2006 IEEE Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Curvature Domain Image Stitching

Simon T.Y. Suen, Edmund Y. Lam, and Kenneth K.Y. Wong

***Abstract*— Digital photograph stitching blends multiple images to form a single one with a wide field of view. However, artifacts may arise, often due to photometric inconsistency and geometric misalignment among the images. Several existing techniques tackle this problem by methods such as pixel selection or pixel blending, which involve the matching and adjustment of intensity, frequency, and gradient values. However, our experience indicates that these methods have yet fully incorporated the uniformity properties of the photometric inconsistency. In this paper, we first explain the causes of inconsistency and its uniformity property. Then, by mathematical analysis, we show that the matching on the intensity and even the gradient domain is insufficient for some non-uniform inconsistencies. Our method thus adds the extra requirement of an optimal matching of curvature. We then explain how its variables can affect the computational and visual performance. Simulations are carried out using our method, with some masks designed with these two concerns. Some real examples show that our method can produce pleasant visual results even when both misalignment and non-uniform inconsistency exist.**

Keywords: Image stitching, panoramic mosaicing, photometric inconsistency, gradient, curvature

I. INTRODUCTION

Panoramic mosaicing is the process that combines multiple images to form a single one with a wide field of view. Fig. 1 shows its simplest case in which only two input images are involved. This is particularly useful when the images taken have a relatively low pixel count, such as in camera phones or PDA cameras. Image alignment and stitching are the main procedures in this application. Image alignment establishes geometric correspondences among the images. Stitching blends the aligned images seamlessly.

Sometimes, artifacts in stitching may arise. It is often due to photometric inconsistency and geometric misalignment among the images. The former is caused by various physical reasons such as lens vignetting, exposure duration, scene illumination and camera gain [1]. They manifest in differences in pixel intensity values even if they represent the same location in the object. Visible cutting curve is the most common artifact (see Fig. 2). The latter is inherited from the imperfection of the alignment process. Pixel intensity values at the same location may differ when they belong to different objects. Blurred edges and double edges are the most common artifacts in this case (see Fig. 3). These problems make the stitching difficult: we cannot simply take one of these values from the input images, as it would be inconsistent with the others.

Authors are with the Department of Electrical and Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong {tysuen, elam, kywong}@eee.hku.hk.

(a) (b) (c)

Fig. 1. The simplest case of panoramic mosaicing in which only two input images are involved. (a) (b) Input images and (c) Their mosaic image.

(a) (b) (c)

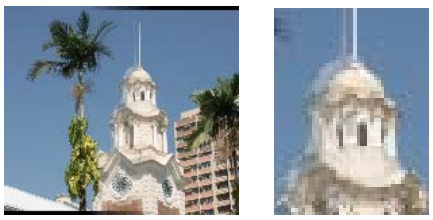
Fig. 2. (a) (b) Two input images with photometric inconsistency and (c) their mosaic image in which a visible cutting curve appears.

II. EXISTING APPROACHES

The existing image stitching methods can be classified into four broad areas: optimal seam approach, smooth transition approach, color tuning approach and optimization approach.

A. Optimal Seam Approach

Optimal seam methods [2], [3], [4], [5], [6], [7] search for the optimal curve in the overlap region where the differences between the two input images are minimal. Each image is then copied to the corresponding side of the curve. Seams on the curve are not noticeable when the differences are close to zero. However, such a seamless curve may not exist when there is photometric inconsistency throughout the area.



(a) (b)

Fig. 3. (a) Stitching result under a misaligned situation and (b) its magnified overlap. Some blurred and double edges appear in the overlap.

Moreover, these methods are less appropriate to the cases with a narrow overlap region, because it may be very difficult to find a potential seamless curve.

B. Smooth Transition Approach

The second approach smoothes the transition between the input images. In feathering [8], each overlapping pixel is computed as the weighted average of the intensity values from the inputs. The weighting coefficients are determined by the distance between the pixel and the corresponding image center. If the inputs are well-aligned, seams become mostly invisible, unless the photometric inconsistency is very severe. However, blurred or double edges often appear when they are misaligned. Laplacian pyramid blending [9] smooths the transition in each frequency band independently. It gives a wider transition zone to the lower frequency bands and a narrower zone to the higher ones. The blurred and double edges appear less noticeably under a misaligned situation. However, the misalignment could result in the mismatches among the frequency bands. Thus, some undesirable brightness variations may appear in the non-overlapping regions of the mosaic image.

C. Color Tuning Approach

The third approach tunes the photometric properties of the input images before stitching. In color correction [10], the photometric inconsistency is first formulated into a linear function by matching the intensity values between the inputs in the overlap region. All pixel intensity values in one of the inputs are then converted by the function in order to match another input before stitching. However, since the formulation of the inconsistency is highly sensitive to the geometric misalignment, a wrong formulation would lead to a wrong correction and thus may give no improvement for stitching.

D. Optimization Approach

The fourth approach computes the mosaic image by an optimization process. Gradient-domain image stitching (GIST) [1] optimizes an energy function defined in terms of pixel gradient values. It aims to obtain a mosaic result whose pixel gradient values are close to that of the inputs. In the process, the creation of new edges are suppressed and the images could blend regardless of their differences in mean intensities. It can stitch seamlessly even when both uniform photometric inconsistency and geometric misalignment exist. However, when non-uniform photometric inconsistency exists, the cutting curve may become noticeable. Moreover, the brightness of the non-overlapping regions are usually modified which is sometimes undesired.

In summary, these approaches tackle the stitching problem by pixel selection or pixel blending, which involve the matching and adjustment of the intensity, frequency, and gradient among the input images. Under the cases with both misalignment and photometric inconsistency, they sometimes fail to meet the two major quality criteria which are (1) the user-desired modification of the image contents in the non-overlapping regions and (2) the absence of noticeable seams in the overlap region.

In Sect. III, we first explain the causes of photometric inconsistency and its uniformity property. In Sect. IV, we give a mathematical analysis to support our claim that curvature can match better than gradient when the inconsistency is non-uniform. In Sects. V and VI, we describe our method which adds the extra requirement of an optimal matching of curvature, in additon to that of gradient and intensity. We also explain how the variables in the method affect the visual and computational performance. In Sect. VII, we test some masks designed with these two concerns. We also demonstrate that our method allows users to determine the degree of modification in the image contents, and stitches seamlessly under the misaligned and photometrically-inconsistent situations.

III. PHOTOMETRIC INCONSISTENCIES

In this section, we describe the causes and the uniformity property of photometric inconsistency. Let $i(x, y)$ be a 2-D image representing the scene radiance. We simulate the image capture geometrically by cropping out $i(x, y)$ in the regions Ω_1 and Ω_2 to get the images $g_1(x, y)$ and $g_2(x, y)$, respectively. The two captured images overlap partially in $\Omega_0 = \Omega_1 \cap \Omega_2$. According to [11], the captured image intensity values relate with the scene radiance as:

$$g_n(x,y) = \left(\frac{\pi t_n}{4} \left(\frac{d_n}{f_n} \right)^2 \cos^4 \left(\theta_n(x,y) \right) \right) i(x,y), \quad \forall (x,y) \in \Omega_n \quad \text{and} \quad n = 1, 2, \qquad (1)$$

where f_n is the focal length of the imaging lens, d_n is the diameter of its aperture, $\theta_n(x, y)$ is the angle subtended by the principal ray from the optical axis, t_n is the exposure duration of the image detectors.

Photometric inconsistency between $g_1(x, y)$ and $g_2(x, y)$ may be caused internally in the capturing system or externally by the change in the scene radiance. For the former, if the exposure duration t_n changes, $g_n(x, y)$ will scale uniformly. This produces a uniform inconsistency. However, when the focal length f_n and the diameter of the aperture d_n change, the corner-darkening effect (i.e. lens vignetting) may appear differently. This can create a non-uniform inconsistency. For the latter, the movement of light sources in the scene environment and the use of flash may likely produce a non-uniform inconsistency. Hence, the non-uniform inconsistency happens more likely than the uniform one.

IV. ANALYSIS IN GRADIENT AND CURVATURE DOMAINS

In order to simplify our analysis, we now discuss in 1-D domain. By spline theory, each image can be approximately represented by a cubic spline which is a joint composite of several pieces of cubic polynomials. We now pick out a certain piece from the scene radiance $i_n(x)$ in the overlap

region Ω_0 for further analysis. We assume a linear polynomial can describe the non-uniform effect, caused during the image capture. Then, the captured piece is described as:

$$g_n(x) = (r_n x + s_n)(c_3 x^3 + c_2 x^2 + c_1 x + c_0), \quad \forall x \in \Omega_0 \quad \text{and} \quad n = 1, 2, \tag{2}$$

where r_n and s_n are the coefficients of the linear polynomial and c denotes the coefficients of the cubic polynomial of that piece. We then look at its gradient and curvature values (i.e. the 1^{st} and 2^{nd} derivatives),

$$g'_n(x) = (r_n x + s_n)(3c_3 x^2 + 2c_2 x + c_1) + r_n(c_3 x^3 + c_2 x^2 + c_1 x + c_0), \tag{3}$$

$$g''_n(x) = (r_n x + s_n)(6c_3 x + 2c_2) + 2r_n(3c_3 x^2 + 2c_2 x + c_1). \tag{4}$$

If we pick $g_1(x)$ and $g_2(x)$ from an approximately flat region such that $c_1 \approx 0$, $c_2 \approx 0$ and $c_3 \approx 0$, then the respective differences between $g_1(x)$ and $g_2(x)$ are

$$g'_1(x) - g'_2(x) \approx (r_1 - r_2)c_0, \tag{5}$$

$$g''_1(x) - g''_2(x) \approx 0. \tag{6}$$

We can notice that the curvature values become zero while the gradient values is still non-zero. To conclude, when the inconsistency is non-uniform, the curvature values between the captured images can match better than the gradient values at a flat region. Thus, in addition to the intensity and gradient values, our method adds an extra requirement of matching the curvature values in order to deal with the non-uniform inconsistency.

V. METHOD

Our method, the *Curvature Domain Image Stitching*, minimizes an energy function defined in terms of pixel intensity, gradient and curvature values.

A. Energy Function

Let $\hat{g}(x, y)$ be the resulting image, $g_1(x, y)$ and $g_2(x, y)$ be the two aligned input images. Let $\alpha(x, y) \in [0, 1]$ be the weighting mask that determines the weighting between g_1 and g_2. Let $\beta_i(x, y) \in [0, 1]$ be an another weighting mask that determines the weighting of the i^{th} derivative among all the considered derivatives i.e. the intensity, gradient and curvature values. Let $\|\cdot\|_p$ be the L_p vector norm. The energy function $E(\hat{g})$ is defined as:

$$\sum_{i=0}^{2}\Bigg(\sum_{(x,y)\in\Omega_1}\alpha(x,y)\beta_i(x,y)\|D^i(\hat{g}(x,y) - g_1(x,y))\|_p^p + \sum_{(x,y)\in\Omega_2}(1-\alpha(x,y))\beta_i(x,y)\|D^i(\hat{g}(x,y) - g_2(x,y))\|_p^p\Bigg), \tag{7}$$

where Ω_1 and Ω_2 are the capturing regions of g_1 and g_2, respectively. $D^i f$ is the i^{th} derivative operator for the real function f such that $D^0 f = f$, $D^1 f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$ and $D^2 f = [\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}]$, representing the intensity, gradient and curvature values respectively. Next, we are going to discuss the three major variables (i.e. p, $\alpha(x, y)$ and $\beta_i(x, y)$) in the following subsections.

B. L_p-norm

The variable p determines the type of vector norm measuring the distance between the result and one of the input images. The choice of norm determines the optimal solution and thus the mosaic image. Suppose the values of the two input images mismatch each other at a pixel. For L_1-norm, any values bounded by these mismatched values can be the optimal solution. However, for L_2-norm, only the weighted average of the mismatched values is the optimal solution while the weighting is determined by the weighting mask $\alpha(x, y)$. In [12], we have shown that their results are similar under a well-aligned situation. But, when the inputs are misaligned, the values mismatch at some pixels. Using L_1-norm gives no smearing artifacts around these pixels, but they appear severely when using L_2-norm.

C. Weighting Mask $\alpha(x, y)$

The weighting mask $\alpha(x, y)$ determines the weighting between the inputs. As defined in Eq. 7, if $\alpha(x, y) = 1$, the latter term in the big bracket becomes zero. The result $\hat{g}(x, y)$ is affected by the derivatives in $g_1(x, y)$ only. If $\alpha(x, y) = 0$, the former term becomes zero. Then the result is affected by the values in $g_2(x, y)$ only. In [12], we have tested three types of masks, namely the uniform mask, the feathered mask and the optimal-cut mask. The uniform mask assigns a constant $\mu \in [0, 1]$, say 0.5, to the overlap region while the feathered mask assigns a gradual transition from 1 to 0 to that region. The optimal-cut mask assigns a step from 1 to 0 when crossing an optimal cutting curve searched by the optimal seam methods. Simulation has verified that only the last type does not produce artifacts such as blurred or double edges under a misaligned situation.

D. Weighting Mask $\beta_i(x, y)$

The use of L_1-norm and the optimal-cut mask prevents noticeable artifacts under a misaligned situation. The weighting mask $\beta_i(x, y)$ determines the weighting of the i^{th} derivative. Its design determines the reduction of artifacts due to the existence of photometric inconsistency and the degree of modification allowed in the image contents. For the former, Sect. IV explains that when photometric inconsistency is non-uniform, the curvature values can match better than the other values. In [12], we have tested some masks emphasizing the use of curvature values in the overlap region. Among them, the uniform and feathered masks are the ones with several derivative values at a pixel. However, their mosaic images give some abrupt edges because the values cannot satisfy with each other. On the other hand, the sharp-cut mask divides the whole region of the mosaic image (i.e. $\Omega_1 \cup \Omega_2$) into several non-overlapping portions and assigns each of them with a single type of derivative. The mosaic image is satisfactory and its computational load is less than the first

two. Here, we explore how to design its variations, in order to increase its computational performance, and to control the degree of modifications in the image contents.

1) Concern on Computational Performance: For each iteration in the optimization scheme, the derivative values are calculated to get the final value of the energy function. For example, when calculating a pixel curvature value, it requires three multiplications and two additions. But fewer operations are required for the other two derivatives. Thus, the ratios among the three derivatives can strongly affect the computational performance. In order to reduce the computational load, we should reduce the use of higher order derivatives.

2) Mask Design based on Modification of Contents: The degree of freedom on modifying the image contents is determined by the spatial assignment of the derivative values. In the non-overlapping regions, if we assign full weighting to the intensity values, the contents can stay nearly unchanged. However, this may sometimes result in a noticeable difference in the mean intensities between the inputs. Giving all weighting to the gradient values makes the problem less severe, as their mean intensities can be shifted to an optimized point. Giving all weighting to the curvature values allows the gradual transition of their brightnesses. However, it requires heavier computational effort. A user can choose to trade-off between the computational performance and the visual results. In Sect. VII-A, we demonstrate some visual and computational results of the above choices.

3) Mask Design based on Photometric Inconsistency: It is good to use curvature values only on the regions with non-uniform inconsistency, as they require heavier computational effort. Thus, in order to extract those regions, we propose to measure the uniformity of the inconsistency in an interested region by calculating its variance. Since the property may vary in its sub-regions, we calculate the variance for each pixel with a small window size. This is sensitive to the changes in uniformity, however, it may not be robust to noise if it is too small. We use a reasonable size for the simulation in Sect. VII-B. If the variance is below a certain threshold (denoted as σ^2_{th}), we classify the location to be uniformly inconsistent and assign to use gradient values around it. Otherwise we assign to use curvature values. In Sect. VII-B, we test some different values for the threshold σ^2_{th} and see how the visual and computational results are affected.

VI. IMPLEMENTATION

We have implemented the optimization process for the energy function defined in Eq. 7 under L_1 and L_2-norm. The 1^{st}-order derivative operator we use is a forward differencing derivative filter $[-1 \quad 1]$ while the 2^{nd}-order one is thus $[1 \quad -2 \quad 1]$. With reference to [1] , for the L_1-case, we can optimize the function shown below by linear programming with the `linprog` command in the Matlab Optimization Toolbox [13]:

$$\begin{aligned} & \min_x \textstyle\sum_i (z_i^+ + z_i^-) \\ \text{subject to} \quad & Ax + (z^+ - z^-) = b, \\ & 0 \le x \le 255, \\ & z^+ \ge 0, \quad z^- \ge 0. \end{aligned} \tag{8}$$

where A is the sparse matrix containing the weighted derivative operators. b is the vector containing the known weighted derivatives of the input images. x is the vectorization of the resulting image. For the L_2-case, using the same A and b, we solve the constrained linear least squares problem defined below by using the `lsqlin` command in the Toolbox:

$$\min_x \|Ax - b\|_2^2 \qquad \text{subject to} \quad 0 \le x \le 255. \tag{9}$$

(a) (b) (c)

Fig. 4. (a) Non-uniform scaling mask for producing inconsistency, where the scaling values are between 0.3 (black) and 1 (white). (b) (c) Synthetic input images with non-uniform inconsistency.

VII. SIMULATION

In the first two simulations, we crop two images (each with 152×114 pixels) from a bigger image in two partially overlapping regions. We then apply a non-uniform scaling mask, shown in Fig. 4 (a), to the left cropped image to simulate non-uniform photometric inconsistency between them. Fig. 4 (b & c) show the two synthetic input images. Moreover, L_1-norm and the optimal-cut mask with a central vertical cutting line for $\alpha(x, y)$ are used in their energy functions.

A. Simulation on Modified Mask Design I

Here, we test some variations of the sharp-cut mask for $\beta_i(x, y)$ to illustrate how the assignment of the derivative values determines the degree of modification in the image contents. Fig. 5 shows the masks used, their resulting image and their average running time for each iteration for each color channel (denoted as t_r). For the masks, vertical, horizontal and diagonal lines mean giving full weighting to intensity, gradient and curvature values in that region, respectively. In Fig. 5 (a & b), the use of intensity values dominates in the non-overlapping regions and thus the contents remain unchanged. However, an abrupt edge appears in the transition region. In Fig. 5 (c & d), the use of gradient values dominates while that of intensity values is abandoned. The brightness of the non-overlapping regions appears to be tuned to match each other. The edge becomes less severe. In Fig. 5 (e & f), the use of curvature dominates while the others are squeezed at the boundaries. The transition between

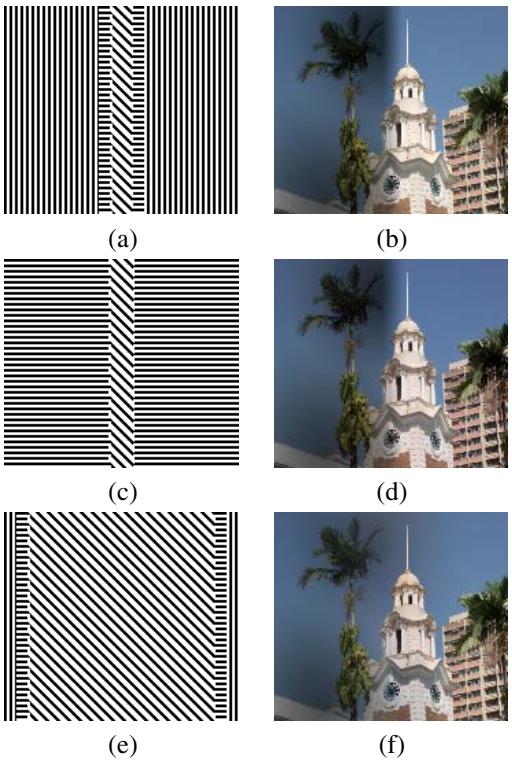


Fig. 5. (a) (c) (e) The variations of the sharp-cut mask for $\beta_i(x, y)$. Vertical, horizontal and diagonal lines mean giving full weighting to intensity, gradient and curvature values in that region, respectively. (b) (d) (f) Their respective visual results. Their computational results are $t_r = 1.664$ sec, $t_r = 3.457$ sec and $t_r = 6.820$ sec respectively, where t_r is the average running time for each iteration for each color channel.

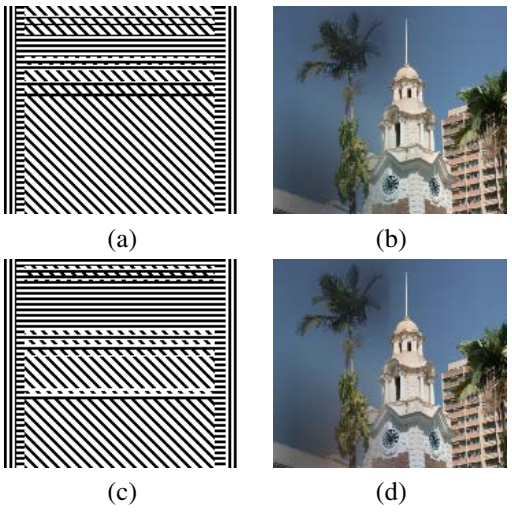


Fig. 6. (a) (b) Mask designed when $\sigma_{\text{th}}^2 = 1.9$ and its mosaic image. When compared with Fig. 5 (e & f), t_r drops from 6.820 to 4.671 sec. (c) (d) Mask designed when $\sigma_{\text{th}}^2 = 3.8$ and its mosaic image. Although t_r becomes 4.187 sec, seams appear at the middle top of the central cutting line.

the inputs becomes more gradual when compared with the others. In this simulation, the computational performance of the three masks are $t_r = 1.664$ sec, $t_r = 3.457$ sec and $t_r = 6.820$ sec, respectively. Thus, the use of higher order derivatives requires heavier computational efforts, but produces some more pleasant visual results. The design of the mask for $\beta_i(x, y)$ is based on this two factors as chosen by the user.

B. Simulation on Modified Mask Design II

Here, we first describe how to measure the uniformity of the inconsistency and then describe how to modify the mask for $\beta_i(x, y)$. As we use the optimal-cut mask for $\alpha(x, y)$, the inconsistency exists only on its cutting curve. To quantify the inconsistency, we form a vector with the intensity differences between the inputs on the curve. We then calculate the variance of the vector for each pixel within a window. We choose 5×1 as our window size because it is robust to noise and sensitive to the changes in uniformity. If the variance is below a certain threshold (denoted as σ_{th}^2), we classify the location to be uniformly inconsistent and replace the row of curvature values with gradient values, in order to improve the computational performance.

We start with the mask shown in Fig. 5 (e). When σ_{th}^2 is set to 1.9, the mask becomes Fig. 6 (a). t_r drops significantly from 6.820 to 4.671 sec while there is no noticeable difference between their visual results, as shown in Fig. 5 (f) and Fig. 6 (b). However, when we double σ_{th}^2 to 3.8, the mask becomes Fig. 6 (c). Seams are visible on the middle top of the cutting line in its result shown in Fig. 6 (d). But, t_r drops only from 4.671 to 4.187 sec. This suggests that there is a critical point for σ_{th}^2. Before it, the computational improvement is more sensitive than the visual degradation, and vice versa. Moreover, we observe that uniform inconsistency is never detected on the lower half of the mask. This is probably because the object contents there containhigh variances. Thus, the tool is less useful when more objects are on the cutting line.

C. Applications to Real Photometric Inconsistencies

Here we demonstrate that our curvature stitching method can produce pleasant visual results on some real examples with both non-uniform photometric inconsistency and misalignment.

First, we demonstrate an example with inconsistency caused by the change of illumination direction. Fig. 7 (a & b) show the input images taken by toggling the lights on the ceiling. We stitch them under a misaligned situation. Fig. 7 (c) shows the result of gradient stitching. Although there is no double edge, noticeable cutting curve appears on the wall, the partition and the drawers. For our method, we use the mask designed by a personal choice for the variable $\beta_i(x, y)$ (see Fig. 7 (d)). Intensity values are dominated in the non-overlapping regions, thus the brightness on the regions will keep unchanged. Fig. 7 (e) shows that the seams disappear in the result of our method.

Secondly, we demonstrate an example with inconsistency caused by the switch of flash. Fig. 8 (a & b) show the input images taken without and with flash, respectively. Again, we stitch under a misaligned situation. Fig. 8 (c) shows the result of gradient stitching. A noticeable cutting curve appears on

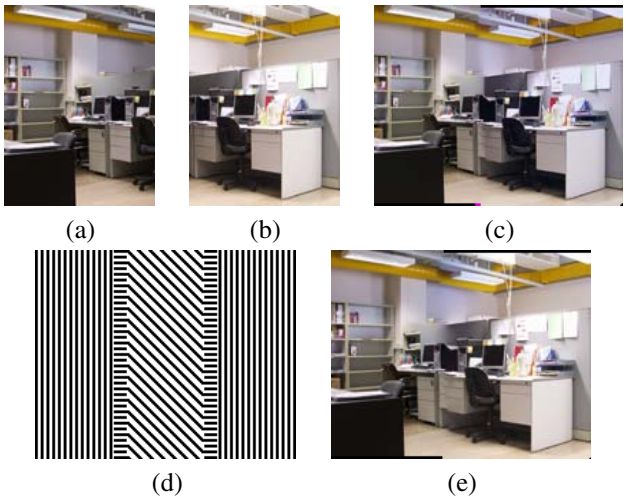


Fig. 7. (a) (b) Input images taken by toggling the lights on the ceiling. (c) Result of gradient stitching. Noticeable cutting curve appears on the wall, the partition and the drawers. (d) Mask designed for the variable $\beta_i(x, y)$ in our method. (e) Result of our method. The seams disappear.

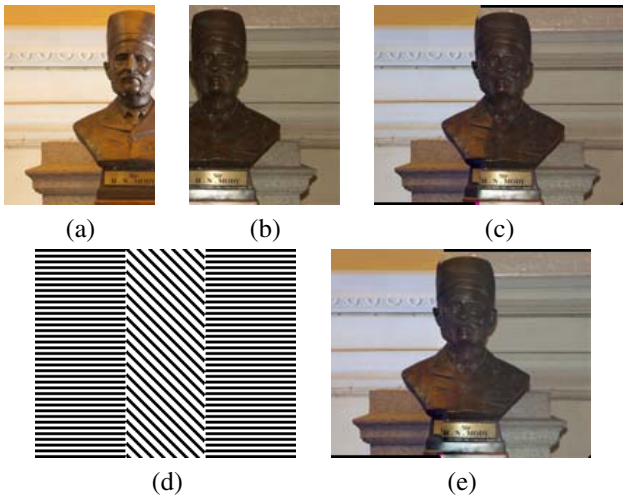


Fig. 8. (a) (b) Input images taken without and with flash, respectively. (c) Result of gradient stitching. Noticeable cutting curve appears on the body of the statue. (d) Mask designed for the variable $\beta_i(x, y)$ in our method. (e) Result of our method. The curve is not noticeable at all.

the body of the statue. For our method, we use the mask shown in Fig. 8 (d) for the variable $\beta_i(x, y)$. Gradient values are dominated in the non-overlapping regions, thus mean intensity offset is allowed. Fig. 8 (e) shows that the curve is not noticeable at all in the result of our method.

Thus, our method has the ability of not producing seams such as double edges under misalignment and a noticeable cutting curve under non-uniform inconsistency, and allowing the user to decide the degree of modification of the image contents.

VIII. CONCLUSION

We introduce the uniformity property of photometric inconsistency and analyze mathematically to claim that the curvature values can match better than the gradient values when the inconsistency is non-uniform. We hence design our method by extending the gradient stitching method. We define a new energy function which involves the pixel intensity, gradient and curvature values. In this paper, we focus on testing some variations for the variable $\beta_i(x, y)$. We suggest how to improve the computational performance and control the degree of modification in the image contents. We also test on a tool using variance to estimate the uniformity of the inconsistency. It can provide a helpful guide for designing the mask with some computational improvement. Some real examples successfully show that our method can provide seamless results under both misalignment and non-uniform inconsistency. In our future work, we will find a better tool on estimating the uniformity without being affected by the object contents. Moreover, we will research on defining a new energy function involving variances.

REFERENCES

[1] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss, "Seamless image stitching in the gradient domain," in *8th European Conference on Computer Vision*, May 2004, vol. 4, pp. 377–389.

[2] David L. Milgram, "Computer methods for creating photomosaics," *IEEE Transactions on Computers*, vol. 24, no. 11, pp. 1113–1119, November 1975.

[3] Shmuel Peleg, "Elimination of seams for photomosaics," in *Computer Graphics and Image Processing*, August 1981, vol. 16, pp. 90–94.

[4] Shiren Yang, Li Li, and Gao Peng, "Two dimensional seam-point searching in digital image mosaicking," in *Photogrammetric Engineering and Remote Sensing*, January 1989, vol. 55, pp. 49–53.

[5] Susanne Hummer-Miller, "A digital mosaicking algorithm allowing for an irregular join line," in *Photogrammetric Engineering and Remote Sensing*, January 1989, vol. 55, pp. 43–47.

[6] Francisco P.A. Junior and Neucimar J. Leite, "A morphological algorithm for photomosaicking," in *8th European Signal Processing Conference*, September 1996, vol. 3, pp. 1881–1884.

[7] Alexei A. Efros and William T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, August 2001, pp. 341–346.

[8] Matthew Uyttendaele, Ashley Eden, and Richard Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," in *Computer Vision and Pattern Recognition*, December 2001, vol. 2, pp. 509–516.

[9] Peter J. Burt and Edward H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics*, vol. 2, no. 4, pp. 217–236, October 1983.

[10] Gui Yun Tian, Duke Gledhill, David Taylor, and David Clarke, "Colour correction for panoramic imaging," in *6th International Conference on Information Visualisation*, July 2002, pp. 483–488.

[11] Tomoo Mitsunaga and Shree K. Nayar, "Radiometric self calibration.," in *Computer Vision and Pattern Recognition*, June 1999, vol. 1, pp. 1374–1380.

[12] Simon T.Y. Suen, Edmund Y. Lam, and Kenneth K.Y. Wong, "Digital photograph stitching with optimized matching of gradient and curvature," in *Proceedings of the SPIE*, January 2006, vol. 6069, pp. 139–154.

[13] *MATLAB Optimization Toolbox User's Guide*, The MathWorks, Inc., 2005.