

A Center Symmetric Padding Method for Image Filtering

Mengqin Li^(✉) and Xiaopin Zhong

College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, China
mengqin_li@126.com, xzhong@szu.edu.cn

Abstract. Padding the boundary is the first step in image filtering. If not appropriately handled, it often cause serious artifacts and losing information. In this paper an optimized boundary padding method is proposed to solve this problem. We analyze the pixel values near the boundary, calculate the gradient of the pixels, and pad the input image with a center symmetric padding method. Consequently, the details of the boundary will be preserved, and the number of unexpected noise will be suppressed as much as possible. We demonstrate that the center symmetric padding method is more effective than the traditional border handling methods in some computer vision applications.

1 Introduction

In computer vision, most of applications have to do some preprocessing to reduce noise or get useful information. Image filtering is a general and effective method in image processing such as non-linear filter in [1] and most of linear filter. Image filtering is generally modeled as a convolution of a pixel's neighborhood. The convolution operator makes use of not only the image in the Field of View (FOV) of the given observation, but also part of the scenery in the area bordering it. The missing pixel information outside the observed image can be synthesized by extrapolating the available image data.

In this paper, we propose a boundary padding application when dealing with the flaw detection of the backlight plate in cell phone. The picture of the backlight plate is taken from the line-scan CCD camera, whose resolution is 8192*14000, see in Fig. 3. In this case, padding the boundary which is the first step in image filtering plays an important role in such an extremely high resolution. Border handling provides approximate output pixels close to the boundary of the image where insufficient data is available to fully compute the local operator. There are many available methods to padding the boundary. The "tile" method in [2] proposed an algorithm that pads the image by replicating small tiles of pixels adjacent to the border to reduce boundary artifacts in image deconvolution. Halide*'s [3, 4] library also includes "tile". Tile padding is particularly useful for Bayer coded color images because it preserves the Bayer color pattern. Most of functions in commercial software or open source library are copying a given image onto another slightly larger image and then automatically padding the boundary, such as Reflect, Reflect101 [5], Replicate [6], Wrap, Constant. But these boundary methods not always work well, especially when the blur filter window size is relatively larger with a high resolution of the backlight plate.

In this paper, we proposed a simple but effective method called center symmetric padding method. First of all, we analyze the padded region by calculating the gradient of the pixel values in 2 dimension. If the gradient is too sharp for traditional border handling method, we pad the boundary with our optimized center symmetric padding method. In case of causing unexpected noise, we blur the padded region. Hence, the information of the boundary will be kept, and the number of unexpected noise will be decreased as much as possible.

2 The Proposed Method

We define the center symmetric method as below: First of all, pad the boundary of the objective picture using the default padding method which is border reflect101 in OpenCV to make border in a given filter window. Then extract the pixel values and coordinate of each row or column near the boundary into a 2 dimensional plane, calculate the mean gradient of these discrete points. If the gradient of the line near the boundary is over than the threshold, using the symmetry method to process the padded boundary. Finally, we apply a local linear model to blur the padded boundary in case of sharp change pixels. So, it is necessary to collect some pixels next to the boundary to predict the gradient of the pixel values in a row, using a minimum mean-square error (MMSE) method.

Assume \hat{Y}_i is a vector of n predictions of the padded boundary, and Y_i is the vector of observed values corresponding to the inputs to the function which generated the predictions, then the MSE of the predictor can be estimated by (1)

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \tag{1}$$

In this paper, the pixel values near the boundary in a row or column can be considered as a line in 2 dimension. Generally speaking, we can predict the pixel values outside the input image and padding the boundary, using the linear regression. For example, we choose to pick some pixels near the boundary in a row. We suppose the gradient of the fitted line which can be calculated as K in (2), and suppose the function of the fitted line is $f(x) = Kx + b$.

$$K = \frac{\sum_{i=1}^N (X_i - \frac{\sum X_i}{n})(Y_i - \frac{\sum Y_i}{n})}{\sum_{i=1}^N (X_i - \frac{\sum X_i}{n})^2} \tag{2}$$

After getting the parameter of the fitted line in linear regression, we can padding the boundary using the fitted line. Suppose we choose the 101*101 operator to filter the input image, the 50 pixels outside the boundary should be padded according to the rule of the center symmetric method: First, we can handle the boundary by reflect101, which can be used in OpenCV or MATLAB; then, reflect the padded boundary by a horizontal

line $f(x) = b$. Finally, we apply a local linear model to blur the padded boundary. We assume that $f(x, y)$ is the pixel values of the input boundary, $h(x, y)$ is the local operator which can be called kernel of the blurring, and $g(x, y)$ is the result pixel values in (3):

$$g(x, y) = \sum_{k,l} f(i + k, j + l)h(k, l) \tag{3}$$

In this paper, we take the horizontal padding as an example and illustrate it in detail. Figure 1 shows an example of a row pixel values in horizontal dimension. The region where the pixel values larger than 80 is the foreground, which is our object region. We can see that the pixel values decrease slowly around the boundary.

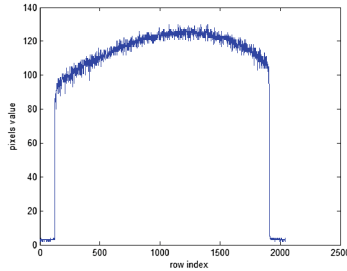


Fig. 1. Illustration of the pixel values distribution in horizontal dimension. The pixel values of the foreground has a sharp change adjacent to the boundary.

Our task is to detect flaws on the digital image of an illuminated screen, which is taken from a CCD camera. First of all, we need to find out the size and location of object region in the input image and crop it from the source image. After getting the object region, we start to do some image processing to locate the flaws of the backlight. According to the statistical results of the backlight, there are many kinds of flaws such as black or white dot, scratch, foreign objects, small or big shadow. In all of these flaws, big shadow is the most difficult problem on account of its ambiguous edges. In order to find out the big shadow flaw whose area is generally more than 20*20 pixels, the window of image filter operator must be set larger enough. Before the filtering operate, the appropriate padding method for image boundary handling is needed. In Table 1, it shows some typical methods for image padding on the boundary. As it mentioned in

Table 1. Image padding methods

Border Types	
Constant	i i i i i l a b c d e f g h l i i i i i
Replicate	a a a a a l a b c d e f g h l h h h h h h
Reflect	f e d c b a l a b c d e f g h l h g f e d c
Wrap	c d e f g h l a b c d e f g h l a b c d e f
Reflect101	g f e d c b l a b c d e f g h l g f e d c b

introduction, different kinds of padding methods show different characters and lead to different results.

In order to visualize the differences of these kinds of padding and find out an appropriate padding method, we pad the source image and plot the pixel values of it on MATLAB. So the padded image results are in Fig. 2.

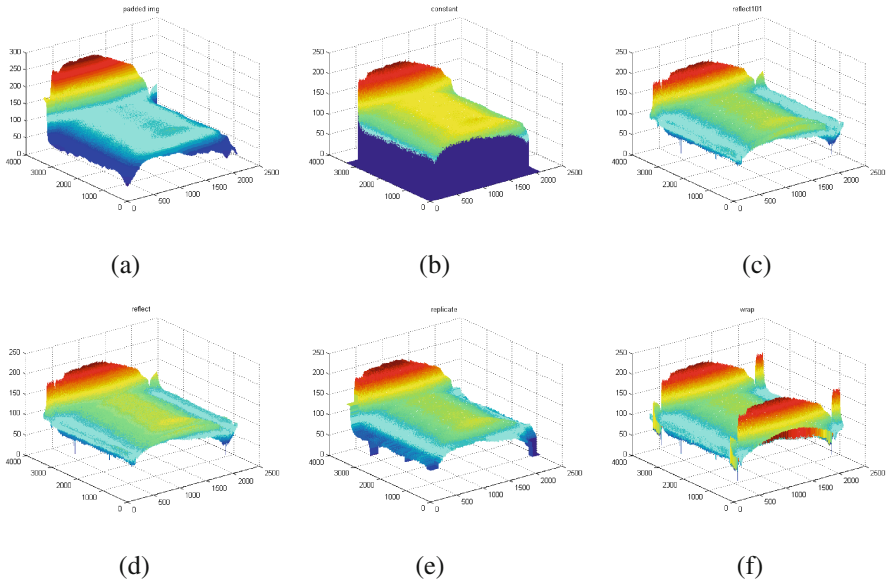


Fig. 2. Padded images in MATLAB with different border methods: (a) image deal with the proposed center symmetric method; (b) constant padded image; (c) reflect101 padded image; (d) reflect padded image; (e) replicate padded image; (f) wrap padded image

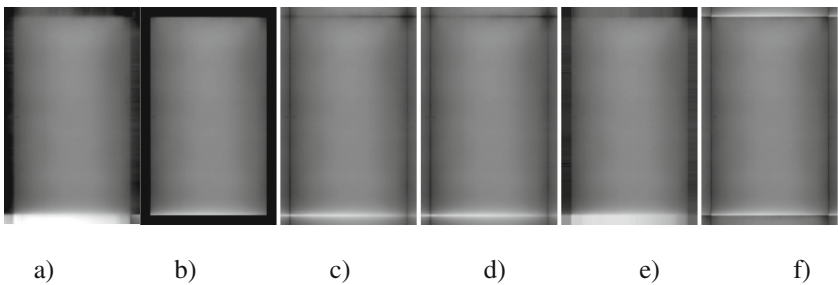


Fig. 3. (a) center symmetric padded image; (b) constant padded image; (c) reflect101 padded image; (d) reflect padded image; (e) replicate padded image; (f) wrap padded image

As the Fig. 2(a) and (e) shows, replicate border handling copies the boundary pixel values of an image or specified value and extends the values to the padded regions. From Table 1 and Fig. 2(b), (c), and (d), there are abrupt gradient changes around the boundary. Because of the abrupt gradient changes, the pixel values of the boundary after filtering

is much larger than the source pixel values. In this case, the boundary after padding becomes an inflection point in two dimension. Therefore, the boundary of the image are predicted as flaw by mistake, and the information of the flaw near the boundary are missing as well. When handling boundary artifacts, there are many algorithms are available such as fast image restoration without boundary artifacts [7, 8] and reducing boundary artifacts method [2]. But a significant majority of them are aimed at reducing the boundary artifacts when image filtering rather than the image padding.

As it shows above, we can draw a conclusion that most of the boundary padding methods can't solve the boundary problem efficiently. In addition, we find out a solution that can padding the boundary smoothly and preserve the boundary information, at the same time, without sacrificing the flaw information near the boundary.

3 Simulation

According to the algorithm presented in previous section, experiments are applied to verify the proposed padding method. Experiment platform is based on MATLAB and Visual Studio. In order to visualize the differences of these padding methods, we padded the border of the backlight plate which is cropped from the row image and applied canny edge detection. The subjective evaluation of the proposed method shows that it's effective comparing to traditional methods. In addition, objective evaluation of the effect of center symmetric padding method proves the algorithm validation.

As a concrete example of this center symmetric method, we try to solve the problem about a backlight plate flaw detection. Many approaches of padding has been taken to operate the source image about backlight plate, while the output boundary present different results respectively. After padding the boundary, it's time to extract the feature of the object, the flaws of the backlight plate. Edge correspond to abrupt changes or discontinuities in certain image properties between neighboring areas [9]. Canny edge detection in image processing is very sensitive to noise as well as the flaws in backlight plate. In this case, canny edge detection is an appropriate way to observe the differences of padding methods in the backlight plate flaw detection. We use canny edge detection on the padded image.

As the pictures show above, we can draw a conclusion that when the filter window size is too large for the traditional border methods to handle, and the gradient of the pixel values decreases around the boundary, our proposed optimized padding methods can fix the problem to some extent. In order to compare these method more intuitively, we apply a canny edge detection to evaluate these methods. After getting the canny results of these methods, we calculate the number of nonzero pixels as the statistical result. As shown in Table 2, our optimized padding method cause less artifacts comparing to others.

Table 2. Comparison on the number of Nonzero pixels in different canny images after padding.

	Center symmetric	Replicate	Reflect	Reflect101	Wrap	Constant
Nonzero	2190	9690	5076	5010	4066	3795

In order to compare the results of these padding images respectively, we choose the part which has distinct difference from the backlight plate image. In Fig. 4, there are the bottom left side of the backlight plate image after canny edge detection. In Fig. 4(b) and (f), the image applied constant and wrap border method remains an obvious line between the border and the boundary of the backlight plate owing to the sharp gradient. In Fig. 4(e), the replicate method makes the pixel values difference of each row or column more significant. From Fig. 4(a), (c) and (d), the boundary of center symmetric padding method is not so obviously than the reflect and reflect101, owing to the more gentle gradient on the boundary.

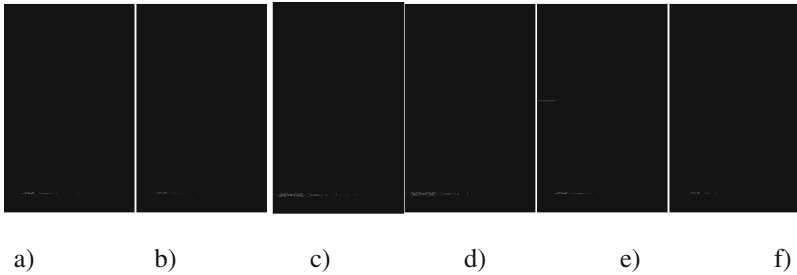


Fig. 4. The bottom left part of padded image after canny edge detection with different border methods: (a) center symmetric image; (b) constant padded image; (c) reflect101 padded image; (d) reflect padded image; (e) replicate padded image; (f) wrap padded image

Furthermore, in order to evaluate the effect of the proposed center symmetric, the PSNR (Peak Signal Noise Ratio) and MSE (Mean Square Error) are used to be objective evaluation parameters. PSNR is most commonly used to measure the quality of image reconstruction. A higher PSNR and lower MSE generally indicates that the reconstruction is of higher quality. Table 3 present the comparison on PSNR and MSE of center symmetric padded image and traditional padding methods respectively.

Table 3. Comparison on PSNR and MES among mentioned padded image

	Center symmetric	Replicate	Reflect	Reflect101	Wrap	Constant
PSNR	43.3643	43.0085	42.5242	42.5056	40.6914	33.6513
MSE	2.9968	3.2526	3.6363	3.6519	5.5455	28.0509

As the result shows above, the optimized center symmetric padding method has the biggest PSNR value, and the smallest MSE value in all the mentioned traditional methods. Therefore, we suspect our optimized algorithm, center symmetric padding method, is very effective. At the same time, it can preserve the edge detail information while not cause boundary artifacts.

4 Conclusion

In this paper, we present an optimized padding method which can be widely applied in image processing of computer vision. Different from the traditional border padding methods, we find out a center symmetric method which has a more gentle gradient on the boundary of the picture. And it can handle the abrupt changes near the boundary more smoothly. Since the center symmetric method take the pixels around the padded border into account and analyze the gradient of the border, the padded image might preserve more information of the image boundary.

The proposed center symmetric method shares a common limitation of the other border handling methods which is not directly applicable for all the raw images. However, we believe that the simplicity and effect of the border padding method still make it beneficial in many situations.

Acknowledgements. This research is financially supported by National Natural Science Foundation of China (No. 61203184).

References

1. He, K., Sun, J., Tang, X.: Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1397–1409 (2013)
2. Liu, R., Jia, J.: Reducing boundary artifacts in image deconvolution. In: 15th IEEE International Conference on Image Processing, San Diego Convention Center, San Diego, CA, USA (2008)
3. Leonard, G., Hamey, C.: A functional approach to border handling in image processing. In: *Digital Image Computing: Techniques and Applications (DICTA)*, Adelaide, Australia (2015)
4. Ragan-Kelley, J., Barnes, C., Adams, A., Paris, S., Durand, F., Amarasinghe, S.: Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines. In: *Proceedings of the 2013 ACM Sigplan Conference on Programming Language Design and Implementation*, Seattle, WA, USA (2013)
5. Li, T., Zhang, X., Li, C.: An improved adaptive image filter for edge and detail information preservation. In: *International Conference on Systems and Informatics*, Yantai, China (2012)
6. Chen, W., Gu, C., Lee, M.C.: Repetitive and morphological padding for object-based video coding. In: *International Conference on Image Processing*, vol. 1, Santa Barbara, CA, USA (1997)
7. Reeves, S.J.: Fast image restoration without boundary artifacts. *IEEE Trans. Image Process.* **14**, 1448–1453 (2005)
8. Ng, P.-E., Ma, K.-K.: A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Trans. Image Process.* **15**, 1506–1516 (2006)
9. Li, S.Z.: *Markov Random Field Modeling in Image Analysis*. Springer, London (2001)