J. KSIAM Vol.22, No.2, 91–100, 2018 http://dx.doi.org/10.12941/jksiam.2018.22.091

ENHANCED EXEMPLAR BASED INPAINTING USING PATCH RATIO

SANGYEON KIM[1], NAMSIK MOON[2], AND MYUNGJOO KANG[3,†]

[1]INTERDISCIPLINARY PROGRAM IN COMPUTATIONAL SCIENCE AND TECHNOLOGY, SEOUL NATIONAL UNIVERSITY, KOREA
E-mail address: kimssang@snu.ac.kr

[2]DEAPRTMENT OF MATHEMATICS, AJOU UNIVERSITY, KOREA
E-mail address: nsmoon@ajou.ac.kr

[3]DEAPRTMENT OF MATHEMATICAL SCIENCES, SEOUL NATIONAL UNIVERSITY, KOREA
E-mail address: mkang@snu.ac.kr

ABSTRACT. In this paper, we propose a new method for template matching, patch ratio, to inpaint unknown pixels. Before this paper, many inpainting methods used sum of squared differences(SSD) or sum of absolute differences(SAD) to calculate distance between patches and it was very useful for closest patches for the template that we want to fill in. However, those methods don't consider about geometric similarity and that causes unnatural inpainting results for human visuality. Patch ratio can cover the geometric problem and moreover computational cost is less than using SSD or SAD. It is guaranteed about finding the most similar patches by Cauchy-Schwarz inequality. For ignoring unnecessary process, we compare only selected candidates by priority calculations. Exeperimental results show that the proposed algorithm is more efficent than Criminisi's one.

1. INTRODUCTION

Inpainting is an interesting subject in computer vision area. It can be apply to many ways like restoring damage parts of image, image compression, red eye correction and object removal in digital photos. Let us focus on object removal in digital photos. After taking photos, we often have an exprience that hope to remove unwanted parts or objects like garbage. After we remove them, the result must be unawkward for human visuality which is the most important aspect of image inpainting. Image inpainting was first introduced by Bertalmio et al. [1], it is used to filling in missing data in a target region of a single or video image. In recent work, we can specify two classes algorithms to restore a digital image from big missing region. "Texture synthesis" [2] algorithm produces a new image from an initial source image by non-parametric sampling. "Inpainting" techniques based on solving PDE thought isophote information should extend to missing region. However, this method actually uses the diffusion techniques and it

Received by the editors April 10 2018; Accepted April 27 2018; Published online June 18 2018.
2010 *Mathematics Subject Classification.* 65K05.
Key words and phrases. Image inpainting, texture synthesis, template matching, patch ratio.
[3,†] Corresponding author.

can cause blurring problem and more time cost. "Texture synthesis" generally select the best candidate from known pixels that avoids blurring. But the result image isn't smooth so it can be seen as unnatural. A novel effective algorithm was proposed by Criminisi et al. [3]. It gives higher priority to the pixel which have more confidence and information and inpainting first.

However, there are some deficiencies in Criminisi algorithm. First, Cheng et al. [3] discovered that the confidence term that was defined in Criminisi's algorithm decreases exponentially and thus the multiplicative definition of the priority term needs to be replaced. Second, bad selection can be occured from defining distance in texture synthesis. The most common method for calculating distance between two matrices are SSD and SAD. For solving those problems, in section 4, we suggest new priority calculation method and template matching using patch ratio after show a simple error example by Criminisi's method. We take a idea about Cauchy-Schwarz inequality. Also for accuracy, we will mention about energy terms suggested by Bugeau, Bertalmio, Caselles and Saprio [4]. However, it takes much time so we just calculate sum of gradients when unknown parts of template are replaced with best candidates, then choose minimize one. In section 5, we will show our results compared with other methods.

2. Inpainting Algorithm of Criminisi et al.

2.1. **Motivaltion of the algorithm.** Criminisi's algorithm considered where to start inpainting. Unknown Pixels surrounding with other unknown pixels have no confidence for inpainting and unknown pixels nearby edge or corner can be regarded as much information than others.

After priority calculation for boundary of missing regions, template matching proceed for inpaint the pixel and do those processes again.

2.2. **Priority Calculation.** Template based inpainting algorithms are filling in blanks using known pixels in image. Especially, unknown pixels are related to surrounding pixels of them, geometrically and brightness values. It should give higher priority for an unknown pixel if there are more known pixels and much geometric information around it. So priority term consists with confidence term and data term.

Take patches which take a center as each unknown pixel. Confidence term is counting number of known pixels in a patch, and data term calculates intensity of isophote reaching unknown pixel of unfilled region in the filling leading edge.

2.3. **Texture Synthesis.** After calculating priority, they use texture synthesis method to fill in unknown pixels succesively as higher priority. Denote square template which is centered with unknown pixel and candidates which is same size block with template which is centered with known pixels.

Among candidates, they calculate distance with the template using SSD. Take best candidates which are close to template and choose one randomly. Finally, unknown pixel value is replaced with a center pixel value of the best candidate.

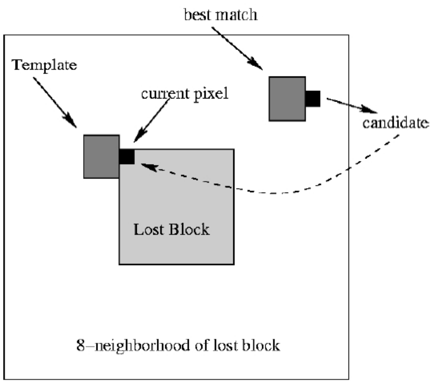

FIGURE 1. How to do texture synthesis.

3. CRIMINISI ALGORITHM

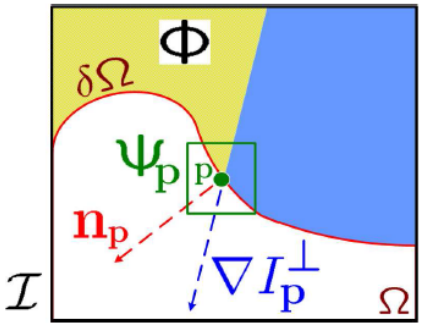

FIGURE 2. In image I, Ω marks the inpainting region, $\delta\Omega$ the boundary of it and the Φ the source region. The point p is an unknown pixel want to be inpainted and Ψ_p centered around p is the patch that regarded as template. $\nabla I_p^\perp$ presents the edge strength and direction at p and n_p is a unit vector orthogonal to $\delta\Omega$ at p.

Step 1: The patch-priority of a patch centered around a $p\in \delta\Omega$ consists of two terms, the confidence term $C(p)$ and the data term $D(p)$. The product of two terms means the priority of p :

$$P(p) = C(p)D(p)$$

The two terms $C(p)$ and $D(p)$ were calculated as :

$$C(p) = \frac{\sum_{q\in\Psi_p\cap(I-\Omega)} C(q)}{|\Psi_p|}, D(p) = \frac{|\nabla I_p^\perp n_p|}{\alpha} \tag{3.1}$$

$|\Psi_p|$ represents the area of Ψ_p, n_p is a unit vector orthogonal to $\delta\Omega$ in p, α a normalization factor (α=1 for double format) . and $\nabla_p^\perp$ representing edge strength and direction at p. The numerator of $C(p)$ implies counting known (or inpainted) pixels in Ψ_p so it rates how sure we can be about a pixel $p \in \delta\Omega$. The data term $D(p)$ envolves the isophote direction and strength at p. Since filling order should be consider linear structures in the image, we multiply $\nabla^\perp$ which is a strength of edge and n_p which is a direction of isophote. Therefore, if a pixel p located near to a linear structure then $D(p)$ is higher than other pixels'.

Step 2: After step one is complete, we get the priority order for p in $\delta\Omega$. Then use Efros and Leung's texture synthesis method [2] to find matching blocks in image. Let the pixel $\hat{p} \in \delta\Omega$ has the highest priority $P(\hat{p}) = \max_{p\in\delta\Omega} P(p)$. Step 2 is finding the most similar patch $\Psi_{\hat{q}}$ with $\hat{q} \in \Psi$ through a comparative simple task. $\Psi_{\hat{q}}$ is the patch with the smallest distance to $\Psi_{\hat{p}}$.

$$\Psi_{\hat{q}} = \min_{\Psi_q \in \Phi} d(\Psi_{\hat{p}}, \Psi_q)$$

d($\cdot$, $\cdot$) is a simple distance function like the sum of squared differences(SSD). After finding $\Psi_{\hat{q}}$ the texture and color values of $\Psi_{\hat{q}}$ are copied to $\Psi_{\hat{p}}$ for all $p \in \Psi_{\hat{p}} \cap \Omega$.

Step 3: Finally the priorities have to be updated.

$$C(p) = C(\hat{p}), \forall p \in \Psi_{\hat{p}} \cap \Omega$$

The confidence term decays after these loops, so we are less sure about the values of pixel near the center of the shrinking target region.

4. Our Algorithm

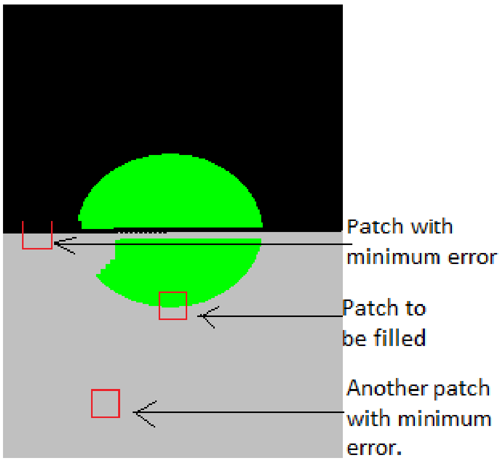

FIGURE 3. Patches with same SSD. Choosing incorrect patch may occur poor results.

4.1. **Calcuate priorities of exemplar blocks.** According to Cheng et al. [3], the confidence term that was defined in Criminisi's algorithm decreases exponentially and thus the multiplicative definition of the priority term needs to be replaced. However, changing it to additive form of priority cause confidence term does not match the order of the data term. Thus the confidence term should be modified with the regularized confidence term. The authors also proposed the weights to different components in the definition of priority term so that a balance between confidence and data term could be maintained. Therefore the modified priority term can now be represented as

$$P(p) = \alpha \times R_c(p) + \beta \times D(p), 0 \leq \alpha, \beta \leq 1, \cdots (1)$$

where α and β are respectively the component weights for the confidence and data terms. Also $\alpha + \beta = 1$ and $R_c(p)$ is the regularized confidence term

$$R_c(p) = (1 - w) \times C(p) + w, 0 \leq w \leq 1, \cdots (2)$$

where w is regularizing factor for controlling the curve smoothness. In this way the modified priority function will be able to resist the confidence term decreasing exponentially.

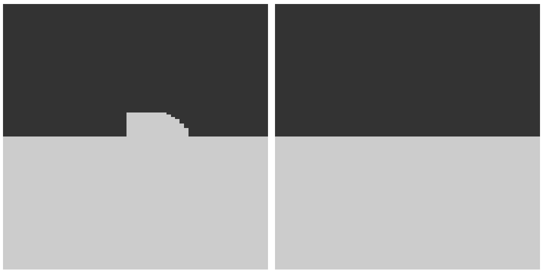

FIGURE 4. (a)Criminisi's algorithm, (b)our algorithm

In our algorithm for calculating priorities, we simply the factors of C(p) and D(p). We can get following $P(p)$ as insert equation (2) to (1).

$$P(p) = \alpha(1 - w) \cdot C(p) + \alpha \cdot w + \beta \cdot D(p)$$

We don't need constant term so we don't have to calculate $\alpha \cdot w$ and we only need to know ratio between $\alpha(1 - w)$ and β. So we use the following equation to get priority terms $P(p)$. $(\alpha = 0.25, \beta = 0.75, w = 0.7)$

$$P(p) = C(p) + 10 \cdot D(p)$$

Still it remains a problem, that if we compute priority for every pixel on boundary, we can't make sure about quality. Even if few pixels around p we know, those algorithms do inpainting and it can occur unnaturallity. So we only calculate priority when confidence term $C(p)$ is larger than threshold and do inpainting among these pixels. This priority algorithm is faster and more accurate for inpainting. (Notice that not only when we use texture synthesis for inpainting, but also this method is necessary for any inpainting methods like PDE based inpainting.)

4.2. **Patch Ratio for Choosing Best Matching Block.** In Criminisi, the mathematical model of selecting the best-matching block is calculating Euclidean distance of matrix between target block and matching block, precisely, $\Psi_{\hat{q}} = \min_{\Psi_q \in \Phi}\ d(\Psi_{\hat{p}}, \Psi_q)$. Euclidean Distance of matrix is a valid method for calculating matrix similarity, especially for the case that most points are unknown in the original matrix. It can be perfectly accepted in the respect of human visuality and image inpainting law by calculating the Euclidean distance between known points in the original matrix and corresponding points in the matching block. However, when the number of known points in original matrix exceeds a certain threshold, precisely, the structure of matrix is determined, it cannot satisfy requirements of human visual psychology in most cases by using Euclidean Distance judging matrix similarity, such as visual discontinuities and image extending from high texture to low in the boundary. See figure 5 as an example. For the continuity

1	2	0
2	3	0
3	4	5

(a)

1	2	3
1	2	4
1	2	3

(b)

3	4	6
3	4	6
3	4	5

(c)

FIGURE 5. Example for Euclidean Distance, (a) 0 is unknown pixel, (b) Candidate1, (c) Candidate2

of image, the gradient of the point in direction x and y cannot be too large, precisely, difference between the point and surrounding should be small. In Figure 5, (a) is a square matrix with unfilled blocks which is numbered with 0 and (b), (c) are both fulfilled matrices. (We will denote as matrix(a), matrix(b), and matrix(c)) According the algorithm of calculating Euclidean distance in Criminisi, we obtain Euclidean distance of matrix is 14 between matrix(a) and matrix(b). Euclidean distance of matrix is 10 between matrix(a) and matrix(c). By Criminisi's method, it should select matrix(c) as the best matching block, and inpainting result is shown in Figure 6.(a) while Figure 6.(b) is the result of selecting matrix(b) as the best-matching block. We can easily find out in Fig 6, that (a) is significantly less smooth than (b) with human visuality.

Therefore, for our algorithm, we use patch ratio to compare patches. See the following Cauchy-Schwarz inequality for vectors $x = (x_1, x_2, \cdots x_n)$ and $y = (y_1, y_2, \cdots y_n)$ is

$$< x, y > \leq \|x\| \cdot \|y\|$$

where $< x, y >= \sum x_i y_i$ and norm of inner product is $\|A\| =< A, A >^{1/2}$. The equality holds if and only if x and y are linearly dependent (In other words, $\dfrac{x_1}{y_1} = \dfrac{x_2}{y_2} = \cdots = \dfrac{x_n}{y_n}$). So if $\dfrac{< x, y >}{\|x\| \cdot \|y\|}$ closer to 1, it means x and y are more similar vectors. Idea from this property, we

1	2	6
2	3	6
3	4	5

(a)

1	2	3
2	3	4
3	4	5

(b)

FIGURE 6. Comparison results when using Euclidean Distance

calculate ratio between template patch A and candidate patch B.

$$ratio(A, B) = \sum \left|\frac{b_{ij} + 1}{a_{ij} + 1} - 1\right|$$

where a_{ij}, b_{ij} are elements in patch A and B, respectively. If candidate B has similar geometric structure with template patch A, $ratio(A, B)$ is close to 0. Finally, we take the best candidates.

$$\Psi_{\hat{q}} = \min_{\Psi_q \in \Phi} \; ratio(\Psi_{\hat{p}}, \Psi_q)$$

We also can write $ratio(A, B)$ as

$$ratio(A, B) = \sum \frac{|b_{ij} - a_{ij}|}{a_{ij} + 1}$$

and note that numerator is same with sum of absolute values. We can think denomiator part as an weight and it is fixed when we choose the template while candidates are changed. It means we consider both distance and geomtrically similarity. When we select candidates to compare with the template , we have to select trustful candidates and that is how much information is in candidate block. Therefore we choose candidates to compare with the template, only if nonzero elements in candidate are more than threshold.

However, it is not still enough to make algorithm fast and exact. So we think about human visual psychology when looking holes in image that want to fill in. Actually, when we look a part of an image and think about its naturality, we only compare the part and surround part of it to decide smoothness. It means human visuality doesn't bother by a relation between the part and another parts which are far from it. So we use "window" concept that take a window around the template. After we decide the template, we only compare patches in window so it fits with human visuality.

After doing these processes, we find out ratio is not clear method since ratio doesn't consider the gradients in image. It just consider how candidates are close to the template. We have to take process for smoothness of image and we take an idea from Rudin, Osher, and Fatemi [5] model. It is one of the simplest form that decides smoothness of image. We choose the smoothest candidates. In case of more than one patch satisfy these conditions, we randomly choose one.

$$\Psi_{best} = argmin_{candidates} \sum |\nabla \Psi_{template+candidates}|^2$$

5. RESULTS

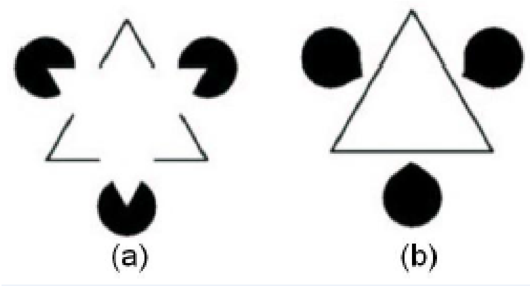

FIGURE 7. (a)"Kanizsa triangle" (b)Our algorithm

In our algorithm, patch size (or block size) is related to time cost and accuracy. If we have a small size patch for complex areas, it might return a poor result. If we have a large size patch, then it takes more time for calculating ratio and also we have to enlarge window size for accuracy. We compare 5, 7, 9, and 11 for patch width (and also height) then the best result occurs when our algorithm uses 11×11 patch.

For calculating ratio, we choose candidates that at least 90% of elements is known and after this process, we choose best candidates smaller than $\min(ratio) \times 1.3$. Finally, we calculate total variation energy for each case with best candidates, and choose minimize one then inpaint the pixel.

"Kanisza triangle" is the interesting picture for image inpainting. It is related with "connectivity principle". The result should be more likely to be two intersected linear lines according to the exemplar-based inpainting algorithm. To analyze further, we find that the imperfections arose from the filling disorder such that redundant pixels are added to change the curvature. On the contrary, in the proposed approach, the filling orders are correctly computed and most of the reconstructed circles are next to perfect.

Now we compare costing time between Criminisi algorithm and our algorithm.

Image	Size	Unknown Pixel	Method	Time
Elephant	225x339	9759 pixels	Criminisi	6649s
			Ours	544s
Bungee	308x206	7996 pixels	Criminisi	4382s
			Ours	598s
Giraffe	210x321	10015 pixels	Criminisi	7223s
			Ours	759s

FIGURE 8. Comprarisons with others: a) Input image, b) Mask Image, c) Bornemann's [6], d) Criminisi's [1], e) Ours

Our algorithm is almost 10 times faster than original Criminisi.

6. CONCLUSION

In this paper, we find out some deficiencies in Criminisi algorithm, and suggest new algorithm ,using patch ratio, which improves speed and nuturallity of inpainted image. We think the most important thing for inpainting is speed. Therefore we use 'window' for speed up based on human visuality property. Also for accuracy, we give some thresholds in algorithm, and this idea is good not only for accuracy, but also for speed up. Finally, we use a new matching strategy, "patch ratio", based on Cauchy-Schwarz inequality. It is simple and fast as computing sum of squared differences, and moreover it is better to compare patches in geometrical sturcture. We have plan to optimize this process using parallel computing skills and find better process for speed up and looking natural.

ACKNOWLEDGMENTS

Myungjoo Kang's work was supported in part by NRF(2017R1A2A1A17069644, 2015R1A 5A1009350) and IITP-MSIT(B0717-16-0107).

REFERENCES

[1] A. Criminisi, P. Perez, and K. Toyama *"Region filling and object removal by exemplar-based image inpainting", IEEE Trans. Image Processing, vol. 13, no.9, pp.1200-1212, Sept. 2004*

[2] Alexei A. Efros and Thomas K. Leung. *"Texture Synthesis by Non-parametric Sampling", IEEE International Conference on Computer Vision, Corfu, Greece, Sept. 1999*

[3] Wen-Huang Cheng, Chun-Wei Hsieh, Sheng-Kai Lin, Chia-Wei Wang, and Ja-Ling Wu. *"Robust Algorithm for Exemplar-Based Image Inpainting", The International Conference on Computer Graphics, Imaging and Vision (CGIV 2005), Beijing, China, 64-69, 2005*

[4] Aurelie Bugeau, Marcelo Bertalmio, Vicent Caselles, Member, IEEE, and Guillermo Sapiro *"A Comprehensive Framework for Image Inpainting", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL.19, NO.10, 2634-2645, OCT. 2010*

[5] Rudin, L. I. , Osher, S. and Fatemi, E. *"Nonlinear total variation based noise removal algorithms". Physica D, 1992*

[6] F. Bornemann and T. März. *"Fast image inpainting based on coherence transport". Journal of Mathematical Imaging and Vision, 2007*