: Color Satura in ordinal or pixel-index mode. Two different image representatives.

image, for each representative RGB-coordinate, $[\hat{R} \quad \hat{G} \quad \hat{B}]^T$, and a constant radius, $\hat{r}$:

$$r = \sqrt{(R-\hat{R})^2 + (G-\hat{G})^2 + (B-\hat{B})^2} \quad (4)$$

$$\alpha = \begin{cases} 0, & \text{if } r > \hat{r} \\ 2 - 2r/\hat{r}, & \text{if } \hat{r}/2 < r \leq \hat{r} \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

$$I(x,y) = \alpha I^*(x,y) + (1-\alpha)L(x,y), \quad (6)$$

where $L(x,y)$ is the luminance of pixel $I^*(x,y)$ represented as an RGB vector (*i.e.*, $R = G = B$). The representative RGB coordinates, $[\hat{R} \quad \hat{G} \quad \hat{B}]^T$, can be chosen in a variety of ways. We tried two: In the first, we choose $N \times N \times N$ values of $[\hat{R} \quad \hat{G} \quad \hat{B}]^T$, equally spaced between 0 and 255 (assuming 8-bit color channels). We find that $N = 4$ and $\hat{r} = 170$ creates pleasing representative images on a variety of images. In the second, we take a subset of the previous set in which exactly one or two of the $\hat{R}$, $\hat{G}$, and $\hat{B}$ values are equal to 255 (these are the $6(N-1)$ most color-saturated coordinates – this reduces the effective dimensionality of the interactive image to $d = 2$).

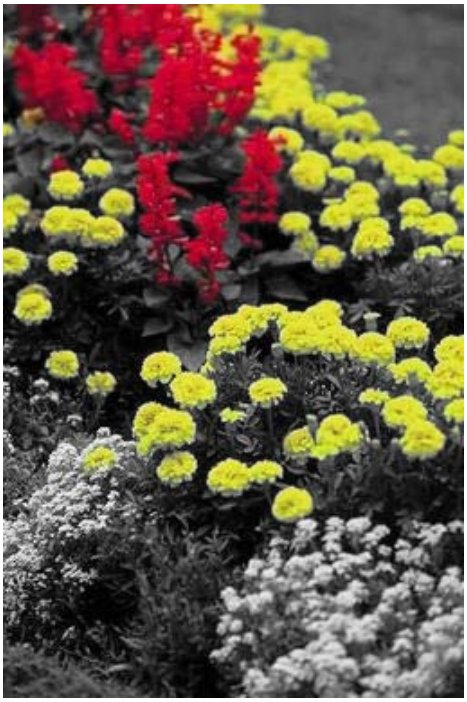


Figure 7: Color Satura in cumulative mode. The user made two clicks – one each on the red and yellow flowers.

To compute the index image, we let

$$J(x,y) = \arg\min_{[\hat{R}\hat{G}\hat{B}]} \|[R(x,y)\ G(x,y)\ B(x,y)]^T - [\hat{R}\ \hat{G}\ \hat{B}]^T\|_{L2}^2, \quad (7)$$

where the J are RGB vector values that index the representative images. Figure 5(b) shows an example of an index image color-coded to show the different indices.

With these images $\{I\}$ and J, it is trivial to construct various interactive images:

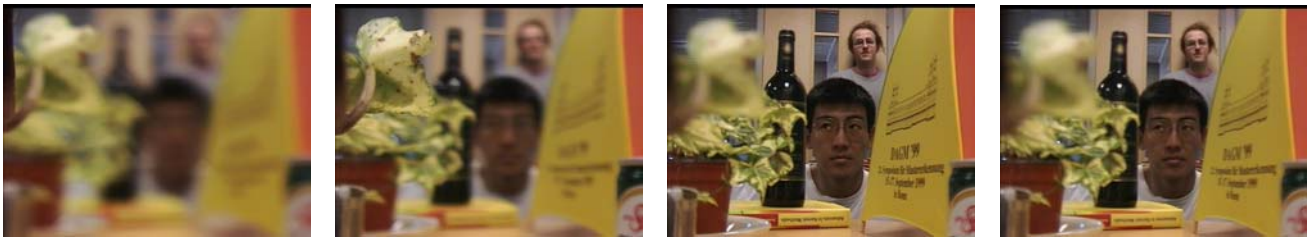
1. **Ordinal:** Implemented with keyboard keys. Using 3 pairs of keyboard keys to indicate moving up in R, G, and B, allow the user to browse through representative images. Output might appear as in Figure 6.

2. **Pixel-Index:** Implemented as a mouseover effect. When the cursor is at location (x,y) in the image, display the representative image, $I_{J(x,y)}$. With the cursor on a red flower, the image is displayed as in Figure 6(a); on yellow, Figure 6(b). We find that using the reduced RGB set creates more interesting interaction by ensuring that some pixels necessarily saturate with color, no matter where the cursor is.

3. **Cumulative:** Implemented with mouse clicks. On each click, replace I_g with the current displayed image. There is no need to recompute representatives. One possible result is shown in Figure 7.

4. **Comprehensive:** The original image I^* is already a comprehensive image (Figure 5(a)) displaying all of the colors simultaneously.

5. HOCUS FOCUS



(a) (b) (c) (d)

Figure 8: Original images I_i^*, for $i = 3, 10, 17, 24$, with varying focus settings.

Hocus Focus images are interactive images in which $d = 1$ and that single parameter is the camera focus setting. In Figure 8, we show a sample of four out of the $n^* = 27$ images that were taken of a particular static scene as the camera focus was varied from near to far. Notice that due to differential blurring based on depth of the object, different objects come into focus in different images.

For this example, we have a sufficient number of images to begin with, so we will use the original images themselves as representative images. Instead, we will concentrate on the computation of the index image.

The index image J will map each pixel $J(x, y)$ to an index i that indexes the image which is most in focus for that pixel. Computation of the index image can be viewed as a variation on "depth from focus" work in computer vision [1, 6, 8, 15, 17]. In particular, where depth-from-focus research is interested in actually determining the relative distance of image objects from the camera, we are only interested in the index of the corresponding image at that depth. Below, we describe a novel algorithm built on depth-from-focus work, which has been adapted to suit the needs of an interactive image.

The standard model of blurring supposes that pixels have been convolved with a pillbox function – a constant-valued disc centered at the origin and zero elsewhere [5]. Effectively, what this means is that blurred pixels are generated by a weighted average of the nearby pixels that might be collected by an ideal pinhole – the more blurring, the more pixels are averaged. Averaging decreases the local contrast in an image, and so it follows that the $J(x, y)$ should be computed to maximize contrast as in Equation 3, but where $C_i(x, y)$ is specified for an even smaller neighborhood. We compute contrast as the sum of the squares of the second spatial derivative (similar to the modified Laplacian used in previous work [8]):

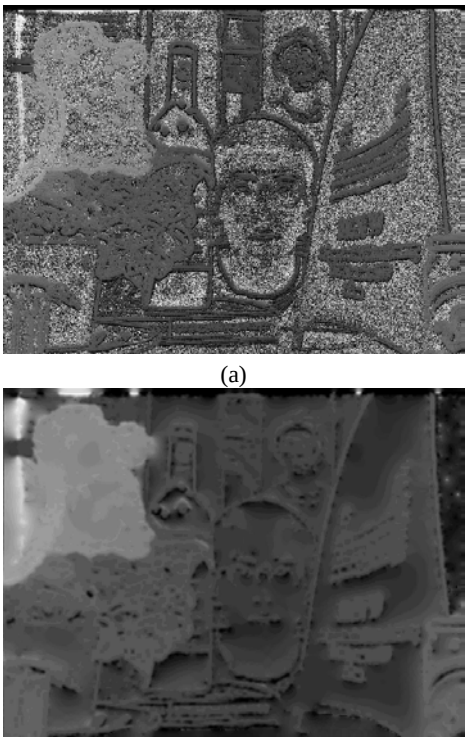
$$C(x, y) = \left(\frac{\partial^2 l}{\partial x^2}\right)^2 + \left(\frac{\partial^2 l}{\partial y^2}\right)^2, \tag{8}$$

where $l(x, y)$ is the (1-dimensional) luminance of pixel $I(x, y)$, and the partial derivatives are computed by repeated application of finite differences.

Empirically, we observed two problems which made this naive computation less than ideal (refer to Figure 9(a)): First, camera noise turns out to be a strong source of apparent contrast by this metric; and second, regions which lack texture do not exhibit strong contrast at any focus setting.

To overcome the first problem, we pre- and post-process all images by convolving them with a Gaussian filter with $\sigma = 2$ pixels. Since the contrast function (Equation 8) is not linear, pre-processing and post-processing have different effects – pre-processing smooths the original images and post-processing smooths the resulting index image.

To mitigate the second problem, we run an anisotropic diffusion process [11] on the index image of Equation 3, where iterations are performed to satisfy the following:



(a)

(b)

Figure 9: Hocus Focus: (a) index image J – darker values correspond to object being in focus far from camera; (b) index image J_{final} after diffusion.

$$\frac{\partial J}{\partial t} = k_d \nabla^2 J. \tag{9}$$

To work toward the steady state, we iterate as follows:

$$J_t = J_{t-1} - \alpha \operatorname{div}\left[\rho'(\nabla \|J\|)\nabla J\right], \tag{10}$$

and ρ is a monotonic function such as $1 - \exp(-kx^2)$. Since we know when we can be confident of our initial index values (see Figure 10), we run the iterations with a clamp on pixels $J(x, y)$:

$$J_t = J_{t-1} \quad \text{if} \quad \max_i C_i(x, y) > k_{mc}, \tag{11}$$

where k_{mc} is set to some fraction of the maximum contrast over all images. The value of k_{mc} is dependent on camera shot noise; we used 0.06 times maximum contrast. Intuitively, the diffusion

Figure 10: Relative maximum contrast values over all images I. Lighter pixels have high maximum contrast and are likely to be reliable indicators of actual depth.

allows good index values to flow into untextured regions, whose index values are assumed to be near those of their bounding edges (which necessarily provide good measures of contrast).

The final index image J_{final} after 100 iterations of diffusion is shown in Figure 9(b).

Again, there are various ways of constructing interactive images:

1. **Ordinal:** Implemented as a GUI slider which allows users to move back and forth between image indices. At any given moment, one of the original images is shown to the user (for example, Figure 8).

2. **Pixel-Index:** Implemented as a mouseover effect (see example contained in accompanying CD-ROM). When the cursor is at location (x, y) in the image, display the representative image, $I_{J_{final}(x,y)}$. Again, one of the original images is shown, with the effect that the object under the cursor is sharply in focus.

3. **Cumulative:** Implemented via mouse clicks. With each click on coordinate $(\hat{x}, \hat{y})$, the set of pixels given by

$$\{(x, y) : J_{final}(x, y) = J_{final}(\hat{x}, \hat{y})\} \qquad (12)$$

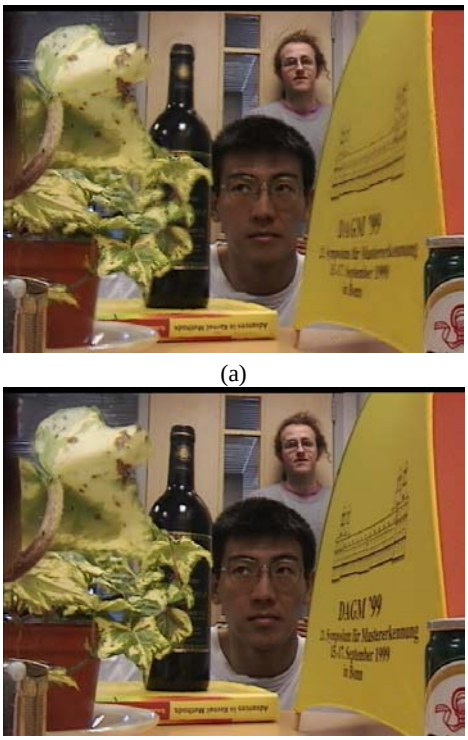
   are set to their values from image $I_{J_{final}(\hat{x},\hat{y})}$, ideally bringing all objects in that depth plane into focus. Figure 11(a) shows an example which brings near and far elements into focus, while keeping middle-ground objects out of focus (an impossibility with analog photos).

4. **Comprehensive:** Collected an image H, where $H(x, y) = I_{J_{final}(x,y)}(x, y)$, for all (x, y), to create a globally in-focus image. See Figure 11(b). (Similar results using different techniques have been achieved elsewhere [4].)

Hocus Focus illustrates another advantage of interactive images. Once the images are collected, the user can play with depths of field and so on *post hoc*. Photographers would only need to capture focus-varied sequences once, instead of having to try several shots with varying parameters to get the right effect, only to discover after developing film whether the right shot was captured.

6. CONCLUSIONS

Users find interactive images very compelling. They provide an additional depth to normal images that are not available with traditional forms of analog photography. Given the relative ease with which they can be created, we believe they will make an important



(a)

(b)

Figure 11: Hocus Focus examples: (a) cumulative image, where only middle-range objects are out of focus; (b) comprehensive, globally in-focus image.

addition to today's digital media, which at present, consist largely of still images, video, and handcrafted GUI effects.

Psychophysical research shows that the human visual system is naturally and immediately attracted to regions of an image which exhibit high frequency (*i.e.*, locally high contrast) [9] or saturated color [3]. By giving the user control to determine what elements of an image "come into focus", interactive images create positive feedback in which the user's object of attention is emphasized, thus reinforcing interest.

There is a vast literature on preattentive visual phenomena [3, 12, 13, 14, 16]. This research shows that certain types of image features "pop-out" immediately for observers without requiring a serial search over the image. Color and high frequency are two of the better-studied pop-out features, but others exist. One can imagine interactive images which allow browsing through scenes by edge orientation, object depth, size, second-order statistics, and so on, and each of these suggests a line of possible future work.

7. REFERENCES

[1] T. Darrell and K. Wohn. Pyramid based depth from focus. In *Proc. Computer Vision and Patt. Recog.*, pages 504–509, 1988.

[2] P.E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH Conf. Proc.*, pages 369–378, 1997.

[3] M. Green. Visual search: detection, identification and localization. *Perception*, 21:765–777, 1992.

[4] P. Haeberli. A multifocus method for controlling depth of field. *http://www.sgi.com/grafica/depth/index.html*, October 1994.

[5] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[6] E.P. Krotkov. *Active Computer Vision by Cooperative Focus and Stereo*. Springer, New York, 1989.

[7] T. Mitsunaga and S. K. Nayar. High dynamic range imaging: Spatially varying pixel exposures. In *Proc. Computer Vision and Patt. Recog.*, 2000.

[8] S. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Patt. Anal. and Mach. Intel.*, 16:824–831, 1994.

[9] P. Reinagel and A. M. Zador. Natural scene statistics at the centre of gaze. *Network: Comput. Neural Syst.*, 10(4):341–350, November 1999.

[10] R. Szeliski. Autobracket. Technical report, Microsoft Research, 2000.

[11] B.M. ter Haar Romeny. *Geometry-Driven Diffusion in Computer Vision*. Kluwer, 1994.

[12] A. Treisman. Preattentive processing in vision. *CVGIP: Image Understanding*, 31:156–177, 1985.

[13] A. Treisman and Stephen Gormican. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, 95(1):15–48, 1988.

[14] Q. Wang, P. Cavanagh, and M. Green. Familiarity and pop-out in visual search. *Perception and Psychophysics*, 56:495–500, 1994.

[15] M. Watanabe and S.K. Nayar. Rational filters for passive depth from defocus. *Int'l J. of Computer Vision*, 27:203–225, 1998.

[16] J. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1995.

[17] Y. Xiong and S.A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. *Int'l J. of Computer Vision*, 22:25–59, 1997.