ILLUMINATION CORRECTION FOR IMAGE STITCHING

Sascha Klement, Fabian Timm and Erhardt Barth
Institute for Neuro- and Bioinformatics, University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck, Germany
Pattern Recognition Company GmbH, Innovations Campus Lübeck, Maria-Goeppert-Straße 1, 23562 Lübeck, Germany

Keywords: Illumination correction, Texture mapping, Image stitching.

Abstract: Inhomogeneous illumination occurs in nearly every image acquisition system and can hardly be avoided simply by improving the quality of the hardware and the optics. Therefore, software solutions are needed to correct for inhomogeneities, which are particularly visible when combining single images to larger mosaics, e.g. when wrapping textures onto surfaces. Various methods to remove smoothly varying image gradients are available, but often produce artifacts at the image boundary. We present a novel correction method for compensating these artifacts based on the Gaussian pyramid and an appropriate extrapolation of the image boundary. Our framework provides various extrapolation methods and reduces the illumination correction error significantly. Moreover, the correction is done in real-time for high-resolution images and is part of an application for virtual material design.

1 INTRODUCTION

Digital images in many ways suffer from deficiencies of the hardware that was used for capturing the image, including sensor, lenses, and illumination. Even with a perfect sensor, a lens without aberrations and a perfectly homogeneous illumination, the image may still show an intensity falloff towards the corners of the image due to natural vignetting. In general, the illumination inhomogeneity is much more complex (Aggarwal et al., 2001) and cannot be described with a simple model.

In various applications, such as 3D-texture mapping, aerial photography or astronomy, illumination gradients need to be removed when creating image mosaics. Without correction, transitions between single images would clearly be visible. These artifacts might mislead further image processing or pattern recognition systems, and therefore need to be reduced to a minimum.

The problem of illumination correction has been addressed by many authors, often with respect to a specific illumination artifact. Vignetting correction methods have been proposed e.g. in (Zheng et al., 2009) and (Kim and Pollefeys, 2008). In magnetic resonance imaging a method as been proposed based on information minimisation where the correction components are modelled as combinations of smoothly varying basis functions (Likar et al., 2001).

In face recognition, methods for illumination compensation have been proposed as a preprocessing step (Adini et al., 1997; Levine et al., 2004), primarily to find a more invariant face representation. Recently, a non-parametric shading correction method has been proposed in (Reyes-Aldasoro, 2009). An overview of stitching and blending methods can be found in (Levin et al., 2004).

We aim for an illumination correction method that not only removes inhomogeneities correctly without boundary artifacts, but which is also qualified for real-time applications such as virtual material design. Thus, we focus on a simple method based on lowpass filtering using Gaussian pyramids and appropriate image extrapolation to avoid boundary artifacts. First, we give a short introduction to Gaussian pyramids to show why boundary artifacts may occur. Then, we propose a framework that covers various standard boundary conditions for filtering and mention suitable extrapolation functions. Finally, we show the performance of our method with some practical examples.

2 THE GAUSSIAN PYRAMID

For more than two decades, pyramid methods have been used for image processing tasks such as image enhancement, compression, interpolation and extrap-

Klement S., Timm F. and Barth E..
ILLUMINATION CORRECTION FOR IMAGE STITCHING.
DOI: 10.5220/0003323300810086
In *Proceedings of the International Conference on Imaging Theory and Applications and International Conference on Information Visualization Theory and Applications* (IMAGAPP-2011), pages 81-86
ISBN: 978-989-8425-46-1

olation of missing image data and numerous others (Ogden et al., 1985).

Given a gray-valued input image with intensity values $G_0(x,y)$ at discrete locations $x \in [0, m-1]$ and $y \in [0, n-1]$ the Gaussian pyramid is an efficient data structure for spectral decomposition as follows. For each level i the image $G_i(x,y)$ is lowpass filtered and downsampled by a factor of two to produce the image $G_{i+1}(x,y)$. This is commonly referred to as the *Reduce* operation.

$$\begin{aligned} G_{i+1} = \mathrm{Reduce}(G_i) &= (\downarrow 2)(h * G_i) \quad \text{with} \\ (\downarrow 2) f(x,y) &= f(2x, 2y) . \end{aligned}$$

This is done successively up to a certain level l. Then, the *Expand* operation is applied successively to get an image with the same size as G_0, but containing only the frequency components of G_l:

$$G'_{i-1} = \mathrm{Expand}(\mathrm{G_i}) = 4h * ((\uparrow 2) G_i) .$$

The weighting function h is often called the *generating kernel* and is commonly chosen to be a 5-by-5 binomial filter to approximate the Gaussian. This method for computing the low frequency components of the input image is obviously more efficient than direct convolution with a large filter but also more efficient than using standard FFT (Adelson et al., 1984).

Finally, we need to decide how the image data is extended when filtering boundary pixels. This is the critical step where boundary artifacts occur. In the following, we present a general framework that covers not only the extrapolation conditions for rectangular images, but also arbitrary image shapes that occur, e.g., during image segmentation.

3 BOUNDARY EXTRAPOLATION

Since the filtering operation is defined as

$$(f * h)(x,y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) f(x-i, y-j)$$

the image needs to be extended by two pixels — either virtually or explicitly — in each direction when using 5-by-5 filters. Commonly, the pixels outside the image are set to a constant value or to the value of the nearest boundary pixel (replicate boundary) or are computed by assuming a periodic image (circular boundary).

Given the input image $f(x,y)$ with $x \in [0, m-1]$ and $y \in [0, n-1]$ we seek an extended image $f'(x,y)$ with $x \in [-2, m+1]$ and $y \in [-2, n+1]$. Next, we define for each pixel (x,y) a set of pixels

$$\mathcal{P}_{x,y} = \left\{ \left(p_i^x, p_i^y \right) \right\}_{i=1}^{|\mathcal{P}_{x,y}|}$$

that have an influence on the intensity at (x,y) in the extended image. Finally, the function $g_{\mathcal{P}_{x,y}}(x,y)$ defines how to calculate the intensity at (x,y) from the intensities of the set of pixels $\mathcal{P}_{x,y}$. Thus, we get

$$f'(x,y) = \begin{cases} f(x,y) & \text{if } x \in [0, m-1] \\ & \text{and } y \in [0, n-1] \\ g_{\mathcal{P}_{x,y}}(x,y) & \text{otherwise.} \end{cases} \quad (1)$$

For convenience, we use $g(x,y)$ instead of $g_{\mathcal{P}_{x,y}}(x,y)$. Next, we show how the standard boundary conditions — replicate and circular — as well as different extrapolation methods fit into this framework.

3.1 Replicate Boundary

Assuming a replicate boundary, the intensity values of pixels outside of the image equal those of the nearest boundary pixel:

$$\mathcal{P}_{x,y}^{\mathrm{rep}} = \left\{ \left(p_0^x, p_0^y \right) \middle| \begin{array}{l} p_0^x = \min(\max(x,0), m-1) \\ p_0^y = \min(\max(y,0), n-1) \end{array} \right\} .$$

Thus, $\left| \mathcal{P}_{x,y}^{\mathrm{rep}} \right| = 1$ for all x and y, and $g^{\mathrm{rep}}(x,y) = f(p_0^x, p_0^y)$. Analogously, a circular boundary condition can be defined:

$$\mathcal{P}_{x,y}^{\mathrm{circ}} = \left\{ \left(p_0^x, p_0^y \right) \middle| \begin{array}{l} p_0^x = x \bmod m \\ p_0^y = y \bmod n \end{array} \right\} .$$

Again, $|\mathcal{P}_{x,y}^{\mathrm{circ}}| = 1$ for all x and y and

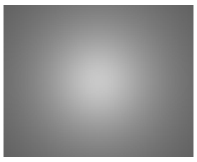
$$g^{\mathrm{circ}}(x,y) = f(p_0^x, p_0^y) .$$

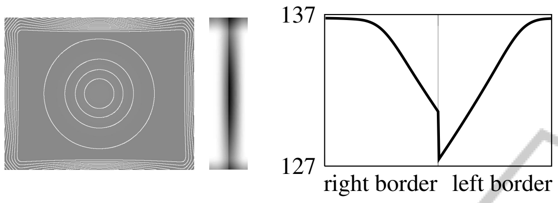
3.2 Simple Linear Extrapolation

Both replicate and circular boundary assumptions, are no valid assumptions when modelling illumination gradients. In almost all cases of natural illumination inhomogeneity the intensity gradient will continue smoothly outside of the image bounds. When compensating for vignetting, the application of replicate or circular boundary assumptions will overestimate the intensity gradient at the boundary.

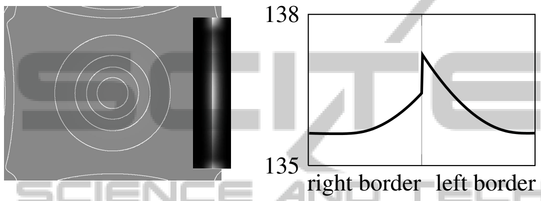
A first step towards more realistic boundary assumptions involves simple linear extrapolation of the intensity values at the boundary. The gradient is the difference between a boundary pixel and its next neighbour towards the centre of the image. Mathematically, this can be expressed in the following way:

$$\begin{aligned} \mathcal{P}_{x,y}^{\mathrm{lin}} &= \left\{ \left(p_0^x, p_0^y \right), \left(p_1^x, p_1^y \right) \right\} \quad \text{with} \\ p_0^x &= \min(\max(x,0), m-1) \\ p_0^y &= \min(\max(y,0), n-1) \\ p_1^x &= \min(\max(x,1), m-2) \\ p_1^y &= \min(\max(y,1), n-2) \quad \text{and} \end{aligned}$$

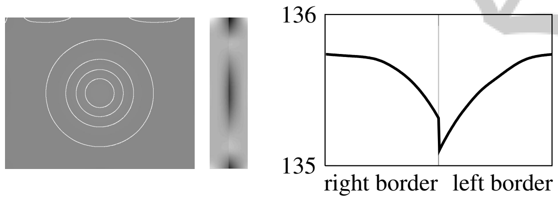

(a) Synthetic input image (natural vignetting).



(b) Replicate condition.



(c) Linear extrapolation.



(d) Polynomial extrapolation.

Figure 1: Illumination correction without texture. The intensity levels of the corrected images (left) are emphasised with contours. Additionally, an image transition (middle column, 128 pixels from each side, intensity stretched to full range) and the intensity across the middle row (right) are shown. Note the different scales on the y-axis and that the boundary artifacts are reduced.

$$g^{\text{lin}}(x,y) = f\left(p_0^x, p_0^y\right) + \left(f\left(p_0^x, p_0^y\right) - f\left(p_1^x, p_1^y\right)\right) \cdot \left\| \begin{pmatrix} x - p_0^x \\ y - p_0^y \end{pmatrix} \right\|_1 ,$$

where $\|\cdot\|_1$ denotes the Manhattan distance. This modification improves the illumination compensation significantly and can be extended, e.g., to least-squares regression with two-dimensional second order polynomials as shown below.

3.3 Least Squares Regression

We define the set $\mathcal{P}_{x,y} = \left\{\left(p_i^x, p_i^y\right)\right\}$ to contain a 5-by-5 block of pixels within the image bounds that minimises the mean distance to the point (x,y). Then, we can formulate the following minimisation problem

$$\min_\beta \quad \|\mathbf{A}\beta - \mathbf{b}\|_2 \quad \text{with}$$

$$\mathbf{A} = \begin{pmatrix} p_0^x & p_0^y & (p_0^x)^2 & (p_0^y)^2 & p_0^x p_0^y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{24}^x & p_{24}^y & (p_{24}^x)^2 & (p_{24}^y)^2 & p_{24}^x p_{24}^y & 1 \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} f(p_0^x, p_0^y) \\ \vdots \\ f(p_{24}^x, p_{24}^y) \end{pmatrix} .$$

This optimisation problem is solved by:

$$\beta = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b} .$$

The intensity at (x,y) is approximated by

$$g(x,y) = \begin{pmatrix} x & y & x^2 & y^2 & xy & 1 \end{pmatrix} \cdot \beta .$$

Note that other regression approaches may also be used. However, the complexity of a regression model, i.e. the number of parameters to be estimated, should be low, as illumination gradients are by definition smooth functions. The choice of the neighbourhood size is a trade-off between stability and runtime, but, as our experiments show, 5-by-5 blocks are sufficient to remove almost all visible boundary artifacts. The computational complexity of the regression method is no crucial factor, as extrapolation is done only for a low number of boundary pixels. So, other operations such as filtering of the whole image dominate the complexity of the illumination correction method.

3.4 Arbitrary Boundary Shapes

In some cases, it might be necessary to remove illumination gradients from images that contain more than one texture. Here, the different textures should be segmented and then individually corrected. In general, these regions are non-rectangular, so the above methods have to be adapted for arbitrary shaped images. Now, the input image $f(x,y)$ is defined only in certain regions, i.e. the image contains valid intensities for a set of pixels $I \subset [0, m-1] \times [n-1]$. Then, (1) simply needs to be modified to

$$f'(x,y) = \begin{cases} f(x,y) & \text{if } (x,y) \in I \\ g_{\mathcal{P}_{x,y}}(x,y) & \text{otherwise.} \end{cases} \quad (2)$$

In this more general case, it is not feasible to pick $\mathcal{P}_{x,y}$ from a fixed-size neighbourhood of a certain shape, e.g. when I is very sparse or has a fractal-like boundary. $\mathcal{P}_{x,y}$ should rather contain a fixed number of adjacent pixels from within I.

3.5 The Final Algorithm

So far, we focused on the boundary extrapolation during filtering, but the aim is to correct illumination inhomogeneities in texture images. In the overall algorithm (see Alg. 1), extrapolation of boundary pixels is used in steps (4) and (13).

Algorithm 1: Illumination correction algorithm.

```
    input  : Intensity image $f(x,y)$, pyramid depth $l$
    output: Corrected image $f'(x,y)$
 1  $G_0 \leftarrow f$;
 2  for $i \leftarrow 0$ to $l$ do
 3      Extend $G_i$;
 4      Extrapolate image $G_i$; store results in $G_i$;
 5      Filter image, i.e. $G_i \leftarrow h * G_i$;
 6      Crop image $G_i$ to original size;
 7      Downsample, i.e. $G_{i+1} \leftarrow (\downarrow 2)(G_i)$;
 8  end
 9  $G'_l \leftarrow G_l$;
10  for $i \leftarrow l$ down to 1 do
11      Upsample, i.e. $G'_{i-1} \leftarrow (\uparrow 2)G'_i$;
12      Filter image, i.e. $G'_{i-1} \leftarrow h * G'_{i-1}$;
13      Extrapolate boundary; store results in $G'_{i-1}$;
14      Scale image, i.e. $G_{i-1} \leftarrow 4 \cdot G_{i-1}$;
15  end
16  Calculate mean intensity, i.e. $\overline{f} = \frac{\sum_x \sum_y f(x,y)}{nm}$;
17  $f' \leftarrow f - G'_0 + \overline{f}$;
```

In the downsampling loop, the image is first extended by two rows and columns in each direction and then the intensity values of these pixels are determined by extrapolation. After filtering, the image is cropped to the original size. In the upsampling loop, the image is first filtered and then the intensity values of the first and last two rows and columns are recalculated from the inner image by extrapolation. Finally, the low-frequency image is subtracted from the original image and the mean intensity of the input image is added to obtain the final image with the proper intensity level.

4 EXPERIMENTS

The performance of our method is demonstrated in a series of experiments. We used gray-valued images with intensity values ranging from 0 to 255. All operations where done in double-precision arithmetic to avoid discretisation artifacts.

We started with artificial images that contained no texture but only smooth intensity gradients as they occur with natural vignetting (see Fig. 1). Perfect illumination correction would result in a completely homogeneous image. Obviously, the choice of the boundary condition does not affect the centre of the image, which shows a slight gradient even after correction. At the image boundary the replicate condition is outperformed by linear and polynomial extrapolation, the latter yielding an image intensity range of less than 1 at the boundary. Thus, the correction is perfect at the boundary for typical 8-bit images.

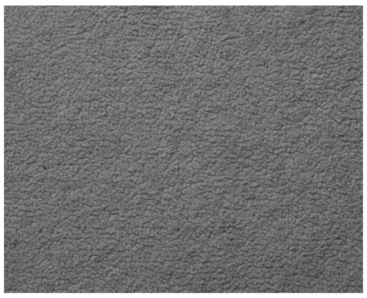
In a second experiment, we measured how boundary artifacts affect the histogram of the intensity values. Therefore, we calculated the difference between the histograms of the image boundary (64 pixels wide) and the image centre (see Fig. 2). Here, the replicate condition causes the corrected image to be too dark at the boundary, i.e. the histogram of the boundary pixels is shifted towards zero and the histogram difference has a maximum for small values. With linear extrapolation, this effect is dramatically reduced. Polynomial extrapolation gives only small additional improvement.

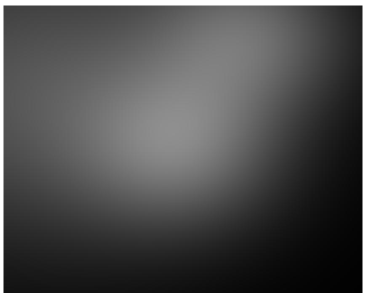
In Fig. 3 some typical textures are depicted as they occur in virtual material design. The images were acquired with an industrial colour camera (1392×1040 pixels) and show clearly visible intensity falloffs towards the image corners. We applied the above described method for each colour channel individually and compared it to the repetitive boundary condition. After correction the intensity gradients are no longer visible, however, other inhomogeneities and artifacts are still obvious. These range from clearly visible repetitive structure (see rows 2 and 3), to very subtle blurring towards the image corners due to lens deficiencies.

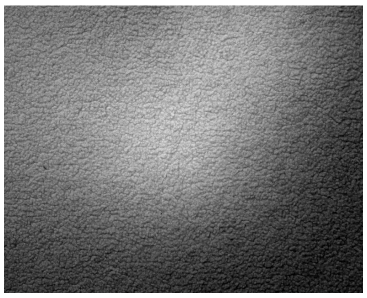
5 CONCLUSIONS

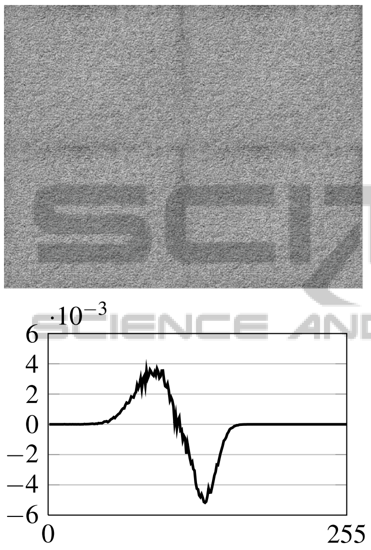
We presented a novel method for removing illumination inhomogeneities from texture images based on the Gaussian pyramid. With standard filtering using the replicate or circular boundary condition, the resulting images will show artifacts at the image borders — mostly visible when stitching images together. These artifacts can be avoided by different types of boundary extrapolation. The framework is not limited to linear or polynomial extrapolation, however, it seems unnecessary to use more complex functions.

The Gaussian pyramid provides a fast way to model arbitrary illumination gradients. All calculations can be done in real-time — the proposed boundary extrapolation techniques do not cause a significant performance drop due to their simplicity and the low

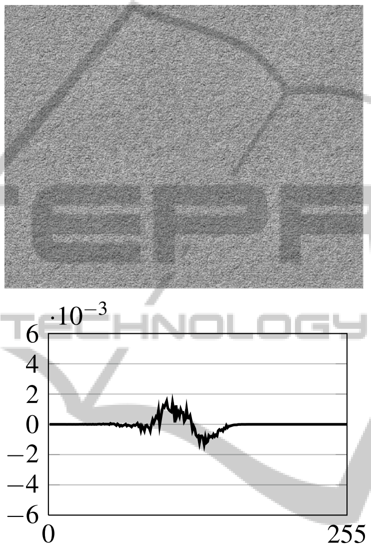

(a) Texture.


(b) Gradient.

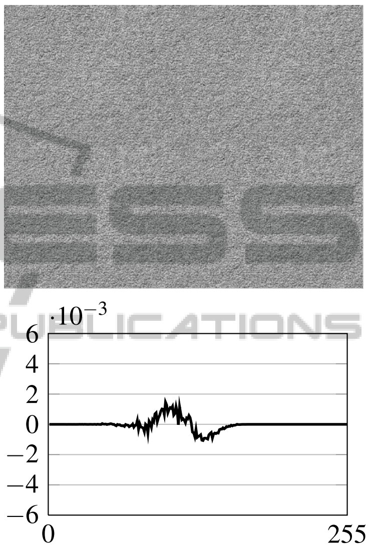

(c) Combined.



(d) Replicate condition.



(e) Linear extrapolation.



(f) Polynomial extrapolation.

Figure 2: Boundary artifacts in image mosaics. The top row shows a texture (a), a simulated illumination gradient (b) and the combination of both (c). The middle row shows the correction results for three methods as 2-by-2 image mosaics. The replicate condition (d) produces dark areas at the image transitions. With linear (e) or polynomial extrapolation (f) the transitions are almost invisible. The difference between the histograms of centre and boundary pixel intensities (bottom) verifies this observation.

number of image boundary pixels. For colour images, we apply the illumination correction for each channel separately. On current PCs, we reach 20 frames per second for SXGA colour images (1280×1024 px).

Finally, we shall discuss the role of illumination correction in texture synthesis as it is used in virtual material design. Besides intensity gradients, other image acquisition artifacts, such as blurred image regions, reflexions, or saturation may occur. All of them will cause the results of naïve image stitching to look repetitive and unrealistic, especially at the image borders where the textures are non-continuous. One solution would be to enhance the original image and remove as many artifacts as possible. However, a texture synthesis approach as proposed by Efros and Freeman (Efros and Freeman, 2001) could avoid all the above image acquisition artifacts by synthesising larger textures from smaller blocks from the original image. Each new block is chosen to optimise an overlapping region and, finally, the blockiness of the boundary is reduced by finding the minimum cost path within the overlapping region. In this scenario, illumination correction is again a fundamental preprocessing step to avoid intensity gradients. Thus, our method, in connection with the above texture synthesis, results in natural looking textures of arbitrary size and shape.

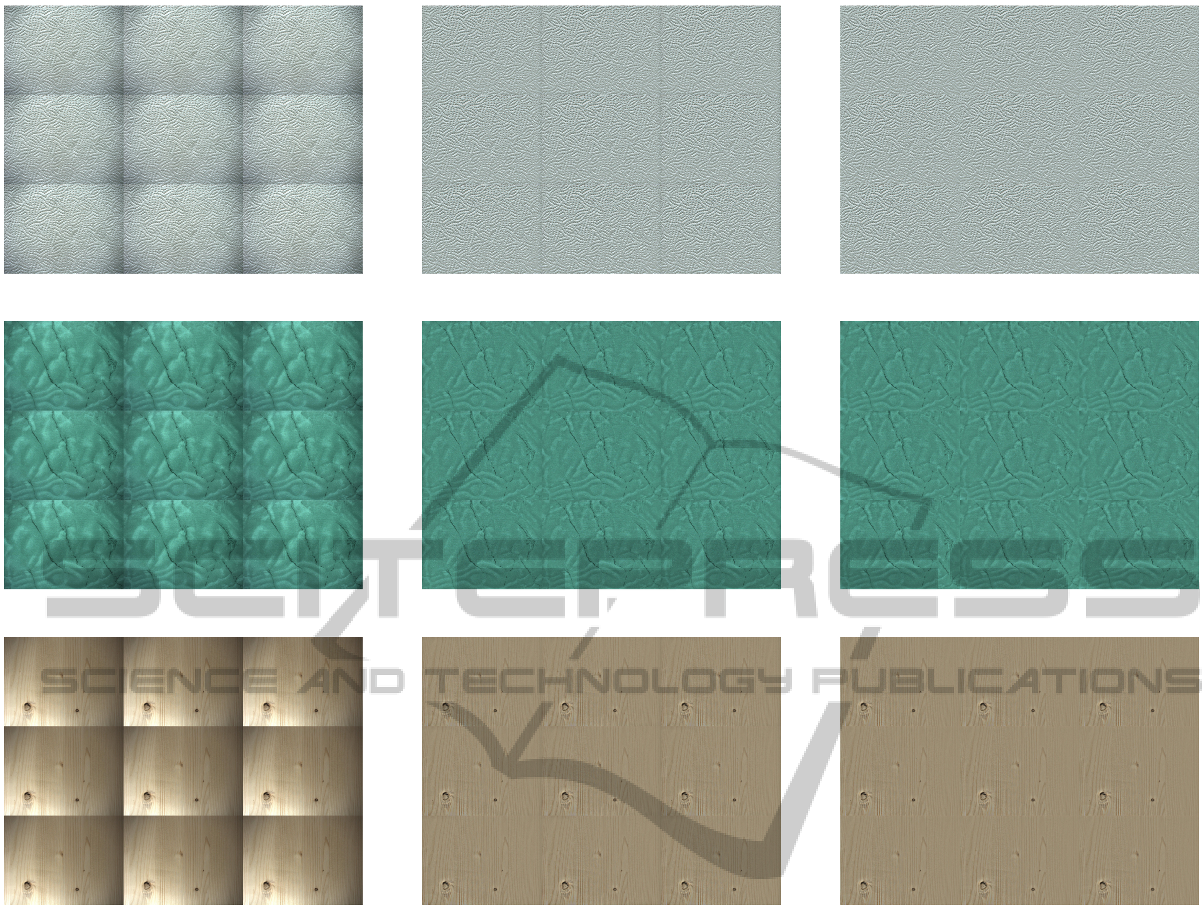


Figure 3: Real world colour textures. The performance of the illumination correction is shown for some real-world textured materials (from top to bottom: paper towel, green linoleum, wood) in 3-by-3 image mosaics. Without illumination correction (left column) vignetting is most obvious. The middle column shows image mosaics after correction with a replicate boundary condition. Obviously, the correction using polynomial boundary extrapolation (right column) gives the best results.

REFERENCES

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. (1984). Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41.

Adini, Y., Moses, Y., and Ullman, S. (1997). Face recognition: the problem of compensating for changes in illumination direction. *IEEE TPAMI*, 19:721–732.

Aggarwal, M., Hua, H., and Ahuja, N. (2001). On cosine-fourth and vignetting effects in real lenses. In *ICCV*, pages 472–479.

Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346, New York, NY, USA. ACM.

Kim, S. J. and Pollefeys, M. (2008). Robust radiometric calibration and vignetting correction. *IEEE TPAMI*, 30(4):562–576.

Levin, A., Zomet, A., Peleg, S., and Weiss, Y. (2004). Seamless image stitching in the gradient domain. In *ECCV*, pages 377–389. Springer-Verlag.

Levine, M. D., Gandhi, M. R., and Bhattacharyya, J. (2004). Image normalization for illumination compensation in facial images. unpublished.

Likar, B., Viergever, M. A., and Pernuš, F. (2001). Retrospective Correction of MR Intensity Inhomogeneity by Information Minimization. *IEEE Transactions on Medical Imaging*, 20(12):1398–1410.

Ogden, J. M., Adelson, E. H., Bergen, J. R., and Burt, P. J. (1985). Pyramid-based computer graphics. *RCA Engineer*, pages 4–15.

Reyes-Aldasoro, C. (2009). Retrospective shading correction algorithm based on signal envelope estimation. *Electronics Letters*, 45(9):454–456.

Zheng, Y., Lin, S., Kambhamettu, C., Yu, J., and Kang, S. (2009). Single-image vignetting correction. *PAMI*, 31(12):2243–2256.