July 2012

Fast and Texture-Structure Preservable Inpainting and Its Application for Creating Virtualized-Reality Indoor Models

Kalaivani Thangamani

Service Engineering and Mixed Reality Laboratory, University of Tsukuba, Japan,
thangamani.kalaivani@aist.go.jp

Tomoya Ishikawa

Center for service research (CFSR), National Institute for Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, tomoya-ishikawa@aist.go.jp

Koji Makita

Center for service research (CFSR), National Institute for Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, k.makita@aist.go.jp

Takeshi Kurata

Service Engineering and Mixed Reality Laboratory, University of Tsukuba, Japan, t.kurata@aist.go.jp

Follow this and additional works at: <https://www.interscience.in/ijcsi>

Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

Thangamani, Kalaivani; Ishikawa, Tomoya; Makita, Koji; and Kurata, Takeshi (2012) "Fast and Texture-Structure Preservable Inpainting and Its Application for Creating Virtualized-Reality Indoor Models," *International Journal of Computer Science and Informatics*: Vol. 2 : Iss. 1 , Article 1.

DOI: 10.47893/IJCSI.2012.1057

Available at: <https://www.interscience.in/ijcsi/vol2/iss1/1>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Fast and Texture-Structure Preservable Inpainting and Its Application for Creating Virtualized-Reality Indoor Models

Kalaivani Thangamani¹, Tomoya Ishikawa², Koji Makita³ & Takeshi Kurata⁴

^{1&4}Service Engineering and Mixed Reality Laboratory, University of Tsukuba, Japan

^{2&3}Center for service research (CFSR), National Institute for Advanced Industrial Science and Technology (AIST), Tsukuba, Japan.

E-mail : {thangamani.kalaivani, tomoya-ishikawa, k.makita, t.kurata}@aist.go.jp

Abstract - This paper discusses the new inpainting algorithm for handling the images in the virtualized-reality indoor models. The untextured regions in the indoor model are textured by this proposed inpainting method which guarantees the texture and the structure propagation. We call this as a hybrid method because it combines the effectiveness of the two popular inpainting algorithms namely, the Exemplar-Based method and the Edge-Based image restoration method and rectifies the drawbacks in these two methods. The tedious patch search in the Exemplar-Based method is simplified by the usage of the hash table which holds the classified textures. The Edge-Based method can handle the structures only in the form of lines and circles whereas the proposed method can handle any type of curves by using the combination of superpixels and spline curves.

Keywords - *inpainting; indoor modeling; search by hashing.*

I. INTRODUCTION

Inpainting, the technique used for modifying/repairing the lost or deteriorated image parts in an undetectable manner finds its application in many fields. This paper shows the application of inpainting in improving the virtualized reality indoor models created by the 3D modeler. Virtualized reality model helps to turn the real world scene into the virtual one, by using the clues such as photos from the real world. The 3D modeler [1] helps the user to create the virtualized reality indoor models of their living room or work environment from a single or multiple photos by using the simple interaction techniques based on computer vision principles.

There are often untextured regions on some of the 3D planes since it is not easy to take a set of photos so that every region of the 3D model is included at least in one of those photos. Inpainting techniques may be employed for making up for the invisible regions with computer generated texture patches. Initially Exemplar-Based inpainting [2] was applied for inpainting the untextured regions and observed that the results were to be improved with the structure propagation, since the Exemplar-Based technique does not handle the structures globally. And also the patch selection process in [2] increases the overall computation time.

The Edge-Based inpainting [3] is another reasonable method which treats the structure in the global aspect. But the limitations are, the structures are treated in the form of lines and circles. This new inpainting technique combines the merits of these two methods and overcome their limitations. The novelty in the proposed method is the introduction of superpixel constraints and the usage of the hash tables.

II. RELATED WORKS

We are keen in maintaining the structures in and out of the mask region since the structures make an impact in every plane of the indoor environment. There are quite a good number of research studies in the area of structure propagation in inpainting. Jason C. Hung et al. [4] extend the Exemplar-Based inpainting by incorporating Bezier curves to construct missing edge information. Jian Sun et al. [5] referred the user drawn lines for the possible structure propagation. Nikos Komodakis et al. [6] proposed an alternate image completion method for the Exemplar based method with the Markov Random Field (MRF). There are good references on tensor voting [7] for the robust image synthesis. Treating the image in the global aspect recommends the good structure propagation [8].

A. Exemplar-Based Inpainting

The Exemplar-Based inpainting proposed by Criminisi et al. [2], fills the invisible region with the texture patches from the neighboring region. The Exemplar-Based algorithm sets the priority for the pixels in the mask boundary since the filling order is important in propagating the structure inside the mask region. The highest priority is given to the pixel which is surrounded by most of the data pixels and is in continuation of the strong edges. The suitable texture patch is selected by calculating the sum of the squared differences (SSD) of the pixel values between the texture patches. The patch size determines the quality of the inpainted result and it depends on the texture and structure nature of the particular image. The user has to adjust the patch size to be slightly larger than the largest distinguishable texture element, or “texel”, in the source region. When the patch size is not adjusted properly, it leads to the visible patch boundary which altogether makes the inpainted region odd from the rest of the image.

Though the Exemplar-Based method sets the priority function to take care of the structure continuity, the method follows the greedy way of sampling for filling the textures. So we propose to treat the structures in the global aspect and fixing the structure skeleton before filling the textures would lead to the guaranteed structure propagation.

B. Edge-Based Image restoration

Edge-Based image restoration by Andrei Rares et al. [3] relies on explicit edge information. The edge information is used for the reconstruction of the structure in the missing areas, as well as for guiding the pixel interpolation. The algorithm is explained in three main steps (a) edge detection and edge feature extraction; (b) image structure reconstruction; and (c) edge-based inpainting.

In the first step, the edges are detected around the mask using the watershed segmentation. The object edges are extracted in clockwise order, from a point of view lying inside the mask. Simple edge features are extracted for each edge, such as the luminance value on both sides of an edge and the local gradient magnitude along the edge. Only the edges that have at least a certain gradient magnitude are retained for the next steps.

In the second step, the algorithm tries to recover the structure of the image inside the mask region. This is an ill-posed problem, where virtually anything could have existed inside the mask area. This algorithm couples the edges based on the cost function which measures the local edge features and their affinity towards a locally fitted circle.

An edge group is formed with the set of consecutive edge couples such that no two couples cross each other. The similar edge groupings form the specific configurations and each configuration is given scores, based on the sequentiality measures. The configuration with minimum score is taken and their edge couples are extended over their fitted circle. This step recovers the missing structure inside the mask region.

In the third step, the recovered structures are filled with textures based on the pixel-based interpolation. This type of texture filling is otherwise called diffusion based inpainting since the pixel values in the neighborhood are diffused over the interpolated paths.

C. Connections with Our Proposed Method

Our proposed method adapts the cost function calculation from [3] and simplifies the cost term with the use of superpixels. In [3], the edges are extrapolated over the locally modelled circles whereas our method can handle any type of curves with the application of splines. The circle fitting function is adapted only for grouping the edge couples.

Our algorithm uses the patch based texture synthesis for filling the textures, as in [2]. Patch based texture filling is speeded up by the design of hash works.

III. 3D MODELER IN BRIEF

Interactive modeler [1], [9] initiates 3D modeling by analyzing 2D input photos. The viewpoint and the rotation angles at which each photo was taken are estimated by using the vanishing points obtained from pairs of lines in the actual 3D world.

The origin of the ground plane is interactively set by the user over which the texture-mapped 3D planes are added one by one for developing the 3D indoor model. The developed 3D planes can be translated, rotated, deformed or deleted using simple interaction techniques based on geometric constraints derived from the photos.

Fig. 1(a) and 1(b) explains simple 3D model creation from a single 2D photo in the modeler. Fig. 1(c) shows the various objects and the planes in the 3D model.

The regions which are invisible in any of the input photos inevitably hold textures of their frontal objects due to the projective texture mapping with GPU. The depth maps are computed for the developed 3D model. The invisible region is carved by checking the depths between the 3D planes.

The dominant point clusters in every plane forms the inpainting mask which is explained in Fig. 1(g). Fig.

1(d) shows the planes with the inpainting mask (shaded in green). The inpainted planes are shown in Fig. 1(e). The inpainted planes are put back into the 3D model and present the virtualized reality indoor environment which is shown in Fig. 1(f).

IV. ALGORITHM OVERVIEW

The comparison layout of our proposed superpixel based inpainting with the Exemplar-Based method and the Edge-Based image restoration method is shown in Fig. 2. The plane that needs to be inpainted is retrieved along its inpaint mask and the textures are grouped into superpixels. The boundary of the mask region is extracted and the superpixels that are in contact with the mask boundary are gathered and these are termed as the nearest superpixels which take part in the structure propagation.

These nearest superpixels are categorized into primary, secondary and discarded superpixels, based on their nature. If the superpixel is found to be of type primary, the edges are smoothly joined by spline curves using the De-Boor algorithm. If the superpixel is found to be of type secondary, the edges are grouped based on the cost function, and joined with the matching neighboring superpixel by spline curves. The third type is the discarded superpixels which are very small in size and do not hold any clues for structure propagation, are

merged with the neighboring primary or secondary superpixels based on their texture nature.

The superpixel categorization helps to handle the structure in global aspect which is lagging in the Exemplar-Based method. Also the superpixels simplify the edge grouping procedure and the cost function calculation in the Edge-Based image restoration.

Once the structure skeleton is formed, then the skeletons are filled with the texture flesh. These skeletons get the texture from the hash tables which are formed based on the Gray Level Co-occurrence Matrix (GLCM). The tedious patch search in the Exemplar-Based method is improved by means of search by hashing. Also this patch based inpainting gives better performance than the diffusion based inpainting which is followed in Edge-Based image restoration.

A. Superpixels

Superpixels correspond to small, nearly-uniform segmented regions in the image. The idea of superpixels is originally developed by X. Ren and J. Malik [10], they mention that the superpixels are local, coherent and preserve most of the structure necessary for segmentation at the scale of interest. The motivation behind our superpixel usage is obtained from Derek Hoiem et al. [11], who applied the superpixels in 3D model creation in their work, Automatic photo pop-up.

Fig. 1 : The overall workflow. (a) Single 2D image used for modeling. (b) 3D model before inpainting. (c) Various planes and objects in the 3D model. (d) Planes that need inpainting (inpaint mask shown in green). (e) Inpainted planes. (f) Inpainted 3D model. (g) Automatic inpainting mask generation.

Fig. 2 : Algorithm overview

The efficient graph based segmentation [12] is used for segmenting the image regions into superpixels. This graph based method segments the region based on the intensity, color, motion, location and other local attributes. The parameters that are used in the efficient graph based method are the smoothing factor sigma (σ), which decides the amount of smoothing applied for the image before segmentation, $K_{\min}$ and $K_{\max}$ which denote the minimum and the maximum size of the superpixels, respectively.

These values may be varied in order to define the suitable superpixels for good structure propagation.

1) *Edge couple*: The edges of the superpixel that are in contact with the mask boundary are defined by the term edge couples which is shown in Fig. 3(a). The

superpixel shown in Fig. 3(a) is one of the superpixels in Fig. 4(e).

The definition stated for edge couples in this paper is different from the one defined in [3]. In [3], the factors that decide the two edges to become edge couples are their luminance values, gradient magnitudes and their fitness on a common circle. This paper simplifies the detection of the edge couples by the application of the superpixels. Every two edges of the same superpixel that touches the boundary of the inpaint mask are considered as the edge couples since it is obvious that they have the same textural features.

2) *Primary and secondary superpixels*: The superpixel which encloses the mask region by its boundary is stated as the primary superpixel, an example of this kind is shown in Fig. 3(b). The mask region enclosed by this superpixel can be filled by the textures from this primary superpixel itself. They mostly do not need textures from the neighboring superpixels and hence no need of cost function calculation to find the best match. So these boundaries are joined by spline curves.

The secondary superpixels are the ones whose edges do not enclose the mask region but expects the continuation of its edge couple on the other side of the mask region which is shown in Fig. 3(f).

V. SPLINE BASED CURVE COMPLETION

The B-Splines [13] are used for completing the missing superpixel boundaries inside the mask region. It is difficult to predict the missing superpixel boundary inside the mask region. Almost all of the inpainting papers treat this issue based on some assumptions. This paper measures the orientation of the incoming edge couples and finds the similar edge couples from the rest of the edge data and applies the B-spline curve.

Fig. 3 : Superpixel boundary extension by splines

Fig. 4 : Example for the proposed inpainting algorithm

De-Boor algorithm is adapted for evaluating the spline curves. Given a parameter value, the algorithm finds the point on the B-spline corresponding to that parameter value. The widely known concept of B-Splines and De-Boor algorithm are not given in detail considering the length of this paper.

Given $n+1$ control points $P_0, P_1, \dots, P_n$ and the knot vector U is given by $\{U_1, U_2, \dots, U_m\}$, the B-Spline curves of degree P is defined by these control points and knot vector U , which is given in (1).

$$C(U) = \sum_{i=0}^n N_{i,p}(U) P_i \quad (1)$$

Where $N_{i,p}(U)$'s are the B-Spline functions of degree P .

$$N_{i,0}(U) = \begin{cases} 1 & \text{If } U_i \leq U < U_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$N_{i,p}(U) = \frac{(U - U_i)}{(U_{i+p} - U_i)} N_{i,p-1}(U) + \frac{U_{i+p+1} - U}{U_{i+p+1} - U_{i+1}} N_{i+1,p-1}(U) \quad (3)$$

A. De-Boor Method for Spline Curves

De Boor's algorithm provides a fast and stable way for finding a point on the B-Spline curve given a U in the domain. The application steps for the De-Boor algorithm is given as follows

Input: value U

Output: the point on the curve, $C(u)$

- (a). If U lies in $[U_k, U_{k+1}]$ and $U \neq U_k$,
let $h=p$ (inserting U up to p times) and $s=0$;
- (b). If $U = U_k$ and U_k is a knot of multiplicity s ,
let $h=p-s$ (inserting U upto $p-s$ times);
- (c). Copy the affected control points,
 $P_{k-s}, P_{k-s-1}, P_{k-s-2}, \dots, P_{k-p+1}$ and P_{k-p} to a new array
and rename them as $P_{k-s,0}, P_{k-s-2,0}, \dots, P_{k-p+1,0}$;
for $r = 1$ to h do
for $i = k-p+r$ to $k-s$ do
begin
Let $a_{i,r} = (U - U_i) / (U_{i+p-r+1} - U_i)$
Let $P_{i,r} = (1 - a_{i,r})P_{i-1,r-1} + a_{i,r}P_{i,r-1}$
end
 $P_{k-s,p-s}$ is the point $C(u)$

The red dots in the Fig. 3(c) shows the traced control points for the primary superpixel and the calculation details are shown in Fig. 3(d). After connecting all the points calculated for every knot interval, the missing structure has been traced out which is shown in Fig. 3(e). After the structure propagation, the texture filling process comes into play which is explained in detail in section VII.

VI. COST FUNCTION FOR SECONDARY SUPERPIXELS

The superpixels whose edges do not enclose the mask region but expects the continuation of its edge couple on the other side of the mask region are called as the secondary superpixels.

The edge couple grouping for the secondary superpixels is the place where the global treating of the structures is implemented. The secondary superpixel needs some weight measures unlike the primary superpixels for their edge couple grouping. The secondary superpixel which got the maximum number of edge couples are considered first for structure propagation. Every edge couple is checked with the rest of the free edge couples with its orientation and incoming angle. The edge couples are fitted to the boundary of the circle and checked the fitness. Based on the fitness measure, the affinity towards the edge couples in forming a group is decided.

At this point, we say that the proposed algorithm simplifies the cost function calculation followed in [3]. The cost function equation in [3] is given in (4). Equation (4) can be rewritten as in (5) for showing the meaning of each term in the cost function.

The terms $Intensity_{cw}$, $Intensity_{ccw}$ measures the intensity values in the clockwise and counterclockwise direction of the corresponding edges. And the $gradient$ measures the orientation of the edges. These terms test the edges and find whether they belong to the same object. If the edges have the similar intensity in the neighborhood and similar gradient measures over its edges, then there are chances for grouping these edges. The same logic can be met by the usage of the

superpixels. Obviously the boundaries of the superpixel have the similar features in terms of intensity and gradient.

The denominator term represents the flag setting for the spare edges. Spare edges are the ones which are not a part of a couple. Since the proposed algorithm have the system of merging the discarded superpixels to the neighboring primary or secondary superpixels, there is no need to monitor the denominator term. So it is enough to take care of the circle fitting term to check the orientation of the incoming edge couple and try to pick a match for the same. Hence the cost function of the proposed algorithm only concentrates on (6),

$$\omega_{i,j} = 1 - \delta_{i,j} \phi_{i,j} \theta_{i,j} \quad (6)$$

where $\delta_{i,j}$ is the spatial deviation of the edge couple from the fitted circle, $\phi_{i,j}$ is the angular consistency factor, and $\theta_{i,j}$ is the aperture quality factor. The circle fitting test for the secondary superpixel is shown in Fig. 3(g). There are two circles shown to fit the inner and the outer edge couples in the secondary superpixel. The inner circle is marked with the center c_1 and radius r_1 .

$$C_{i,j} = \sqrt{\frac{\beta_{i,j}^c (\lambda 1_{i,j}^c - \lambda 2_{i,j}^c)^2 + \beta_{i,j}^T (\lambda 1_{i,j}^T - \lambda 2_{i,j}^c)^2 + \beta_{i,j}^c \beta_{i,j}^T (\gamma 1_{i,j} - \gamma 2_{i,j})^2 + \omega_{i,j}^2}{\beta_{i,j}^c + \beta_{i,j}^T + \beta_{i,j}^c \beta_{i,j}^T + 1}} \quad (4)$$

$$C_{i,j} = \sqrt{\frac{Intensity_{cw} + Intensity_{ccw} + gradient + circle_fitting}{flags_for_spare_edges}} \quad (5)$$

Fig. 5 : Hash table making. Storing the texture patches in the corresponding hash bin

The fitting distance d , represented as the distance between the edge pixel and the center of the fitting circle. The angle $\Delta\alpha_1$ is used to calculate the angular consistency. The circle fitting test is similar to the Edge-Based image restoration algorithm, so the formulae and the conditions for calculating the spatial deviation, angular consistency and the aperture quality are adopted from [3]. Fig. 4 shows the implementation of the proposed inpainting algorithm with an example. The 2D image used for modeling and the created 3D model are shown in Fig. 4(a) and 4(b), respectively. Fig. 4(c)

shows the magnified view of the 3D plane that needs inpainting. The green region in Fig. 4(d) indicates the mask region which is detected automatically and the Fig. 4(e) shows the superpixel with the smoothing factor, σ is 0.5, K_{min} and K_{max} are 200 and 250, respectively. Fig. 4(f) shows the nearest superpixels and Fig. 4(g) to 4(i) shows the various superpixel boundary extension with splines and the final inpainted result is shown in Fig. 4(j). The 3D model after inpainting is shown in Fig. 4(k).

Fig. 6 : Patch query with hash table

Computation time chart for the Fig. 4(j)

Stages	Time (min)
Supersixel creation	0.11
Supersixel categorization	0.08
Hash table making	0.06
Inpainting	0.04
Total	0.29

(a) Result from the proposed algorithm
Time (min) : 0.29 (b) Trials with Exemplar-based inpainting
(Total number of mask pixels = 5533)

Fig. 7 : Result analysis

VII. GLCM AND HASH TABLE

Once the structures are propagated by supersixels and spline curves, then comes the part of texture filling. This section explains the usage of GLCM and hash table in texture propagation. The texture patches are classified and stored in the hash table for quick search during the patch filling process. The hash function relates the incoming texture patch to the address of the particular hash table bin. During the query step, the necessary patch is retrieved by a single/ minimum search. The texture information is adequately specified by a set of Gray Tone Spatial Dependence matrices [14] otherwise known as Gray Level Co-occurrence Matrix (GLCM) which are computed for various angular relationships (Ex. 0° , 45° , 90° or 135°) with defined distances between neighboring resolution pairs.

Fig. 5 explains the storage of texture patches in the hash table. The input image is quantized and the number of gray levels defines the row and column of the GLCM. Every cell in GLCM matrix holds the count for the corresponding gray tones as being neighbors. For Ex. $\#(0,1)$ denotes the number of times, the gray levels 0 and 1 happen to be neighbors. The patch size is supposed to be greater than the structures in the image which can be adjusted by the user. Co-occurring pairs oriented at 0° are also oriented at 180° , which generates a symmetrical GLCM. This concept extends to 45° and 225° , 90° and 270° , as well as 135° and 315° . As a result, only the lower triangular portion of the GLCM needs to be retained.

Every bin in the GLCM is given an ID and these are the inputs given to the hash function. For example, if the sample patch at the location (12,12) having the GLCM

entries at 1, 6, 13 and 20 bins, the hash function sums the number of entries and puts the texture patch at the hash bin 4. Inside every hash bin, the details such as the total number of patches in that bin, the GLCM entries for the corresponding patches, their patch location are stored in the separate data buckets.

During the patch query step, which is shown in Fig. 6, GLCM is calculated for the available data pixels in the masked patch and based on their ID, the patches are searched only in the respective bin. This step drastically reduces the patch search time whereas in the Exemplar-Based inpainting method, the system has to check all the patches for every patch fill.

The inpainted image is compared with the results from the Exemplar-Based inpainting method. Our proposed method is proved to give better results with less computation time. The graph in Fig. 6 compares the computation time between the Exemplar-Based method with our proposed method. The images which are inpainted in minutes by the Exemplar-Based method are inpainted in just few seconds with the help of the hash tables. Also the computation time is checked for various image size and various mask size. The performance is unaffected by these constraints such the patch size, mask size and the image size. The time taken for different stages in the proposed algorithm for the test image shown in Fig. 4(j) is given in the chart in Fig. 7. The resultant inpainted image is compared with the results from the Exemplar-Based method with different patch sizes. The image in Fig. 7(a) is superior in quality and inpainted in few seconds when compared to the Exemplar-Based method trials. Fig. 8 shows more examples for the proposed inpainting method. The images in Fig. 8(a), Fig. 8(c) and Fig. 8(e) are the

virtualized reality models before inpainting and the images in Fig. 8(b), Fig. 8(d) and Fig. 8(f) show the same models after inpainting.

VIII. RESULTS AND DISCUSSION

The proposed inpainting algorithm is tested in the 3D indoor model of our office. The aerial view of the 3D model before inpainting is shown in Fig. 9(a) and the 3D model after inpainting is shown in Fig. 9(b).

Also our modeling tool enables the user to have the first person view inside the 3D model. Fig. 10 shows the inpainted model and the first person view of the area enclosed by red rectangle. The white circle shows the location and orientation of the virtual camera. The inpainted 3D planes provide the photorealistic 3D model which is very much visible in the 3 different first person views as shown in Fig. 10.

Another example is given in Fig. 11 (virtualized reality model of a Japanese restaurant). Fig. 11(a) shows the top view of the entire 3D model and the area enclosed by the red rectangle is enlarged in Fig. 11(b) and Fig. 11(c). Fig. 11(b) shows the particular room before inpainting and Fig. 11(c) shows the same room after inpainting. The irrelevant textures are corrected by our proposed inpainting approach. Fig. 11(d) shows the texture details inside the room before inpainting and Fig. 11(e) shows the inpainted room. Fig. 11(f) and Fig. 11(g) shows another side of the same room before and after inpainting respectively. These scene comparisons show the importance and effectiveness of the proposed inpainting algorithm.

A. Quantitative Analysis

Table 1, shows the quantitative analysis of our proposed method. The proposed method is tested over three different 3D indoor models and the details are given in the table. The first column shows the name of the 3D model and their overall view. Second column states the number of 2D photos used for making the 3D model. Third column states the total number of the 3D planes used in the 3D model and the fourth column states the number of 3D planes that need inpainting. The fifth column records the time taken for the inpainting in Exemplar-Based inpainting method. The final column shows the time taken for our method to create the

inpainted models. Our method can produce the inpainted model of our office in just 23 min when compared to the traditional Exemplar-Based method which took nearly 3 hours. The other 2 inpainted models shown in the table also shows the efficiency of our method.

B. Qualitative Analysis

Since humans can evaluate image quality better, we conducted the subjective analysis by circulating an online questionnaire to compare the images inpainted by the proposed inpainting algorithm with 3 other inpainting algorithms namely, Navier-Stokes method, Exemplar-Based method and CS5 features of Adobe Photoshop. The above mentioned methods are chosen in order to have a fair comparison, since they too focus on preserving the texture and structure combination.

Fig. 12 shows the screenshot of our subjective analysis questionnaire. In the questionnaire, the names of the inpainting methods are not disclosed to the users to maintain impartial ratings. We have included nearly 30 screenshots of the indoor model environment and the inpainted results by the above mentioned methods. We set 5 scales for every inpainted result and the subjects are asked to rate the images based on the image quality. For Ex. rank 5 denotes the high quality image and the rank 1 shows the poor quality image. More than 40 subjects with computer science background answered our online survey and their scores are stored in the database.

1) *Merits and limitations of our algorithm:* The algorithm is tested for various kind of images, such as the simple textured images and the images with straight and/or curved structures and images with both texture and structure.

Table II shows the average scores received for some of the images in the questionnaire that contain simple textures. Each row contains the scene from the indoor model before inpainting, their inpainting mask and their inpainted results arranged in descending order according to their survey scores.

Table III summarizes the scores obtained for the questionnaire images that contain the simple textures, with the highest scores voted for our proposed method.

Fig. 8 : Qualitative analysis. (a), (c) and (e) Portion of the 3D model before inpainting. (b), (d) and (f) After inpainting

Table IV shows the second category of images from the questionnaire that contain both texture and structure where our algorithm has scored higher votes for preserving the texture and structure combination. The average scores for this category is shown in table V. Our algorithm has topped the list and proved to provide good quality inpainted images.

Table VI shows the third category of images, with the transparent surfaces where our algorithm has some

limitations. Though our algorithm has topped the score table VII, there are little differences between the other algorithms.

The reason is that our proposed algorithm does not able to differentiate the textures that belong to the corresponding plane from the one that is seen through the transparent surfaces. Since our algorithm follows the patch based texture filling, it searches for the least SSD patch irrespective of the patch location. Though this is a

critical problem that is to be focused in every other algorithm, Navier-Stokes method and the Photoshop have the good scores. The reason is that these algorithms follow the diffusion based inpainting where

the pixels in the neighborhood are taking part in the diffusion process.

Fig. 9 : 3D office model before and after inpainting (aerial view)

Fig. 10 : Inpainted office model in first person view

Fig. 11 : Japanese restaurant model. (a) aerial view. (b) Highlighted room before inpainting. (c) Highlighted room after inpainting. (d) and (f) Before inpainting (first person view). (e) and (g) After inpainting (first person view).

There are always chances for texture blurring in the diffusion based inpainting whereas the patch based texture filling guarantees the texture preservation.

C. Statistical Analysis

The survey scores are mapped in a box plot in Fig. 13, to have easy-to-interpret information in the compact way. The plot shows the minimum, maximum, median, first quartile (25 percentiles) and the third quartile (75 percentiles) values of the scores obtained by each of the category of images for the four comparison inpainting methods. The median value represents the middle value in the survey scores, is denoted by a line that separates the first and the third quartiles in the box plot. The position of the median helps to find the skewness in the score distribution. For the first category of images, our method is positively skewed (the position of the median is shifted towards the top of the box), which means that 50% of the scores are above the median value (3.47). There are also positive skewness found in the second and third category of images, whose median values are 3.48 and 3.49, respectively. Our method could not show the maximum score in the third category due to the limitations in our algorithm.

We also conducted the t-test on the survey results, to check whether the means of the two groups are statistically different from each other. The t-test is calculated by dividing the difference between the group means by the variability of groups.

Every two methods in the survey are taken and the t-test is calculated with 0.01 significant levels. If the

comparison methods have shown the t-value higher than or equal to the significant level, then we can say that the difference between the groups is not likely to have been a chance finding. The compared two methods might have shown a significant difference between their image qualities. The t-test is calculated for all the three categories of images shown in table II, table IV and table VI.

Table VIII shows the t-values calculated between every pair of the comparison methods for the images with simple textures. The t-values which are above the significance level of 0.01 are underlined.

Table IX shows the t-test results for the images in the second category which contain the texture and structure combination. The underlined t-values show the statistical difference between the compared methods. There is no significant t-value shown for the t-test between Exemplar-Based and the proposed method. The reason is that, the main advantage between the proposed method over the Exemplar-Based method is the computation time reduction which is already proven in quantitative analysis.

Table X shows the t-test results for the images in the third category which contain the transparent surfaces. There are significant values measured between each of the two compared methods. These evaluations prove that the survey results are statistically acceptable.

TABLE I: QUANTITATIVE ANALYSIS

3D Indoor Model		# of photos used	# of 3D planes used	# of 3D planes that need inpainting	Time taken for inpainting (Exemplar-Based inpainting)	Time taken for inpainting (Our method)
Our office		68	331	102	3.2 hrs	23 min
Japanese restaurant		68	425	89	2.8 hrs	18 min
ISMAR2009 Conference site		57	333	62	1.2 hrs	16 min

Fig. 12 : Screenshot of the online questionnaire (<http://staff.aist.go.jp/t.kurata/lab/Kalai/Questionnaire2011.pdf>)

TABLE II : QUESTIONNAIRE RESULTS (SIMPLE TEXTURED IMAGES)

No.	Before inpainting (Simple textured images)	Image with inpaint mask	Highest scored inpainted result (scores out of 5)	Second highest scored inpainted result (scores out of 5)	Lower scored inpainted result (scores out of 5)	Least scored inpainted result (scores out of 5)
1			 Our method (3.47)	 Exemplar-Based method (3.28)	 Navier-Stokes method (2.90)	 Photoshop (2.19)
2			 Our method (4.38)	 Exemplar-Based method (4.00)	 Photoshop (2.85)	 Navier-Stokes method (2.23)
3			 Photoshop (3.33)	 Our method (2.85)	 Exemplar-Based method (2.38)	 Navier-Stokes method (2.08)

TABLE III : AVERAGE SCORE FOR SIMPLE TEXTURED IMAGES

Method (for simple textured)	A (Adobe Photoshop CS5 feature)	B (Navier-Stokes method)	C (Exemplar- Based method)	D (Our proposed method)
Average score	2.76	2.55	3.17	3.47

IX. CONCLUSION AND FUTURE WORKS

We have proposed the fast inpainting method with the help of the superpixels and the hash tables which will preserve both the texture and the structure. The developed algorithm is good at handling the images from the virtualized reality indoor models which has been proven by the set of quantitative, qualitative, subjective and statistical evaluations. We work on

improving the algorithm further by making use of the 3D modeling data. The initial steps in our algorithm, such as the superpixel creation and categorization consumes trivial amount of time during the whole process, but these steps should be improved further to have the optimized version of our algorithm. Also we work to reduce the number of user inputs which are needed to accomplish these steps. We plan to improve

the design of the hash table from a single table to the hash library which will hold the classified texture patches from the entire indoor model. The hash library will provide the flexible choice of texture selection for the inpainting planes, in case of limited.

TABLE IV : QUESTIONNAIRE RESULTS (IMAGES WITH TEXTURE AND STRUCTURE)

No.	Before inpainting (images with texture and structure)	Image with inpaint mask	Highest scored inpainted result (scores out of 5)	Second highest scored inpainted result (scores out of 5)	Less scored inpainted result (scores out of 5)	Least scored inpainted result (scores out of 5)
1			Our method (3.71)	Photoshop (3.66)	Exemplar-Based (3.19)	Navier-Stokes method (2.95)
2			Our method (3.49)	Exemplar-Based method (3.08)	Navier-Stokes method (2.80)	Photoshop (2.61)
3			Our method (3.48)	Exemplar-Based method (3.32)	Photoshop (2.90)	Navier-Stokes method (2.28)
4			Photoshop (4.00)	Exemplar-Based method (2.90)	Our method (2.71)	Navier-Stokes method (2.38)
5			Photoshop (3.40)	Exemplar-Based method (3.00)	Our method (3.00)	Navier-Stokes method (1.90)
6			Our method (3.94)	Exemplar-Based method (3.26)	Photoshop (3.09)	Navier-Stokes method (2.09)
7			Navier-Stokes method (2.85)	Our method (2.79)	Photoshop (2.71)	Exemplar-Based method (1.80)

TABLE V : AVERAGE SCORES FOR IMAGES WITH TEXTURE AND STRUCTURE

Method (for texture and structure)	A (Adobe Photoshop CS5 feature)	B (Navier-Stokes method)	C (Exemplar-Based method)	D (Our proposed method)
Average score	2.97	2.63	3.16	3.40

TABLE VI : QUESTIONNAIRE RESULTS (IMAGES WITH TRANSPARENT SURFACES)

No.	Before inpainting (Images with transparent surfaces)	Image with inpaint mask	Highest scored inpainted result	Second highest scored inpainted result	Less scored inpainted result	Least scored inpainted result
1			Navier-Stokes method (4.04)	Photoshop (3.19)	Our method (2.66)	Exemplar-Based method (2.33)
2			Navier-Stokes method (3.95)	Our method (3.57)	Photoshop (3.33)	Exemplar-Based method (2.95)
3			Our method (3.90)	Exemplar-Based method (3.61)	Photoshop (3.42)	Navier-Stokes method (2.19)
4			Exemplar-Based method (3.59)	Our method (3.42)	Photoshop (3.35)	Navier-Stokes method (2.61)

TABLE VII : AVERAGE SCORES FOR IMAGES WITH TRANSPARENT SURFACES

Method (for transparent surfaces)	A (Adobe Photoshop CSS5 feature)	B (Navier-Stokes method)	C (Exemplar- Based method)	D (Our proposed method)
Average score	3.32	3.19	3.12	3.38

Fig. 13 : Statistical analysis (comparison between the survey scores)

TABLE VIII : STATISTICAL ANALYSIS (T-TEST FOR THE SIMPLE TEXTURED IMAGES)

Method (for simple textured)	Photoshop CS5 features	Navier-Stokes method	Exemplar-Based method	Proposed method
Photoshop CS5 features	-	<u>0.37</u>	<u>0.25</u>	<u>0.06</u>
Navier-Stokes method	-	-	<u>0.08</u>	<u>0.02</u>
Exemplar-Based method	-	-	-	<u>0.45</u>

TABLE IX : STATISTICAL ANALYSIS (T-TEST FOR THE IMAGES WITH TEXTURE AND STRUCTURE)

Method (for texture and structure)	Photoshop CS5 features	Navier-Stokes method	Exemplar-Based method	Proposed method
Photoshop CS5 features	-	<u>0.30</u>	<u>0.07</u>	<u>0.02</u>
Navier-Stokes method	-	-	<u>0.71</u>	<u>0.45</u>
Exemplar-Based method	-	-	-	0.0003

TABLE X : STATISTICAL ANALYSIS (T-TEST FOR THE IMAGES WITH TRANSPARENT SURFACES)

Method (for transparent surfaces)	Photoshop CS5 features	Navier-Stokes method	Exemplar-Based method	Proposed method
Photoshop CS5 features	-	<u>0.80</u>	<u>0.55</u>	<u>0.82</u>
Navier-Stokes method	-	-	<u>0.89</u>	<u>0.73</u>
Exemplar-Based method	-	-	-	<u>0.53</u>

textures in some of the planes. Also there are necessities to handle the patches in different angles. In some cases, the patches are to be rotated to fix in to the curved path. These necessities and challenges will encourage our future works.

ACKNOWLEDGMENT

This work was supported by Strategic Japanese-French Cooperative Program on Information and Communications Technology Including Computer Sciences (ANR and JST).

REFERENCES

- [1] T. Ishikawa, K. Thangamani, M. Kourogi, A.P. Gee, Walterio Mayol-Cuevas, Keechul Jung and T. Kurata, "In-Situ 3D indoor modeler with a camera and self contained sensors", Proc. HCII2009, San Diego, CA, LNCS, vol. 5622, pp. 454-464.
- [2] A. Criminisi, P. Perez and K. Toyama, "Object Removal by Exemplar-Based inpainting", Proc. IEEE CVPR, vol. 2, 2003, pp. 721-728.
- [3] A. Rares, M.J.T. Reinders, J. Biemond, "Edge-Based image restoration", IEEE Transactions on Image Processing, vol. 14, 2005, pp. 1454-1468.
- [4] Jason C. Hung, Chun-Hong Hwang, Yi-Chun Liao, Nick C. Tang, Ta-Jen Chen, "Exemplar-based image inpainting base on structure construction", Journal of software, Special issue on "Mobile System, Agent Technology, and Web Services", vol. 3, no.8, 2008, pp. 57-64.
- [5] Jian Sun, Lu Yuan, Jiaya Jia and Heung-Yeung Shun, "Image completion with structure propagation", SIGGRAPH2005, vol. 24, pp. 861-868.
- [6] N. Komodakis and G. Tziritas, "Image completion with global optimization", Proc. IEEE CVPR, 2006, pp. 442-452.
- [7] Jiaya Jia and Chi-Keung Tang, "Image repairing: Robust image synthesis by adaptive ND tensor voting", Proc. IEEE CVPR, 2003, pp. 643-650.
- [8] A. Levin, A. Zomet and Y. Weiss, "Learning how to inpaint from global image statistics", Proc. ninth IEEE conference on computer vision, vol. 1, 2003, pp. 305-312.
- [9] T. Kurata, M. Kourogi, T. Ishikawa, J. Hyun and A. Park, "Service cooperation and co-creative intelligence cycles based on mixed-reality Technology, Proc. INDIN 2010, pp. 967-972.
- [10] X. Ren and J. Malik, "Learning a classification model for segmentation", Proc. 9th International conference on computer vision, vol. 1, 2003, pp. 10-17.
- [11] D. Hoiem, A. A. Efros, M. Hebert, "Automatic Photo pop-up", ACM Transactions on Graphics(TOG), vol. 24, no. 3, 2005, pp. 577-584.
- [12] P.F. Felzenswalb, D.P. Huttenlocher, "Efficient Graph-Based Image Segmentation", International Journal of Computer Vision, vol. 59, no. 2, 2004.
- [13] Spline details, <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/>
- [14] R.M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification", IEEE Transactions on Systems, man and Cybernetics, vol. SMC-3, no. 6, 1973, pp. 610-621.