

# DISTANCE BLURRING FOR SPACE-VARIANT IMAGE CODING

T. Popkin, A. Cavallaro

Multimedia and Vision Group,  
Queen Mary, University of London, E1 4NS, UK

D. Hands

British Telecommunications PLC,  
Martlesham Heath, IP5 3RE, UK

## ABSTRACT

We present a novel selective blurring algorithm that mimics the optical distance blur effects that occur naturally in cameras and eyes. The proposed algorithm provides a realistic simulation of distance blurring, with the desirable properties of aiming to mimic occlusion effects as occur in natural blurring, and of being able to handle any number of blurring and occlusion levels with the same order of computational complexity. We have performed subjective experiments to compare the perceived quality of distance blurred images with that of foveation-filtered images under equivalent conditions, when both are used as space-variant prefiltering stages prior to a JPEG encoder. The results show that the distance-based blurring was significantly preferable to the foveation blurring for four out of nine test images, whereas a significant converse preference was found for only one test image.

**Index Terms**— Depth of field, foveated image coding, foveation.

## 1. INTRODUCTION

Lossy image and video coding systems aim to allow, for a given bitrate constraint, distortion that is maximally acceptable to the observer. The usual approach in aiming for maximally acceptable distortion is to aim for minimally visible distortion. One approach, known as *foveated* coding, does this by applying a spatially-varying fidelity or bitrate which aims to match the spatial fall-off in retinal cell density (hence resolution) of the human eye [1], thereby saving bitrate by exploiting the fact that, at any instant in time, the eye will only see a narrow region in sharp focus [2]. In *gaze-contingent* coding, in which the display adapts in real time to viewer eye movements, clear benefits have been demonstrated, such as an 18.8-to-1 reduction in bandwidth with minimal perceived distortion [3]. However, in the scenario where the observer's eyes are free to look away from the assumed foveation point, as in systems which rely on the prediction of eye movements (e.g., Itti [4]) the benefits are not so clear.

An alternative to aiming for minimally noticeable distortion is to aim for distortion of a style that is visually more

acceptable if noticed, thereby reducing or removing the need for human fixation to be correctly predicted. One such an alternative, borrowed from photography, is the idea of reducing depth of field, which produces visible distortion but is considered to enhance overall picture quality. A convenient side effect of this is that the resulting reduction of image bandwidth will reduce the average bitrate if used as a preprocessing stage in front of most image or video encoders. A related approach is the extension of foveation into the third dimension, by also matching the depthwise fall-off in resolution due to eye focal effects [5] as well as matching sideways fall-off in retinal resolution as in ordinary foveation. However, to our knowledge, distance blurring on its own, with the aim of disguising image or video bandwidth reduction as a natural camera effect, has not been employed for bitrate reduction purposes.

We propose a novel selective blurring algorithm that aims to mimic the optical distance blur effects that occur naturally in a camera or eye. The proposed algorithm differs from existing approaches [5, 6, 7] by virtue of emulating occlusive effects (as occur in natural blurring at object edges) while being able to handle any number of blurring levels and occlusion levels with the same order of computational complexity. We present it in the context of image bitrate reduction, which we support by a subjective comparison with foveation filtering.

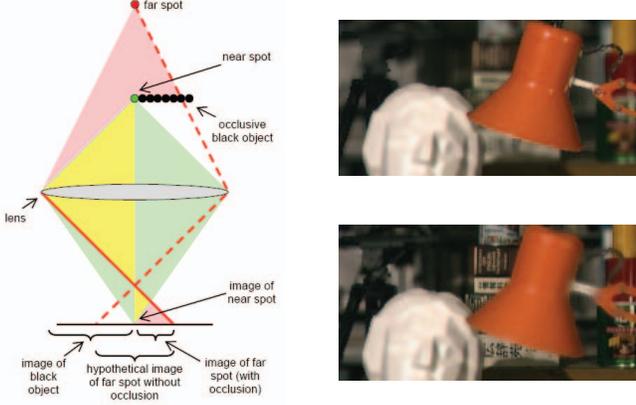
Sec. 2 of this paper describes the algorithm. Sec. 3 presents the results of the subjective comparison. Sec. 4 concludes the paper.

## 2. PROPOSED APPROACH

In order to mimic the style of blurring that a human viewer would be accustomed to seeing in photographs, it is necessary to cater for the effects that occur around the boundaries of objects which occlude further-away objects in the scene. These occlusive blurring effects occur when, for example, a sharply-focussed near object is in front of a blurred distant object, so that the blur of the distant object stops sharply at the edge of the near object, with no part of the blur overlapping any part of the near object (see Fig. 1). Conversely, if a blurred near object is in front of some sharply-focussed distant object, the edges of the blur of the near object will spread over the distant object. However, in contrast, an ordinary, unocclusive selective blurring technique would cause every blurred pixel

---

This work was jointly supported by the Engineering and Physical Sciences Research Council (UK) and British Telecommunications PLC.



**Fig. 1.** Left: example of occlusive effect of distance blurring. The light from the near spot is focussed sharply on one point, but the light from the far spot (out of focus) is spread over a region which cuts off sharply (due to the occlusive black object) at the image of near spot. Right: examples of occlusive selective blurring (top: lamp in focus; bottom: cans in focus). Image: *Head & Lamp* from Tsukuba University.

to be spread over its neighbours regardless of whether they are considered nearer to or further from the camera.

We propose an occlusive selective blurring algorithm that works whereby the intensity from each pixel is spread uniformly over a square area of variable size, subject to sharp occlusions by any nearer pixel. Given an image we apply a spatial map of blur levels combined with a spatial map of occlusion levels. Ordinary, unocclusive square blurring could be done with computational simplicity using an approach involving cumulative images (integral pictures), but to incorporate occlusion effects, we use a modified approach in which each pixel sum is handled using a special look-up tree that performs any partial sum that discards all pixels below a given occlusion level. The details of the algorithm are given below.

Consider a greyscale image or colour plane  $g: \mathbb{Z}^2 \rightarrow \mathbb{R}$ , a blur map  $b: \mathbb{Z}^2 \rightarrow \{0, 1, 2, \dots\}$  and an occlusion map  $\omega: \mathbb{Z}^2 \rightarrow \mathbb{Z}$ , whereby each  $\omega(x, y)$  gives the occlusion level at image location  $(x, y)$ , and whereby locations with higher occlusion levels will occlude locations with lower occlusion levels. We define these over  $\mathbb{Z}^2$  for simplicity; in practice we only work over a narrow subset of this. Let  $\mathbb{I}$  represent a binary-valued indicator function which states whether a proposition is true (1) or false (0), and define  $\beta_b(x, y) = 2b(x, y) + 1$  for all  $x, y$ .

We define an *occlusively selectively pixel-spread* array,  $S_{g,b,\omega}: \mathbb{Z}^2 \rightarrow \mathbb{R}$ , in terms of  $g, b$  and  $\omega$  as follows:

$$S_{g,b,\omega}(x, y) = \sum_{(m,n) \in \mathbb{Z}^2} \left( \frac{g(m,n)}{\beta_b(m,n)^2} \times \mathbb{I}(\omega(x, y) \leq \omega(m, n)) \times \mathbb{I}(\max\{|x-m|, |y-n|\} \leq b(m, n)) \right) \quad (1)$$

for all  $x$ . In other words, the greyscale intensity  $g(m, n)$ , from location  $(m, n)$  will only spread over another location  $(x, y)$

if  $\max\{|x-m|, |y-n|\} \leq b(m, n)$  and  $\omega(x, y) \leq \omega(m, n)$ . Consider the three-argument function  $S'_{g,b,\omega}: \mathbb{Z}^3 \rightarrow \mathbb{R}$ , defined as follows:

$$S'_{g,b,\omega}(x, y, \alpha) = \sum_{(m,n) \in \mathbb{Z}^2} \left( \frac{g(m,n)}{\beta_b(m,n)^2} \times \mathbb{I}(\max\{|x-m|, |y-n|\} \leq b(m, n)) \times \mathbb{I}(\alpha \leq \omega(m, n)) \right) \quad (2)$$

for all  $(x, y)$  and  $\alpha$ . Note that  $S_{g,b,\omega}(x, y) = S'_{g,b,\omega}(x, y, \omega(x, y))$  for all  $(x, y)$ . Now consider a function  $\underline{S}''_{g,b,\omega}: \mathbb{Z}^2 \rightarrow \mathbb{R}^\infty$ , defined such that for each  $x$  and  $\alpha$  (both integers),  $\underline{S}''_{g,b,\omega}(x, y)$  is an  $\infty$ -dimensional vector, the  $\alpha^{\text{th}}$  component of which is  $S'_{g,b,\omega}(x, y, \alpha)$  and the  $\omega(x, y)^{\text{th}}$  component of which is  $S_{g,b,\omega}(x, y)$ . Consider also a function  $g'': \mathbb{Z}^2 \rightarrow \mathbb{R}^\infty$ , defined such that for each  $m, n$  and  $\omega$  (all integers),  $g''_\omega(m, n)$  is an  $\infty$ -dimensional vector, the  $\alpha^{\text{th}}$  component of which is  $g(m, n) \mathbb{I}(\alpha \leq \omega(m, n))$ . It can be shown that

$$\underline{S}''_{g,b,\omega}(x, y) = \sum_{(m,n) \in \mathbb{Z}^2} \left( \frac{g''_\omega(m,n)}{\beta_b(m,n)^2} \times \mathbb{I}(\max\{|x-m|, |y-n|\} \leq b(m, n)) \right). \quad (3)$$

From this, the following can be derived:

$$\underline{S}''_{g,b,\omega}(x, y) = \sum_{m=-\infty}^x \sum_{n=-\infty}^y \underline{D}''_{g,b,\omega}(m, n) \quad (4)$$

for all  $(x, y)$ , where  $\underline{D}''_{g,b,\omega}$  is defined as follows:

$$\underline{D}''_{g,b,\omega}(m, n) = \sum_{(p,q) \in \mathbb{Z}^2} \left( \frac{g''_\omega(p,q)}{\beta_b(p,q)^2} \cdot (\delta_{q-b(p,q)}^{p-b(p,q)} + \delta_{q+b(p,q)+1}^{p+b(p,q)+1} - \delta_{q-b(p,q)}^{p+b(p,q)+1} - \delta_{q+b(p,q)+1}^{p-b(p,q)}) \right) (m, n) \quad (5)$$

for all  $(m, n)$ , where  $\delta_n^m(x, y) = \mathbb{I}(x=m \text{ and } y=n)$  for all  $m, n, x$  and  $y$ .

Equations (4) and (5) define the pixel spreading approach in terms of infinite dimensional vectors but do not show how we deal with them computationally. Our computation involves two main stages: firstly we construct what we refer to as *corner lists* and secondly we build from these a number of chains of interconnected binary tree structures, referred to herein as *occlusive sum look-up trees*.

Each  $\pm g''_\omega(p, q) / \beta_b(p, q)^2$  vector is represented by just two values,  $\omega(p, q)$  and  $\pm g(p, q) / \beta_b(p, q)^2$ . The  $\underline{D}''_{g,b,\omega}(x, y)$  values are each represented and manipulated as an unordered list of all  $(\omega(p, q), \pm g(p, q) / \beta_b(p, q)^2)$  pairs at the four locations which satisfy  $(x, y) = (p + \frac{1}{2} \pm (b(p, q) + \frac{1}{2}), q + \frac{1}{2} \pm (b(p, q) + \frac{1}{2}))$  (that is, each at a corner of the square spread of the pixel). At the start, each  $\underline{D}''_{g,b,\omega}(m, n)$  is represented by an empty list; subsequently, additions and subtractions are performed simply by appending the  $(\omega(p, q), \pm g(p, q) / \beta_b(p, q)^2)$  pairs to the end of the corner lists.

After the corner lists have been assembled, they are replaced with tree structures that contain the same information

but which allow partial sums of pixel values over certain subdivisions of the image while excluding anything below a certain occlusion level. The image is partitioned into a hierarchy of groups of 1, 2, 4, 8, etc. adjacent rows, for each of which a partial sum, to a given occlusion level and to a given horizontal location, can be performed efficiently using our occlusive sum look-up trees. For each level of the hierarchy, and each row group, a one-dimensional array of look-up trees is constructed, each of which can be used to efficiently look up the sum of all values in that row group to the left of the given column. The look-up tree for each location in these arrays can be considered to hold a partial  $S''_{g,b,\omega}(x, y)$  vector; that is, an array of partial sums, one for each occlusion level. To calculate the  $S_{g,b,\omega}(x, y)$  value for a given row, column and occlusion level, the appropriate partial sums from the appropriate row groups are separately extracted then added together. This costs  $O(\log N)$  for each extraction, with  $O(\log N)$  extractions, so costing  $O(\log(N)^2)$  for each pixel location.

As a final stage, our algorithm applies an adjustment factor to each location in result of this pixel-spreading, by dividing each pixel colour component by the luminance of a pure-white image after subsection to the same occlusive pixel-spreading. This compensates for the fact that this approach would otherwise unnaturally darken the image near object boundaries.

The images on the right-hand side of Fig. 1 were generated by the proposed algorithm, with different parts of the scene chosen to be in focus.

The next section reports on our subjective experiment in which we employed our algorithm.

### 3. EVALUATION AND RESULTS

We have performed a subjective experiment to test the hypothesis that distance blurred coding is preferable to foveated coding.

Forty-three non-expert subjects compared the results of distance-based blurring and foveation filtering as preprocessing stages for a JPEG encoder. The test method was a non-standard, single-stimulus modification of Variant I of the *Double-Stimulus Continuous Quality Scale* (DSCQS) [8] test, named herein *SSCQS*. Three images were used (see Fig. 2), from the Middlebury Stereovision test set [9] along with their associated disparity maps (from which blur maps and occlusion level maps were derived). We use a cut-off frequency interpretation [2] of the Geisler and Perry eye model, whereby  $f_c(e) = e_2 \ln(1/CT_0)/((e + e_2)\alpha)$  for all retinal eccentricities  $e$  (deg), where  $f_c(e)$  is the spatial cutoff frequency (cycles/deg) and  $(e_2, \alpha, CT_0) = (2.3, 0.106, 1/64)$ . Eccentricities were approximated according to the relationship  $e = (360 \sqrt{(x-x')^2 + (y-y')^2})/(2\pi HR)$ , for each image location  $(x, y)$ , where  $(x', y')$  is the fixation point, with image height  $H$  pixels, and distance/height ratio  $R$  (given viewing distance 40cm and screen resolution 0.264mm per pixel).



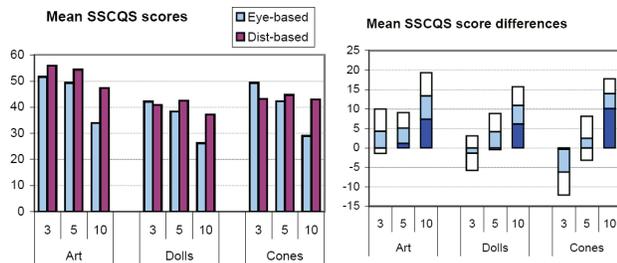
**Fig. 2.** Top: raw images (*Art*, *Dolls* and *Cones*), with our chosen points of interest shown by a white dot. 2<sup>nd</sup> row: associated disparity maps. 3<sup>rd</sup> row: foveation filtered, JPEG-encoded test images for mean blur 10. Bottom: images using distance blurring.

The conversion from  $f_c(e)$  values into a blur map was done by calculating the corresponding  $1/f_c(e)$  values and scaling them proportionally such that the overall mean blur level was a predefined value. In order to provide a fair comparison, both blurring schemes were implemented using the same occlusive selective blurring code, with the only differences being the blur maps and occlusion maps used. The foveation filtering was done using a uniform occlusion map (i.e., no occlusion). The blur maps for the distance-blurring were created such that the blur-level histogram was exactly the same as for the foveation filtering. Mean blurring levels of 3, 5 and 10 were investigated, at 0.4 bpp.

Fig. 2 shows the raw images (with point of interest highlighted) and disparity maps, along with example test images for each of the two types of blurring. Fig. 4 shows extracts from these images.

The results of the evaluation are summarized in Fig. 3, which shows the mean scores and score differences when comparing equivalent pairs of foveated and distance-blurred images. Overall, five out of nine of the images gave statistically significant results, of which four out of five results indicated an average preference for the distance-based blurring over the foveation filtering.

The results show a clear preference for the higher blurring level of ten pixel-widths, whereas the lower-blur-level results



**Fig. 3.** Left: mean test scores between foveation filtered (*eye-based*) and distance blurred (*dist-based*) images. Right: mean differences. 95% confidence intervals are indicated by a white bar area on the away-from-zero side of each mean and a pale-coloured bar area on the towards-zero side of each mean. Significant differences have an area of darkly-coloured bar.

were more prone to random error, as is to be expected considering that the blurring at these levels was very slight, so ordinary JPEG artifacts dominate the visible distortion. One result (minimum-blur *Cones*) gave a significant negative result, favouring foveation-filtered JPEG over distance-blurred JPEG. At this low blur level, JPEG compression artifacts had visual dominance over the blurring effects, so the question which type of blurring was better was obscured by random artifact differences. Both versions of this image are shown in Fig. 5, with zoomed-in extracts showing that the favoured image exhibited marginally smoother features near the point of interest.

#### 4. CONCLUSION

We have made a case for applying selective preblurring of a style which a human viewer may mistake for distance blur effects as naturally occur in cameras or eyes, and for this we have proposed a selective blurring algorithm which applies occlusive blurring effects around object boundaries. In subjective quality tests on preblurred JPEG, we found the distance-based blurring to be preferable to the equivalent foveation filtering in four out of nine of the tests, and the converse in only one out of nine.

Future work includes extending the proposed distance blurring concept to video coding to establish whether the same perceived quality advantages can be obtained.

#### 5. REFERENCES

- [1] W. S. Geisler and D. B. Hamilton, "Sampling-theory analysis of spatial vision," *J. Optical Society of America A*, vol. 3, pp. 62–, 1986.
- [2] Z. Wang, L. Lu and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 243–254, Feb. 2003.
- [3] P. Kortum and W. Geisler, "Implementation of a foveated image



**Fig. 4.** Zoomed-in portions of test images (blur level 10). Top row: created using foveation filtering. Bottom row: created using distance-blurring.



**Fig. 5.** Zoomed-in portions of low-blur *Cones* test image (blur level 3). Left: raw image; middle: created using foveation filtering; right: using distance-blurring.

coding system for image bandwidth reduction," *Proc. SPIE*, Vol. 2657, pp. 350–360, Apr. 1996.

- [4] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 10, Oct. 2004.
- [5] I. van der Linde, "Multi-resolution image compression using image foveation and simulated depth of field for stereoscopic displays," in *Proc. SPIE*, 2004, vol. 5291.
- [6] J. Krivanek, J. Zara and K. Bouatouch, "Fast depth of field rendering with surface splatting," in *Proc. Computer Graphics International*, Los Alamitos, 2003, pp. 196–201.
- [7] L. Blonde, T. Viellard and D. Sahuc, "Method of generating blur," 2005, European patent EP1494174.
- [8] ITU-R Rec. BT.500-11, "Methodology for the subjective assessment of the quality of television pictures," 2002.
- [9] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International J. Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.