INPAINTING TECHNIQUES - A SURVEY

C K Syama[1], Maya Mohan[2], and Sruthy Manmadhan[3]
[1]M.Tech First Year Student, [2,3]Assistant Professor
Department of Computer Science and Engineering,
N.S.S College of Engineering, Palakkad
Email: [1]cksyama@gmail.com, [2]mayajeevan@gmail.com, [3]sruthym.88@gmail.com

***Abstract*—Inpainting is the process of reconstructing lost parts of images and videos. Inpainting finds numerous applications like repairing photographs, removal of unwanted objects. Mainly inpainting is classified into two techniques; image inpainting and depth map inpainting. Image inpainting is mainly classified into three categories i.e.Statistical based, Partial differential Equation (PDE) based and Exemplar based. In 3 dimensional computer graphics a depth map is an image that contains information relating to the distance of the surfaces of scene objects from a viewpoint. Inpainting applied on such images are termed as depth map inpainting.**

***Keywords*-Depth image,Partial differential equation**

I. Introduction

Reconstruction of missing or damaged portion of an image is known as inpainting [1]. This activity consists of either filling in missing part of the image or modifying damaged part of the image.inpainting is classified into two techniques; image inpainting and depth map inpainting. In this paper Image inpainting approaches are used. Image inpainting is mainly classified into three categories i.e. Statistical based, Partial differential Equation (PDE) based and Exemplar based inpainting

Depth map inpainting is similar with traditional colorimage inpainting to some extent. Depth map inpainting is done using the depth information of an image. depth map is an information relating to the distance of the surfaces of scene objects from a viewpoint.

Statistical Based Image Inpainting approach is mainly used for texture synthesis purpose through use of compact parametric statistical models. Portilla and Simoncelli [3] use joint statistics of wavelet coefficients. Heeger and Bergen [4] make use of color histogram at multiple resolutions for analysis of texture. Main problem of these methods which are based on parametric statistical models is that, they are applicable only to the problem of texture synthesis and not to the general problem of image completion.

PDE-Based Image Inpainting, try to fill the missing region of an image through a diffusion process, by smoothly propagating information from the boundary towards the interior of the missing region. According to these techniques, the diffusion process is simulated by solving a partial differential equation, which is typically nonlinear and of high order. Bertalmio et al. [5] try to fill a hole in an image by propagating image Laplacians in isophote direction. Recently, Bertalmio et al [6] proposed to decompose an image into two components. First component is representing structure and it is filled by PDE based methods, while second component represents texture and is filled by using texture synthesis method. Chan and Shen [7] use variational model for filling missing part of an image. These methods are mainly used when missing area or filling area is small. These methods introduce some blur when filling area is large.

Exemplar-Based Image Inpainting method, try to fill up the unknown region of an image simply by copying the content from the known part of the image. This approach is either patch based [8] or pixel based inpainting approach, meaning that final texture is synthesized by one pixel or by one patch at a time. Patch based methods achieve results of higher quality and they maintain higher order statistics of input texture. This method can be also used for video. It synthesizes new video texture by arranging recorded frame of the input video. Criminisi et el [1] used the exemplar-based inpainting technique for region filling and object removal purpose. In this, authors use filling order computations and texture synthesis technique.

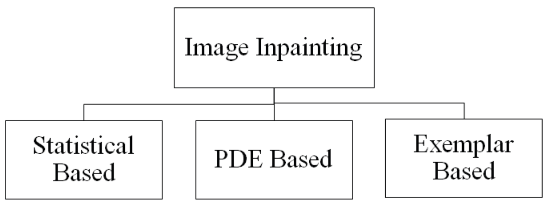

Figure 1. Image inpainting approaches

II. INPAINTING TECHNIQUES

A. Depth image inpainting

Depth map inpainting is similar with traditional color image inpainting to some extent.In color image inpainting domain,there are many well-developed algorithm. Therefore many researchers utilized the color image inpainting techniques to fill holes in depth map

In this method, a depth-assisted edge detection algorithm for depthmap inpainting is proposed.First, edge detection is carried out in both color and depth image. Then the edges extracted from color image are optimized using depth data,while the edges extracted from depth map are also optimized using color data. Finally,the optimized color-edges and depth-edges are fused together

1) Color-edge extraction: Color images captured by Kinect sensor usually have much noise. The smoothing process is always needed before edge detection.Gaussian smoothing technique has been widely used in edge extraction.However it will cause some details lost around object boundaries and affects the accuracy of edge detection. Therefore some edge-preserving smoothing algorithms have been proposed.

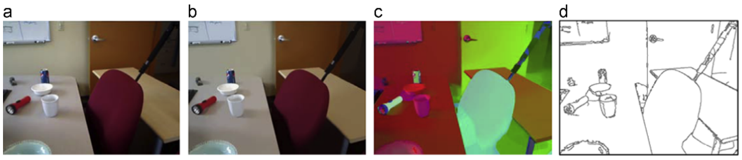

Figure 2. Color image edge extraction.(a)Color image.(b)Filtered by L0 smoothing.(c)Converted to HSV color space.(d)Extracted edges by using Canny operator

Fig. 2(a) shows the color image downloaded from the RGB-D Object Dataset, which is captured by Kinect sensor. Fig. 2(b) shows the smoothed result by using L0 smoothing technique with $\lambda = 0{:}005$ and K = 2:0. We can see that the details on objects surfaces are filtered and the boundaries of objects are preserved. Then the smoothed image is converted from RGB color space into HSV color space, as shown in Fig. 2(c). After the conversion,the back of the chair and the ground can be distinguished correctly. Finally,the Cannyoperator is applied to the HSV image to extract initial edges, as illustrated in Fig. 2(d). The upper edge of the table is failed to be extracted because the table and the wall have similar colors in both RGB image and HSV image. Besides, the characters on the white board are not object boundaries, but they are extracted,too.Therefore the edges extracted from color image need to be optimized using depth information.

2) Depth-edge extraction: In Depth-edge extraction, Canny operator is directly applied to the original depth image.Fig. 3(a) shows the depth image corresponding to the color image in Fig. 2. The extracted edges are shown in Fig. 3(b). According to Fig. 3, it can be seen that the upper edge of the table is extracted correctly. But there are many edges that cannot represent real object boundaries because they are caused by missing regions other than valid depth values. Therefore the results should be optimized using color information

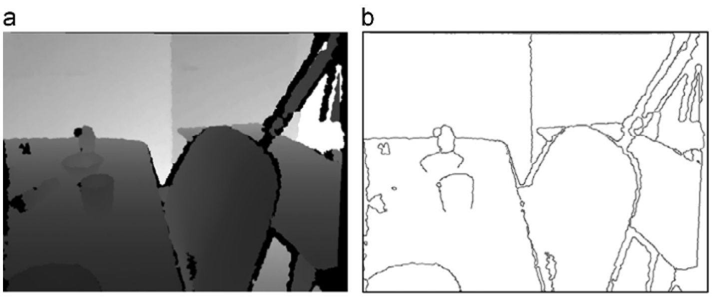

Figure 3. Depth map edge extraction.(a)Depth image.(b) Extracted edges by using Canny operator

3) Color-edge optimization: There are two problems with the edges extracted from the color image:One is that some edges are not extracted because of similar colors, such as the upper edge of the table. And there are some non-boundary edges, such as the characters on the white board and the textures on the can, which are not desired in depth map inpainting. Solution to these problems are as follows: first problem should be solved by combining the edges extracted from depth map.The second problem can be solved by checking depth values around those edges. Therefore the color - edges are optimized using depth data to eliminate non-boundaries firstly

A pixel is called an edge - pixel if it is located on one of the extracted edges. For each edge pixel $e_c = (x,y)^T$ in color image search for its 8 neighbours $e_c = (x^{'},y^{'})^T$ Normal vector of e_c is calculated using $g_c = n[(y-y^{'},x-x^{'})^T]$

If $e_c$belongs to real boundary then the pixels in depth map around e_C must have a significant gradient along the direction of g_c e_c is considered as real boundary edge pixel if the largest gradient of the depth pixels is larger than some threshold.Otherwise it will be deleted. The optimized results can be seen in Fig. 3. The characters on the white board and the textures on the can are removed.In this way, most of the non-boundary edges can be eliminated.

4) Depth-edge optimization: The initial edges extracted from the depthmap can be classified into two categories.The first category contains the edges that are formed by valid pixels with different depth values, which are called real-edges. The other is formed by missing regions, which are called missing -edges. The real-edges can represent object boundaries and should be preserved, while the missing-edges should be optimized using color information. As shown in Fig. 3(a), the pixels in missing regions of the depth map have the value of zero, while the valid pixels have the value larger than zero. To judge whether an edge-pixel belongs to a real- edge or missing-edge, we only need to check whether there are non-valid pixels around it in the depthmap . For each edge-pixel $e_d = (u,v)^T$ in the depth-edge image, we search its 8-neighbors in the depthmap to check whether there are non-valid pixels. If non- valid pixels are found,the edge-pixel e_d is regarded belong to missing-edges. Otherwise it is considered to be a real-edge pixel. All real-edge pixels are preserved, as illustrated in red in Fig. 5.

As illustrated in Fig. 6, giving the missing- edge- pixel pair (e_d, e'_{d}) and the normal vector g_d, the pixels between e_{d1} and e_{d2} in the color image are checked. If the largest gradient at pixel e_l is larger than some threshold, e_l is preserved as the optimized edge-pixel. No matter e_l is preserved or not, both e_d and $e_d^{'}$ are removed from the depth- edge image.

5) Edge fusion: After the optimization of the edges extracted from both color image and depthmap, edges are fused together to get the final detection result. The edge extracted from color image and the edge extracted from depthmap are not totally coincident. In most cases, the color edge is more reliable. Therefore only the lost boundaries in the color -edge image should be complemented according to optimized depth-edges.Our fusion strategy is as follows. For each optimized depth- edge, its direction is calculated firstly. If there is an optimized color- edge near it, which has the same direction, the depth- edge is neglected. Otherwise, the depth-

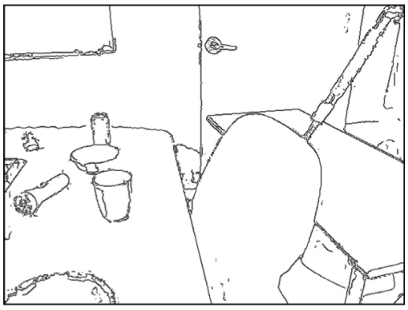

Figure 4. Results of combining optimized color-edges and depth-edges.

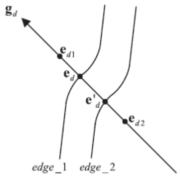

Figure 5. Schematic diagram of searching scope.$edge_1$ and $edge_2$ are two missing- edges extracted from the depthmap. e_d is a pixel on $edge_1$. g_d is the normal vector of $edge_1$ at pixel e_d. $e_d^{'}$ is the corresponding pixel on $edge_2$. Pixels between e_{d1} and e_{d2} will be checked to find the pixel that has the largest gradient

edge is added to the optimized color- edge set. The overall edge fusion result is shown in Fig. 7

6) Depthmap inpainting: Assume Ω is the region where depth information is lost. For each missing pixel $p \in \Omega$, its depth value d(p) can be predicted by valid pixel q as Eqn.(1) where p is the missing pixel, q is valid pixel, w(p,q) is weight function,i.e distance between p and q, and N(p) is set of valid depth pixels near p

$$d(p) = \Sigma_{q(p)} w(p,q)[d(q) + [\langle (q), p-q \rangle] \quad (1)$$

Blue and green areas in Fig. 8 represent two objects. White areas indicate the missing regions, while red curves represent the optimized edges of the object. .The pixel, on one of these rays, is added to N(p) if it is not in region Ω. The selected pixels are marked by little squares. This method is quite efficient, however it does not consider edge information. The set N(p) is determined as follows. Giving the missing pixel p, the pixels on the rays of eight directions are checked, as illustrated in Fig. 8. Depth values of these pixels and the edges extracted from both the color image and the depth map are considered as well. Only the pixels that are on the same side of the curve with p are added to set N(p). The pixels on the opposite side are ignored even though their depth information is valid.

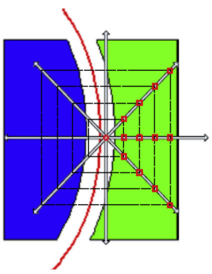

Figure 6. strategy for selecting N(p).

B. Statistical Based Image Inpainting

The overall architecture of the proposed system is shown in figure 3.9. The original image which is to be inpainted is input to the system. The input image is converted to gray-scale image. Then the image gradients are calculated. In next step the orientation matrix is calculated using calculated image gradients and then inpainting process is applied.

The first step after converting to gray-scale image is the calculation of the input image gradients in both x and y direction and then orientation matrix is calculated using these gradients.

1) Orientation matrix: For a given input image I, the Eqn.(2) for calculating our orientation matrix Θ

$$\Theta(x,y) = round\left(\left(arctan\frac{I_x(x,y)I_y(x,y)}{\Pi} + 1/2\right)n\right) \quad (2)$$

in which I_x and I_y are gradients in x and y directions. This formula generates a matrix containing integer values ranging from 0 to n.

2) Filling process: The unknown areas are filled using the pre calculated orientation matrix.The figure illustrates an input image I with a target region. The goal is to estimate the color components of all pixels of the target region using the information exists in the source region Φ. To do this, a model is presented which gradually propagates the information from outside of the boundaries toward the interior regions. The hole diminishes pixel by pixel until no unknown pixel remains unfilled. As shown in figure 3.10, firrst identify the one-pixel thick exterior and interior boundary regions which are denoted by $\delta\Omega^+$ and $\delta\Omega^-$ respectively. The Eqns.(3) and (4) morphological operations to extract these regions where B is a 3×3 structural element.:

$$\delta\Omega^+ = (\Omega \oplus B) - \Omega \quad (3)$$

$$\delta\Omega^- = \Omega - (\Omega \ominus B) \quad (4)$$

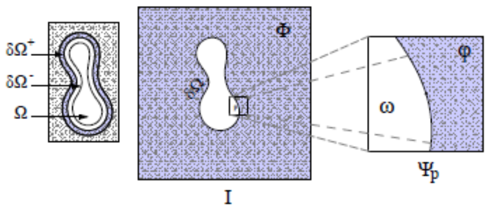

Figure 7. Notations and Definitions

16-bin histogram H of oriented gradients over Θ_p^ϕ. The histogram of oriented gradients (HOG) can capture edges or gradient structures and thus represents one of the most important local features of an image. Fig.3.11 illustrates an example of a 7×7 patch Θ_p of the orientation matrix centered at point p. Orientation of point p according to the orientation values of its neighboring pixels is calculated by making a histogram over Θ_p^ϕ and choosing the orientation of the highest peak. The highest peak in the histogram H corresponds to the dominant orientation of local gradients. However, this is not an appropriate method for our problem because in this way all neighboring pixels would have the same impact in determining the orientation value of p. Therefore,the following Eqn.(5):

$$H_k = \Sigma_{q\varepsilon\Theta_p^\phi} exp\left(-\frac{d_q^2}{2\sigma^2} \right) \tag{5}$$

When the propagation process of the orientation values gets completed, the filling process over the interior boundary regions $\delta\Omega^-$ commences by using the following Eqn.(6)

$$I(p) = \frac{\Sigma_{q\varepsilon\phi W_i(q) I_p(q)}}{\Sigma_{q\varepsilon\phi W_i(q)}} \tag{6}$$

in which p is a pixel on $\delta\Omega^-$, I_p is a 5×5 patch of I centered at p, W_i is a symmetric weight matrix of the same size, and ϕ is the source region of that patch.The weight matrix W_i, is defined with respect to the value of θ^*.

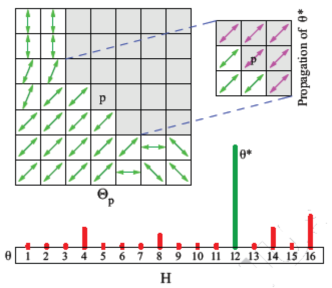

Figure 8. Illustration of a 7×7 patch Θ_p and its histogram. θ^* is the dominant orientation of the local gradients which is propagated into p and all its 8-connected neighboring pixels

C. PDE/Diffusion based inpainting

The basic idea of diffusion-based inpainting is to propagate local information with smoothness constraints, like heat propagation in physical structures. The propagation process imitates what is done by art restoration experts. In general, two issues need to be carefully considered in diffusion based inpainting: the first is how to describe the local image structure, and the second is in which direction the local image structure should be propagated.

A typical diffusion-based inpainting method was proposed by Bertalmio et al. [18] in 2000. In this work, the image Laplacian is used as a smoothness predictor for describing the local structural information, and an anisotropic model is employed to propagate the image Laplacian along the direction of the image isophote, which is perpendicular to the image gradient in each pixel point. Formally, the algorithm updates the pixel intensities iteratively inside the unknown region by solving the Eqn.(7), where t is the iteration time, t' is the update speed, $dI^t(x,y)$ is the update signal for $I^t(x,y)$. And $dI^t(x,y)$ is calculated by Eqn.(8), where $\bigtriangledown$ is the gradient operator, Δ I represents the image Laplacians, $\bigtriangledown I^\perp$ is the isophote direction (perpendicular to the gradient direction)

$$I^{t+1}(x,y) = I^t(x,y) + t'.dI^t(x,y) \tag{7}$$

$$dI^t(x,y) = \bigtriangledown(\Delta I^t(x,y)).\bigtriangledown I^{t\perp}(x,y) \tag{8}$$

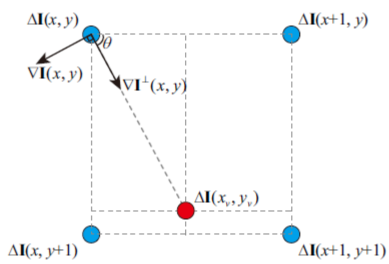

Figure 9. Illustration of the calculation of the change in the image Laplacian along the isophote direction, where Δ is the Laplacian operator and ÃŚ is the gradient operator.

Diffusion-based inpainting methods tend to prolong structures arriving at the boundary of the region to be filled in; hence they are suitable for propagating strong structures, or for filling small regions. If the region to be completed is large or of complex texture, the inpainted region appears blurred after a few diffusion iterations. Therefore, when such a technique is used for object removal, the altered region tends to be small, otherwise there would be obviously visible artifacts.

D. Exemplar-based image inpainting

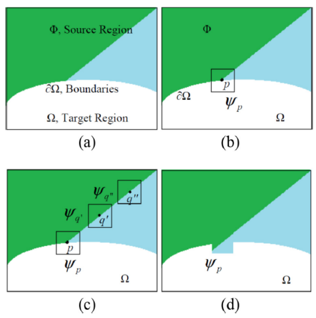

Figure 10. Structure propagation by exemplar-based inpainting. (a) Source and target regions; (b) target patch; (c) target patch and source patches; (d) filling result of one patch.

In this method, first a space varying updating strategy for the confidence term and a matching confidence term are proposed to reduce the fast dropping effect and improve the priority estimation, which is critical to the final inpainting results. Second, a structure consistent patch matching to take the distribution of patch differences into account. In the last, fast Fourier transform (FFT) is adapted for full image searching to achieve better and faster matching results. The image inpainting method can be summarized as follows

1) Initialize the confidence term C_s and the matching confidence C_m ;
2) Identify the boundary $\delta\Omega$connecting source region Φ, if $\delta\Omega$ is empty, the inpainting is finished.
3) Calculate the priorities for each pixel in $\delta\Omega$;
4) Select the pixel with highest priority $\hat{p}$ and search for best matching source patch $\Psi_{\hat{q}}$;
5) Fill the unknown pixels in $\Psi_{\hat{p}}$ with the pixels in corresponding

positions in $\Psi_{\hat{q}}$;
6) Estimate the confidence term C_s and the matching confidence C_m for newly filled pixels
7) Repeat steps (2) to (6).

$$P(p) = C_s(p) * C_m(p) * D(p) \tag{9}$$

Priority function with space varying updating strategy and matching confidence: The priority function is crucial to the final inpainting results. Priority function for each pixel in $\delta\Omega$ is calculated using Eqn.(9), where $C_s(p)$ is the confident term with space varying updating strategy and $C_m(p)$ is the matching confidence term. Select the pixel with highest priority p and search for best matching source patch φ_q based on Eqn.(10)

$$d(\varphi_{\hat{p}}, \varphi_q) = d_{SSD}(\varphi_{\hat{p}}, \varphi_q) * \left\{ \left[d_{STD}(\varphi_{\hat{p}} - \varphi_q) \right]^{\beta} + 1 \right\} \tag{10}$$

E. Hybrid image inpainting

To overcome the blur effect in diffusion-based inpainting and the long run time in exemplar-based inpainting, hybrid inpainting methods. A hybrid inpainting algorithm, called InP−h, to overcome the long processing time of the conventional hybrid methods while maintaining the visual quality. Precisely, InP−h addresses two tasks: (i) determining the best application sequence for inpainting textual and structural missing target patches and (ii) extracting the sub-image (i.e. source window) containing the best candidates of source patches.Basically InP−h performs the following three steps:
(i) separating the original image into two image layers, one containing structure information such as strong edges and corners, and the other containing texture information such as texture patterns;
(ii) applying an existing exemplar-based inpainting technique to the texture image layer and an existing diffusion-based inpainting technique to the structure image layer; and
(iii) combining the two inpainted results obtained from step (ii) in the missing region into one.

1) Application order for inpainting: A hybrid inpainting can consider several options for the application order of exemplar-based inpainting and diffusion-based inpainting. Three options are feasible: (i) diffusion-first, (ii) exemplar-first and (iii) diffusion-exemplar-alternate. From the facts that our objective is to reduce the processing time of hybrid inpainting without losing the inpainting quality and most diffusion-based inpainting techniques can formulate multiple target patches to be inpainted 'all together' into a set of PDEs, the first and second options would be more acceptable than the last option in terms of saving processing time. Furthermore, since contrary to exemplar-based inpainting, diffusion-based inpainting requires the processing time for selecting target patches in the execution order of the first option, the second option will save more time than the first one.

pre-processing: Generation of auxiliary images: From I_0, InP−h extracts three types of images: I_{mask} (mask image), I_{s_edge} (strong edges) and I_{w_edge} (weak edges).

- *I_mask* : it is used to mark the region to be inpainted, that is, setting the pixels in that region to white and the rest to black.
- I_{s_edge} : the Canny edge detector assigns a value for each pixel to indicate the degree of edgeness in I_0. I_{s_edge} is the collection of pixels whose edgeness values are $>$ 0.9. I_{s_edge} represents strong textures.
- I_{w_edge} : similar to the definition of I_{s_edge}, I_{w_edge} is the collection of pixels whose edgeness values are in between 0.3 and 0.9. I_{w_edge} represents weak textures.

Initialisation of parameters: Each of I_0, *I_mask*, I_{s_edge} and I_{w_edge} is then uniformly partitioned into basic blocks B_i of n×n size. For each B_i , two parameters p(i) and L(i) are initialised

2) exemplar-based inpainting: This step consists of three sub-steps:

- re-evaluation of parameters,
- selection of target patch, generation of source window and inpainting, and updating parameters
- termination conditions

re-evaluation of parameters: Algorithmic flow asserts that p(.) is 0 for every block with no known pixel at the current iteration. This step estimates the p(.) values for such blocks by referring the p(.) values of their neighbour blocks. The p(.) is computed by Eqn.(11)

$$p(i) = \left(\begin{array}{cc} \text{p(i)} & \text{if p(i)} \geq 0 \\ \eta \times max_{B_j}(\text{p(i)}) & \text{if p(i) = âĹŠ1} \end{array} \right. \tag{11}$$

From the p(.) values of all blocks, the average (p_{avg}), minimum (p_{min}) and maximum (p_{max}) are computed. Then, find two parameters p_{smooth} and p_{sharp}. By using the values of p_{smooth} and p_{sharp}, divide the interval [p_{min}, p_{max}] into three sub-intervals: rg_{small} = [p_{min}, p_{smooth}], rg_{middle} = [p_{smooth}, p_{sharp}] and rg_{large} = [p_{sharp}, p_{max}].

$$\text{p}_{smooth} = p_{min} + \varepsilon_1 \; (p_{avg} \text{ - } p_{min}) \tag{12}$$

$$\text{p}_{sharp} = p_{avg} + \varepsilon_2 \; (p_{max} \text{ - } p_{avg}) \tag{13}$$

Selection of target patch and generation of source window: InP-h iteratively expands B_{start} towards one of four directions: left, right, up and down (see Fig. 3b). The expansion direction is chosen based on the p value of the resulting window of each expansion. Let p_{exp} be the largest value and the corresponding direction is chosen for expansion. The process of iterative expansion considers the following two cases

- Case-1 (Figs. 3.15c1 and c2): the expansion stops if p_{exp} ε rg_{small} since the expansion leads to dimming texture feature.
- Case-2 (Fig. 3.15c3): the expansion continues as long as p_{exp} ε $rg_{middle} \cup rg_{large}$

In addition, there are three exceptions for handling the termination/continuation of the expansion other than cases 1 and 2 as in [12]. Once a source window is extracted by the

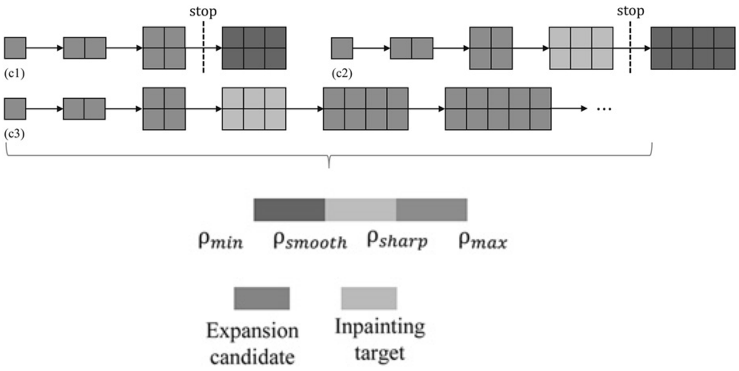

Figure 11. Termination of window expansion

expansion, a conventional exemplar-based inpainting is applied to the target block in the window using the source information in the window.

3) Termination: The iteration of exemplar-based inpainting stops when there is no block to be inpainted, which means there is no block whose p value is no less than p_{sharp}.

4) Diffusion-based inpainting: The target blocks that remain after Step 2 will be those located on the relatively smooth regions. Diffusion-based inpainting is applied to all the remaining blocks in Ω all together by generating a set of PDEs for the blocks

III. Discussions and finding

a) : Statistical Based Image Inpainting approach is mainly used for texture synthesis purpose through use of compact parametric statistical models. Portilla and Simoncelli [3] use joint statistics of wavelet coefficients. Heeger and Bergen [4] make use of color histogram at multiple resolutions for analysis of texture. Main problem of these methods which are based on parametric statistical models is that, they are applicable only to the problem of texture synthesis and not to the general problem of image completion. PDE-Based Image Inpainting, try to fill the missing region of an image through a diffusion process, by smoothly propagating information from the boundary towards the interior of the missing region. According to these techniques, the diffusion process is simulated by solving a partial differential equation, which is typically nonlinear and of high order. Bertalmio et al. [5] try to fill a hole in an image by propagating image Laplacians in isophote direction. Exemplar-Based Image Inpainting method, try to fill up the unknown region of an image simply by copying the content from the known part of the image. This approach is either patch based [8] or pixel based inpainting approach, meaning that final texture is synthesized by one pixel or by one patch at a time. Patch based methods achieve results of higher quality and they maintain higher order statistics of input texture. This method can be also used for video. It synthesizes new video texture by arranging recorded frame of the input video. Criminisi et el [1] used the exemplar-based inpainting technique for region filling and object removal purpose. In this, authors use filling order computations and texture synthesis technique.

b) : A hybrid inpainting algorithm, called InP−h, overcome the long processing time of the conventional hybrid methods while maintaining the visual quality. Precisely, InP−h addresses two tasks: (i) determining the best application sequence for inpainting textual and structural missing target patches and (ii) extracting the sub-image (i.e. source window) containing the best candidates of source patches. Hybrid inpainting method overcomes the problem of the traditional inpainting methods when they were combined. This overcomes the blur effect in diffusion-based inpainting and the long run time in exemplar-based inpainting.

TABLE I
Image inpainting approaches

Paper	Advantage	Disadvantage
[9]	Reproducing irregular texture	Not used for general problem of image completion.
[10]	Maintain structure of inpainted area	Not applicable when missing area is large.
[11]	Maintain good quality when missing area is large	Not applicable for low-resolution images.
[12]	Blur effect of diffusion inpainting and long run time of Exemplar inpainting are removed	

IV. Conclusion

Inpainting techniques can be classified into depth image inpainting and image inpainting. Depth image inpainting is done depth images. The image inpainting techniques are mainly divided into statistical based, PDE or diffusion based, exemplar based image inpainting. Statistical Based Image Inpainting approach is mainly used for texture synthesis purpose through use of compact parametric statistical models. PDE-Based Image Inpainting, try to fill the missing region of an image through a diffusion process, by smoothly propagating information from the boundary towards the interior of the missing region. Exemplar-Based Image Inpainting method, try to fill up the unknown region of an image simply by copying the content from the known part of the image. Hybrid inpainting is a combination of diffusion based and exemplar based inpainting. overcome the blur effect in diffusion-based inpainting and the long run time in exemplar-based inpainting. Hybrid inpainting has many advantages over the other methods. The disadvantages of diffusion method and exemplar method can be overcomed by using hybrid inpainting.

References

[1] Antonio Criminisi, Patrick Perez, Kentaro Toyama,(2004),"Region filling and object removal by

Exemplar-based Image Inpainting", IEEE transaction on image processing ,vol13,no9.IEEE .
[2] Olivier Le Meur, Mounira Ebdelli, Christine Guillemot,(2013), "Hierarchical Super-Resolution Based Inpainting",IEEE transaction on Image processing ,vol22.no10.IEEE.
[3] J.Portilla and E.P.Simoncelli,(2000),"A parametric texture model based on joint statistics of complex and wavelet coefficients "Int J comput.vis.vol40 no1 PP.49-70,SPRINGER.
[4] D.J.Heeger and J.R Bergen,(1995),"Pyramid-based texture analysis ",in proc ,PP.229-238.IEEE.
[5] M.Bertalmio, G.Sapiro, V.Caselles and C.Ballester, (2000), "ImageInpainting" in Proc.SIGGRAPH,PP.417-424.
[6] M.Bertalmio, G.Sapiro, V.Caselles and C.Ballester, (2003), "Simulataneous structure and texture image inpainting"comput.vis.Pattern Recognit pp 707-712.
[7] Y. Wei and M Levoy ,(2000),"Fast Texture synthesis using tree structured vector quantization",in Proc , PP 479-488.
[8] A. T. Umarani, P. J. Kulkarni,(2016),"A Novel Scheme of Color Image Inpainting by Prioritized Patches",International Conference on Advanced Communication Control and Computing Technologies,IEEE
[9] Weihai Chen, Haosong Yue n, Jianhua Wang, and Xingming Wu April 2014, "An improved edge detection algorithm for depth map inpainting", Optics and Lasers in Engineering, Vol 55, PP 69-77.ELSEVIER.
[10] Vahid. K. Alilou, Farzin Yaghmaee, (2017), "Non-texture image inpainting using histogram of oriented gradients", Journal of Visual Communication and Image Representation(accepted), Vol 48, PP 43-53.ELSEVIER.
[11] Haodong Li, Weiqi Luo, Jiwu Huang,(2017), "Localization of Diffusion-based Inpainting in Digital Images" ,Transactions on Information Forensics and Security.IEEE.
[12] Haixia Wang, Li Jiang, Ronghua Liang , Xiao-Xin Li, (2017), "Exemplar-based image inpainting using structure consistent patch matching", Neurocomputing, Vol 267.Elsevier.
[13] Lu Cai, Taewhan Kim, (2015), "Context-driven hybrid image inpainting", IET Image Processing, Vol 9, Issue: 10, PP 866 − 873.IEEE.
[14] H. Farid,(2009), "Image forgery detection," IEEE Signal Process. Magazine,vol. 26, no. 2, PP. 16-25.IEEE
[15] Guillemot, C., Le Meur,(2014) O.: "Image inpainting: overview and recent advances",IEEE Signal Process. Mag., PP. 127-144.IEEE