Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 23 (2005) 611–618
www.elsevier.com/locate/imavis

Fusion of multi-exposure images

A. Ardeshir Goshtasby*

Department of Computer Science and Engineering, Wright State University, 303 Russ Engineering Center, Dayton, OH 45435, USA

Received 29 January 2004; received in revised form 1 October 2004; accepted 2 February 2005

Abstract

A method for fusing multi-exposure images of a static scene taken by a stationary camera into an image with maximum information content is introduced. The method partitions the image domain into uniform blocks and for each block selects the image that contains the most information within that block. The selected images are then blended together using monotonically decreasing blending functions that are centered at the blocks and have a sum of 1 everywhere in the image domain. The optimal block size and width of the blending functions are determined using a gradient-ascent algorithm to maximize information content in the fused image.

Keywords: Image fusion; Image entropy; Image information; Image blending; Multi-exposure

1. Introduction

When irradiance across a scene varies greatly, there will always be over- or under-exposed areas in an image of the scene no matter what exposure time is used. Recorded scene irradiance usually varies from region to region depending on the local surface reflectance, surface orientation, and shadows. An over- or under-exposed area carries less information than the same area when well-exposed. To capture details about an entire scene, it is necessary to capture images at multiple exposures. This paper describes a method for fusing a set of multi-exposure images of a scene into an image where all scene areas appear well-exposed. This fusion process is not only helpful for scene understanding by humans but by computer vision systems also.

The proposed method is related to methods that map high-dynamic range images to low-dynamic range images for display on low-dynamic rang monitors. In a high-dynamic range reduction method, image intensities are mapped to new intensities to decrease global contrast while maintaining local contrast. A large body of work on high-dynamic range reduction is available with excellent surveys provided by DiCarlo and Wandell [4] and Devlin et al. [3]. If the range of intensities in a high-dynamic range image is subdivided into smaller ranges and each range is mapped to [0,255], a set of images will be obtained that appear like multi-exposure images. Therefore, by fusing multi-exposure images, dynamic range in an image can also be reduced.

In a related study, Mann [15] obtained a sequence of overlapping images of a scene by fixing the camera center with respect to the scene and slightly rotating the camera. The gain of the camera automatically varies with scene content as the camera is rotated, creating images with different brightness levels. The images are then registered by projective transformations [14], and the intensity at a pixel in the output is computed from a weighted sum of intensities of the same pixel in the input images. Smaller weights are assigned to very high and low intensities while higher weights are assigned to mid intensities. Robertson et al. [16] used a similar method but instead of relying on the automatic gain of a camera to create images with different brightnesses, images were captured with different exposure levels by changing the shutter speed of the camera.

In a method developed by Szeliski [18], first, an average image was created from a set of multi-exposure images where intensity at a pixel in the average image was the average of intensities of the same pixel in the multi-exposure images. Next, the histogram of the average image was computed and through a histogram equalization process, intensities in the average image were mapped to new intensities in such a way that the average intensity of

* Tel.: +1 937 775 5170; fax: +1 937 775 5133.
E-mail address: ardeshir@cs.wright.edu.

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.02.004

the image did not change. Histogram equalization is used to increase image contrast and compensate for the contrast lost during intensity averaging.

The method described here is based on the observation that if irradiance across a scene varies greatly, when images at different exposures are obtained, an image area may appear the best exposed in one image when compared to all other images. An under- or over-exposed image area carries less information than when the area receives the right amount of exposure. Contrary to the previous methods that use the average or weighted average of intensities of a pixel in input images to represent the intensity at the pixel in the output, in the proposed method, for each small image block the 'best-exposed' image among all the input images is selected to represent the block in the output. Minimal averaging is performed to smoothly combine the image blocks into an image. An image is considered best-exposed within an area if it carries more information about the area than any other image. Information content will be measured using entropy. Image entropy, a measure of uncertainty, was first used by Shannon [17] to measure the information contained in a message. Since then entropy has been used to measure the information contained in a digital image [5,9].

2. Approach

The problem to be solved is as follows: Given N images of a static scene obtained at different exposures using a stationary camera, it is required to combine the images into a single image that has the maximum information content without producing details that are non-existent in the given images.

The approach proposed here selects the most informative image for each local area and blends the selected images to create a new image. When irradiance across a scene varies greatly, images with various exposure levels are needed to capture all scene details. The main problem to be solved here is to identify the image that contains the most information within a particular local area. An image that is over- or under-exposed within an area does not carry as much information as an image that is well-exposed in that area.

In a gray-scale image, entropy is measured by

$$E_{\mathrm{g}} = \sum_{i=0}^{255} -p_i \log(p_i), \quad (1)$$

where p_i is the probability that an arbitrary pixel in the image has intensity i. To estimate $\{p_i: i=0,\ldots,255\}$, first the histogram of the image is computed. Assuming the number of pixels having intensity i is n_i and the image contains n pixels, $p_i=n_i/n$.

The human visual system is capable of discriminating far more colors than luminances [10]; therefore, it makes sense to compute image information using all color components rather than only the luminance component. Since we are interested in maximizing perceived information, the *RGB* colors are transformed to *CIELab* colors [2] and image information is computed using *Lab* color values. This transformation non-linearly maps colors in such a way that most colors distinguishable by the human visual system are obtained [12,13]. To determine the entropy of a color image, a 3D histogram that represents the frequency of colors appearing in the image is needed. Since such a histogram may have more elements than the number of pixels in the image itself, instead of using all colors in an image, the colors are clustered to the 256 most dominant clusters with centers $\{\mathbf{C}_i: i=0,\ldots,255\}$. An efficient algorithm for doing this has been given by Xiang [19]. Then, the color at each pixel is approximated by the color of the cluster center closest to it. Entropy of a color image is then approximated by

$$E_{\mathrm{c}} = \sum_{i=0}^{255} -p_i^{\mathrm{c}} \log(p_i^{\mathrm{c}}) \quad (2)$$

where p_i^{c} is the number of pixels within cluster i (with cluster center $\mathbf{C}_i$) divided by the total number of pixels in the image.

To determine the image that is best exposed within a local neighborhood, first the images are divided into blocks of size $d \times d$. d will be one of the parameters of the method to be determined by maximizing information content in the fused image. Image entropy E_{c} is determined for each block, and the image that provides the highest entropy within the block is selected to represent that block in the output. Optimal block size could vary from region to region but since the image blending process requires square blocks, if a well-exposed area in an image has an irregular shape, it can be assumed that the irregular area is divided into small square blocks. Since entropy is to be computed for each block, to obtain a meaningful measure, block size should not be smaller than 16 pixels.

If an image is created from the composition of image blocks obtained in this manner, an image will be obtained that may have sharp discontinuities across image blocks. Although each block will be informative, the image created from them will be awkward to view. An example is shown in Fig. 1. Fig. 1a–e shows a set of images of an office room taken at five different exposure levels. These images are of size 480×640 pixels. Dividing the images into blocks of size 160×160 pixels, we obtain an array of 3×4 image blocks. If the entropy within each block is calculated for the five input images, the block from the image that contains the highest entropy is cut out, and the blocks are put back together, the image shown in Fig. 1f will be obtained.

To remove discontinuities across image blocks, instead of cutting and pasting the blocks to obtain an image, the images representing the blocks are smoothly blended together. A monotonically decreasing blending function is centered at each block in the selected image and image colors are multiplied with the corresponding blending function values. A blending function assigns the maximum weight to the pixel at the center of a block and assigns

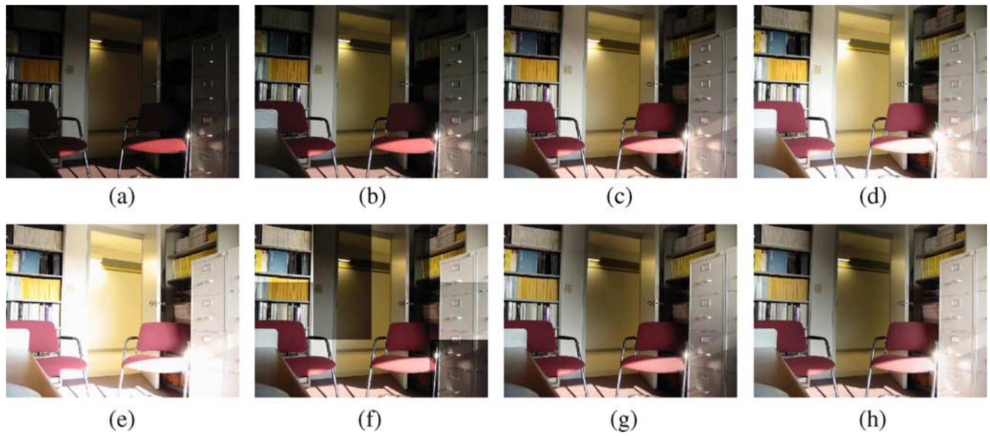

Fig. 1. (a)–(e) Images representing different exposures of an office room. (f) The image obtained by composing the 12 blocks cut out of the five images. (g) The image produced by centering the blending functions at the selected blocks in the images, multiplying the blending functions with image colors, and adding the weighted colors together. (h) The most informative image constructed from images (a)–(e). Optimal values found for d and σ are 224 and 128 pixels, respectively. These images are of size 768×1024 pixels.

weights to other pixels inversely proportional to their distances to the center of the block.

Assuming N images are given and each image has been subdivided into an array of $n_r\times n_c$ blocks, and assuming j and k denote the row and column indices of a block, and also letting $\mathbf{I}_{jk}$ denote the image that has the highest entropy among N images for block jk, we compute output image $\mathbf{O}(x, y)$ by

$$\mathbf{O}(x,y) = \sum_{j=1}^{n_r}\sum_{k=1}^{n_c} W_{jk}(x,y)\mathbf{I}_{jk}(x,y), \tag{3}$$

where $W_{jk}(x, y)$ is the value of the blending function centered at the jkth block at location (x, y), and $\mathbf{I}_{jk}(x, y)$ is a vector representing the actual Lab color coordinates of the image representing block jk at location (x, y) (before color clustering). Clustering is performed to facilitate computation of the entropies of the image blocks. Once the best-exposed image for each block is selected, the actual color values in the input images are used in (3). Note that the value at (x, y) in the output is determined from the weighted sum of input image values at (x, y), where the weight at the selected block is usually 80% or more and the sum of the weights associated with the remaining images is 20% or less. The color at a pixel within a block in the output, therefore, is very close to the color of the pixel in the best-exposed image within that block.

The number of images being blended is equal to the number of blocks and not the number of input images. When the blocks are small, many blocks may be selected from the same image. Also note that, rather than using image values within only the selected blocks, entire images are blended together, although the contribution of far away blocks on a pixel may be negligible. This process smoothly blends the images, avoiding the creation of discontinuity across blocks.

Since the blocks form a uniform grid, tensor-product basis functions, such as B-spline basis functions [6], may be used. When B-spline basis functions are centered at the blocks, the functions will smoothly blend the interior parts of the images. B-spline surfaces do not extrapolate beyond the border knots, which are positioned at the border block centers. They can blend image pixels up to the centers of the border blocks, but they cannot blend image pixels between the centers of the border block and the image borders. In order to blend image intensities across the entire image domain, it is necessary to use functions that are defined in terms of the basis or blending functions that do not have a limited support and rather have an infinite support, extending over the entire image domain. Examples of such functions are thin-plate or surface splines [7,11] and rational Gaussian (RaG) [8] surfaces. In this work, RaG surfaces will be used to blend the image intensities. RaG blending functions are defined by

$$W_{jk}(x,y) = \frac{G_{jk}(x,y)}{\sum_{m=1}^{n_r}\sum_{n=1}^{n_c} G_{mn}(x,y)}, \tag{4}$$

where n_r and n_c denote the number of image blocks vertically and horizontally, and $G_{jk}(x, y)$ represents the value of a Gaussian of height 1 centered at the jkth block at (x, y):

$$G_{jk}(x,y) = \exp\left\{-\frac{(x-x_{jk})^2+(y-y_{jk})^2}{2\sigma^2}\right\}. \tag{5}$$

(x_{jk}, y_{jk}) are the coordinates of the center of the jkth block, and σ is the standard deviation or the width of the Gaussians. In the following, we will refer to σ as the *width* of the blending functions.

Fig. 1g shows the image obtained by centering blending functions of width 80 pixels at blocks of size 160×160 pixels in the selected images and multiplying color values at image pixels with weights given by the blending functions and adding the weighted colors together. This is the kind of output we will obtain. σ is one other unknown parameter that has to be determined to maximize information content in the output. Since more than one image block in the output may point to the same input image, an input image may be used more than once in relation (3). The blending functions

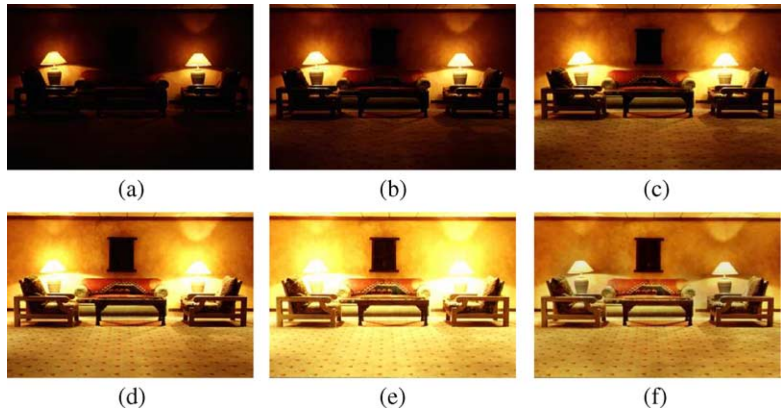

Fig. 2. (a)–(e) Images representing different exposures of an indoor scene. Entropies of the images are 2.82, 3.36, 4.21, 4.78, and 4.40, respectively. (f) The image obtained by blending the five images. Entropy of this image is 5.29. These images are of size 231×343 pixels, and optimal parameters found are $d=32$ pixels and $\sigma=32$ pixels.

blend the proper amounts of the input images to create the output image.

There are two parameters to be determined in the proposed method: d, image block size, and σ, the width of the blending functions. As these parameters are varied, different images are obtained. We have to determine the d and σ that maximize image entropy as computed by (1) or (2) depending on whether gray-scale or color images are provided. The optimal parameters are found using the gradient-ascent algorithm described below.

Algorithm BLEND:

1. Set parameters d, σ, and Δ to some guesses.
2. Determine the entropy of block jk in each image using formula (2) for $j=1,\ldots,n_r$ and $k=1,\ldots,n_c$ and let the image having the highest entropy within block jk be $\mathbf{I}_{jk}$.
3. Find the entropy of the image obtained by blending the images of the blocks selected in Step 2.
4. Increment or decrement d and σ by Δ in the gradient-ascent direction and repeat Steps 2 and 3 until the highest entropy is reached. Let the d and σ producing the highest entropy be $d_{\max}$ and $\sigma_{\max}$, respectively.
5. Select the images identified in Step 2 when setting $d=d_{\max}$, and create a new image by blending the selected images with blending functions of width $\sigma_{\max}$.

The initial values for parameters d and σ are set to the averages of optimal parameters found in a number of test runs of the algorithm. Applying this algorithm to images a–e, the image shown in Fig. 1h is obtained. As initial values, $d=\sigma=64$ pixels were used, and Δ was 32 pixels. $d_{\max}$ and $\sigma_{\max}$ obtained by the algorithm were 224 and 128 pixels, respectively. Smaller increments produce higher optimal entropies at a higher computational cost. For very large images, initially the increment Δ should be set to a large value to find estimates for the optimal parameters and then repeat the algorithm using a smaller Δ to refine the parameters. For small images, Δ may be taken small from the start.

3. Results and discussion

Experimental results of the proposed algorithm are shown in Figs. 2–7. The original images in Figs. 2–6 are courtesy of Shree Nayar, and the images in Fig. 7 are

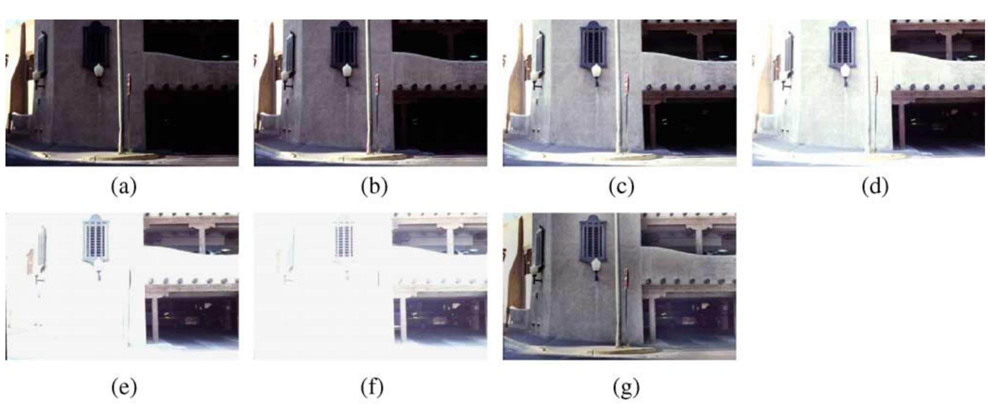

Fig. 3. (a)–(f) Images of a garage scene obtained at multiple exposures. (g) The image obtained by blending the six images. Entropies of images (a)–(g) are 3.02, 3.29, 5.04, 4.70, 4.03, 3.90, and 5.25, respectively. The optimal $d=16$ pixels and the optimal $\sigma=16$ pixels. These images are of size 222×348.

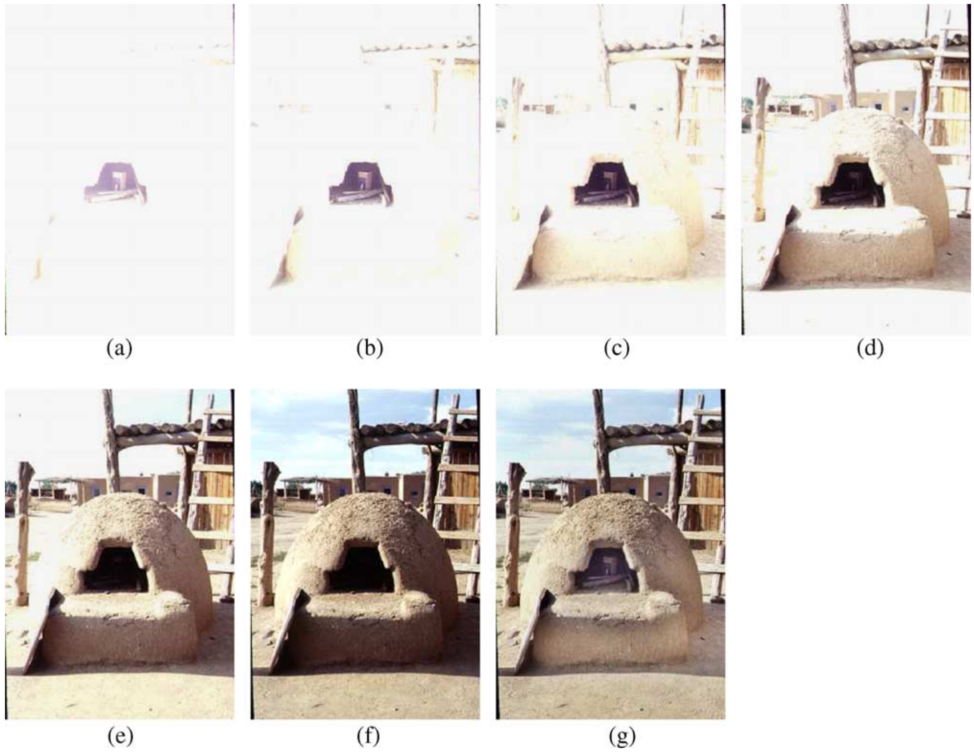

Fig. 4. (a)–(f) Images of an igloo scene obtained at multiple exposures. (g) The image obtained by blending the six images. Entropies of these images are 2.92, 3.02, 3.97, 4.39, 4.86, 5.18, and 5.38, respectively. The optimal values of d and σ are both 16 pixels. The images are of size 341×236 pixels.

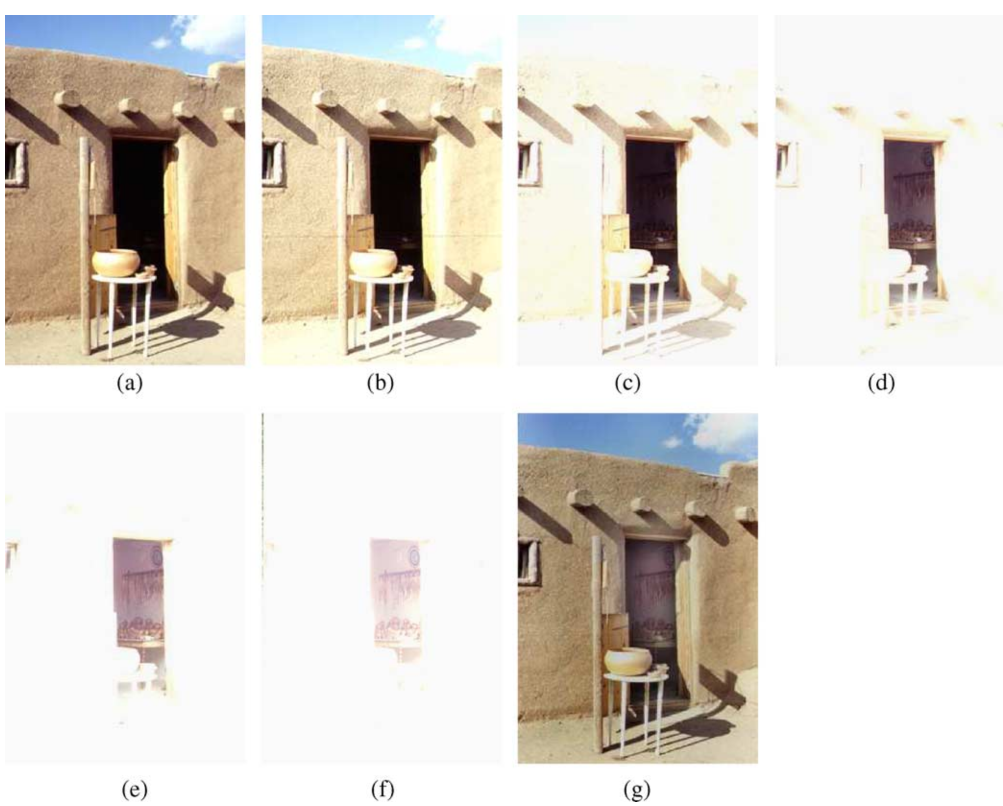

Fig. 5. (a)–(f) Images of a door to a dark room obtained at multiple exposures. (g) Image obtained by blending the six images. Entropies of the images are 5.09, 4.81, 3.86, 3.17, 3.04, 2.81, and 5.21, respectively. Optimal $d=16$ pixels and optimal $\sigma=16$ pixels. These images are of size 338×231 pixels.

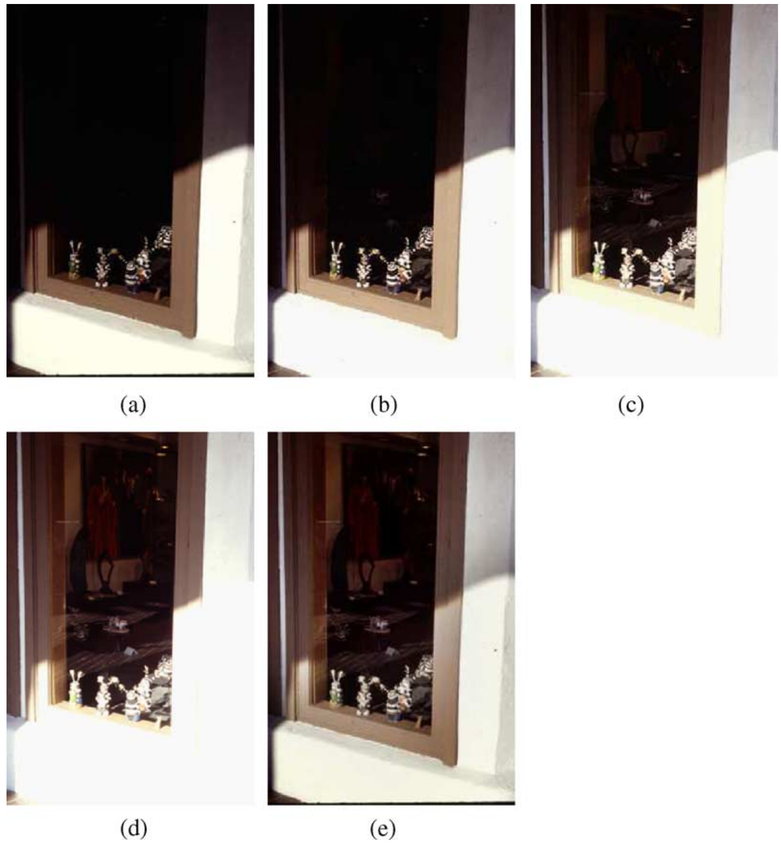

Fig. 6. (a)–(d) Images representing different exposures of a window scene. Entropies of the images are 3.24, 3.56, 4.31, and 4.20, respectively. (e) The image obtained by blending the four images. Entropy of this image is 5.14. Optimal values of d and σ are both 32 pixels. These images are of size 341×226 pixels.

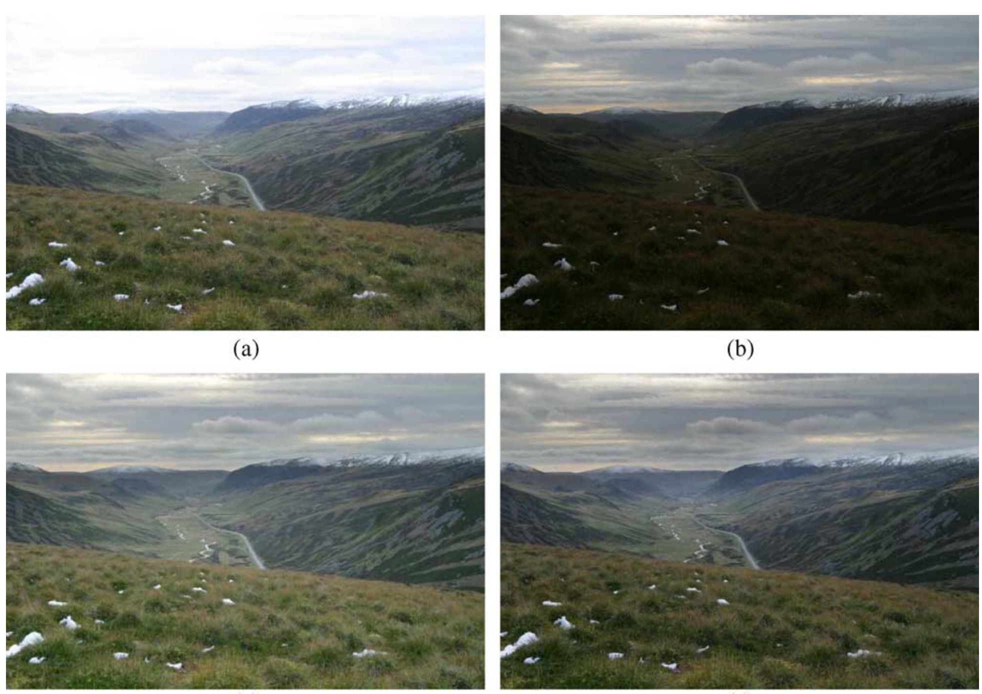

Fig. 7. (a) and (b) Two images of the Scottish Highlands representing different exposure levels. Entropies of these images are 4.86 and 4.34. (c) The image created by manual photographic techniques. (d) The image obtained automatically by the proposed method. Entropies of (c) and (d) are 5.22 and 5.20, respectively. Optimal values of d and σ are both 64 pixels. These images are of size 1024×1536 pixels.

courtesy of Max Lyons. Entropies of the original images and the resultant images are shown in the figure captions. Initial values of d and σ for images in Figs. 2–6 were both 32 pixels and Δ was 8 pixels. The initial d and σ for Fig. 7 were 128 and Δ was also 128. After finding the approximate optimal d and σ, the algorithm was run again by setting the initial d and σ to the obtained estimates and letting Δ to 16 pixels. This produced the result shown in Fig. 7d. In these experiments, when d reached 16 pixels, it was taken as the optimal block size without reducing it further.

Some of the input images in Figs. 2–6 have large well-exposed areas. Therefore, entropies of some of the input images are quite high, but the entropies of the output images are even higher. To compare the proposed automatic method with manual photographic methods, an example is given in Fig. 7. Results obtained manually by a professional photographer (Max Lyons) and by the automatic method are virtually indistinguishable and have very close entropies.

Optimal parameters d and σ vary from image to image. Images representing scenes with highly varying reflectances, highly varying surface orientations, and highly varying environmental factors such as shadows and specularities, produce smaller optimal d's and σ's than scenes with smoothly varying reflectances and smoothly varying surface orientations. d and σ are generally close, but examples to the contrary can be found. d depends on the sizes of well-exposed regions and σ depends on the rate the well-exposed areas transition to under- and over-exposed areas.

The computational complexity of Algorithm BLEND depends on: (1) image size, (2) initial block size d and the standard deviation of the Gaussian σ, and (3) increment Δ in the block size and the standard deviation. For the images shown in Figs. 2–6, computation time varied between 6 and 10 s. The large images shown in Fig. 7 (1024×1536 pixels) required about 1 min processing time. About one-third of the computation time in each case is spent on determining the 256 dominant colors in an image by clustering. The remaining two-thirds of the time is spent on iteratively computing the entropies of the individual blocks, selecting the images, and blending them. These times are measured on a Windows PC with a 1.8 GHz Pentinum IV processor and 512 MB RAM.

4. Concluding remarks

Image analysis techniques rely on critical image information, which may not be available in image areas that are over- or under-exposed. In situations where images at multiple exposure levels of a scene can be taken, the proposed image fusion may be used to combine the images into an image that is well-exposed everywhere and provides the critical information needed in a particular vision task.

The proposed fusion method preserves scene highlights if color information within a highlight area is quite high. A characteristic of the method is that it does not have a side effect and will not change the local color and contrast in the input. The contrast in an image area in the output cannot be higher than the contrast in the best-exposed image. If further contrast enhancement is needed, traditional methods such as inverse filtering [1] should be used.

In this work, entropy was used as the measure for optimization when fusing the images. Other measures may be more appropriate when fusing temporal, multi-sensor, and multi-focus images. Depending on the properties of the given images and the desired property of the fused image a proper measure should be used to select the images for blending.

Acknowledgements

The author would like to thank Shree Nayar and Max Lyons for providing the images shown in Figs. 2–7. The author also would like to thank the reviewers for their insightful comments. This work was supported in part by the Ohio Board of Regents.

References

[1] K.R. Castleman, Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1996. pp. 569–570.

[2] C. Connolly, T. Fliess, A study of efficiency and accuracy in the transformation from *RGB* to *CIELAB* color space, IEEE Transactions on Image Processing 6 (7) (1997) 1046–1052.

[3] K. Devlin, A. Chalmers, A. Wilkie, W. Purgathofer, Tone reproduction and physically based spectral rendering, Eurographics, STAR 2002; 101–123.

[4] J.M. DiCarlo, B.A. Wandell, Rendering high dynamic range images, Proceedings of the SPIE: Image Sensors 3965 (2002) 392–401.

[5] B.R. Frieden, Restoring with maximum likelihood and maximum entropy, Journal of the Optical Society of America 62 (1972) 511–518.

[6] W.J. Gordon, R. Riesenfeld, B-spline curves and surfaces in: R. Barnhill, R. Riesenfeld (Eds.),, Computer Aided Geometric Design, Academic Press, Inc., Orlando, FL, 1974. pp. 95–126.

[7] A. Goshtasby, Registration of images with geometric distortions, IEEE Transactions on Geoscience and Remote Sensing 26 (1) (1988) 60–64.

[8] A. Goshtasby, Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces, International Journal of Computer Vision 10 (3) (1993) 233–256.

[9] E.L. Hall, Almost uniform distributions for computer image enhancement, IEEE Transactions on Computers 23 (2) (1974) 207–208.

[10] E.L. Hall, Computer Image Processing and Recognition, Academic Press, New York, N.Y. 1979. pp. 29–30.

[11] R.L. Harder, R.N. Desmarais, Interpolation using surface splines, Engineering Notes 9 (2) (1972) 189–191.

[12] B. Hill, Th. Roger, F.W. Vorhagen, Comparative analysis of the quantization of color spaces on the basis of the CIELAB color-difference formula, ACM Transactions on Graphics 11 (4) (1992) 373–405.

[13] J.K. Kasson, W. Plouffe, An analysis of selected computer interchange color spaces, ACM Transactions on Graphics 11 (4) (1992) 373–405.

[14] S. Mann, Compositing multiple pictures of the same scene, Proceedings of 46th Annual IS&T Conference, Cambridge, MA, 1993.

[15] S. Mann, Joint parameter estimation in both domain and range of functions in same orbit of the projective-wyckoff group, Proceedings of IEEE International Conference on Image Processing (ICIP96), Lausanne, Switzerland, 1996, pp. 193–196.

[16] M.A. Robertson, S. Borman, R.L. Stevenson, Dynamic range improvement through multiple exposure, Proceedings of IEEE International Conference on Image Processing, vol. 3, Kobe, Japan, 1999, pp. 159–163.

[17] C.E. Shannon, W. Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, 1949/1963.

[18] R. Szeliski, System and process for improving the uniformity of the exposure and tone of a digital image, US Patent No. 6,687,400, Feb. 3, 2004.

[19] Z. Xiang, Color image quantization by minimizing the maximum intercluster distance, ACM Transactions on Graphics 16 (3) (1997) 260–276.