IEIEC Technical Report MVE2012-18, Vol.112, No.106, pp.55-60 (2012)

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

仮想化現実屋内モデリングのためのハッシュに基づく高速なインペインティング

カライバニ タンガマニ[†,‡] 石川 智也[‡] 牧田 孝嗣[‡] 蔵田 武志[†,‡]

† 筑波大学 〒305-8573 茨城県つくば市天王台 1-1-1
‡ 産業技術総合研究所 〒305-8568 茨城県つくば市梅園 1-1-1
E-mail: {thangamani.kalaivani, tomoya-ishikawa, k.makita, t.kurata}@aist.go.jp

あらまし 本稿では，ハッシュテーブルを利用することで従来の事例に基づくインペインティング処理を高速化する方法を提案する．事例に基づくインペインティング処理で最も計算コストが高い処理は，インペインティング対象となる領域を構成する小領域と，画像中のそれ以外の領域との比較探索処理である．提案手法では，グレーレベルの同時生起行列に基づいて設計したハッシュ関数により，この比較探索処理を定数時間での探索とする．このハッシュ関数で生成されたハッシュテーブル上で各小領域のテクスチャは独立したアドレスが割り当てられているため，低計算コストでの比較探索が可能となる．提案手法と従来手法を写真から対話的に作成された仮想化現実屋内モデル中の写真未撮影箇所に適用し，インペインティング処理に要する計算時間を比較した．その結果，従来手法に比べ短時間でのインペインティング処理が可能であることを確認した．また，現実環境の再撮影と再モデリングを行うことなくモデルの視覚的違和感を軽減することが可能であり，モデリングの後処理として提案手法が効果的であることを確認した．

キーワード ハッシュ，インペインティング，事例に基づくインペインティング，グレーレベルの同時生起行列

Fast Hash-Based Inpainting for Virtualized-Reality Indoor Modeling

Kalaivani THANGAMANI[†,‡] Tomoya ISHIKAWA[‡] Koji MAKITA[‡] and Takeshi KURATA[†,‡]

† University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki, 305-8577 Japan
‡ Center for Service Research, 1-1-1 Umezono, Tsukuba-shi, Ibaraki, 305-8568 Japan
E-mail: {thangamani.kalaivani, tomoya-ishikawa, k.makita, t.kurata}@aist.go.jp

Abstract This paper discusses the hash table usage for the Exemplar-Based inpainting method to speed up the inpainting process. The time consuming stage in the Exemplar-Based inpainting is the texture patch searching process where the occluded texture patch has to be compared with all other texture patches available in the rest of the image. The proposed hash function reduces the patch search into single/minimum search. Each texture patch is given an unique location in the hash table so that they are easily picked up by their addresses during every iteration. Gray level co-occurrence matrix (GLCM) is used for designing the hash function. The proposed hash works are added to the Exemplar-Based inpainting and this new hash-based inpainting method is tested in the planes of the virtualized-reality indoor model. There are often un-textured regions or the distorted textures in the 3D planes of the virtualized-reality model which needs inpainting. Applying the proposed fast hash-based inpainting, works for the efficient post processing in the virtualized-reality indoor modeling. This removes the necessity for the users to capture the left out scenes for the hidden regions. The proposed fast hash based inpainting reduces the computation time and provides good quality textures. The novelty of this paper lies in the application of the GLCM matrix in designing the hash function and the occlusion handling measures.

Keyword Search by hashing, Digital inpainting, Exemplar-Based inpainting, Gray level co-occurrence matrix

1. Introduction

Inpainting, the technique used for modifying/ repairing the lost or deteriorated image parts finds its application in many fields. This paper proposes the new inpainting algorithm which finds its application in improving the virtualized reality models created by the 3D modeler [1,2].

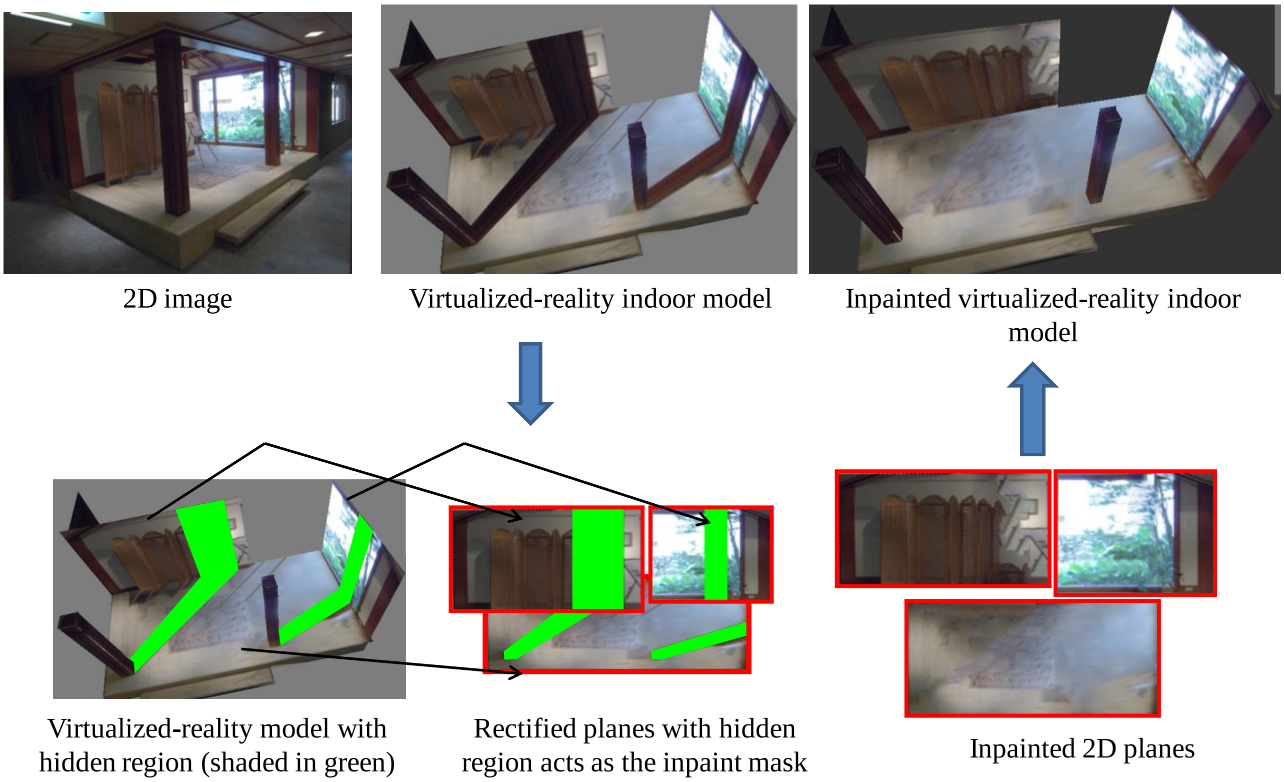


Figure 1. Inpainting for the virtualized-reality indoor modeling

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 allows the user to create the indoor environments of their desired location from a single or multiple photos by using simple interaction techniques based on computer vision principles. Projective texture mapping is used in the 3D modeler for mapping the textures over the 3D planes. 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. Exemplar-Based inpainting method [3] which is known for its fine structure propagation details, is adopted for handling the untextured regions. The computation time becomes a constraint in using this method and it is necessary to reduce the computation time for the efficient post processing in the virtualized reality indoor modeling.

2. Virtualized-Reality indoor models

Figure 1 shows the example for the virtualized-reality indoor modeling with the interactive indoor modeler [1,2]. The figure shows the 2D image that is used for modeling the indoor environment. The interactive indoor modeler helps the user in the creation of the indoor model. The regions that are not captured in the 2D image are not mapped correctly in the 3D world. Obviously these hidden regions will get the textures of their frontal planes, as the result of the projective texture mapping. The inpainting process are applied for the correcting these hidden regions. The hidden region is traced automatically by the modeler features and these hidden regions act as the inpainting mask. The figure shows the complete flow of the post processing in the virtualized reality indoor modeling. The planes that needs texture correction are rectified along with their inpaint mask and subjected to inpainting. The inpainted planes contribute for the visually pleasing indoor model.

3. Exemplar-Based inpainting

The Exemplar-Based inpainting proposed by Criminisi et al. [3], fills the invisible region with the texture patches from the neighboring region. This method is the patch based inpainting, which fills the mask region by one texture patch per iteration. The patch size determines the quality of the inpainted result and it depends on the texture and structure nature of the particular image. The iteration is continued till the entire mask region is filled by the texture patches. The Exemplar-Based algorithm sets

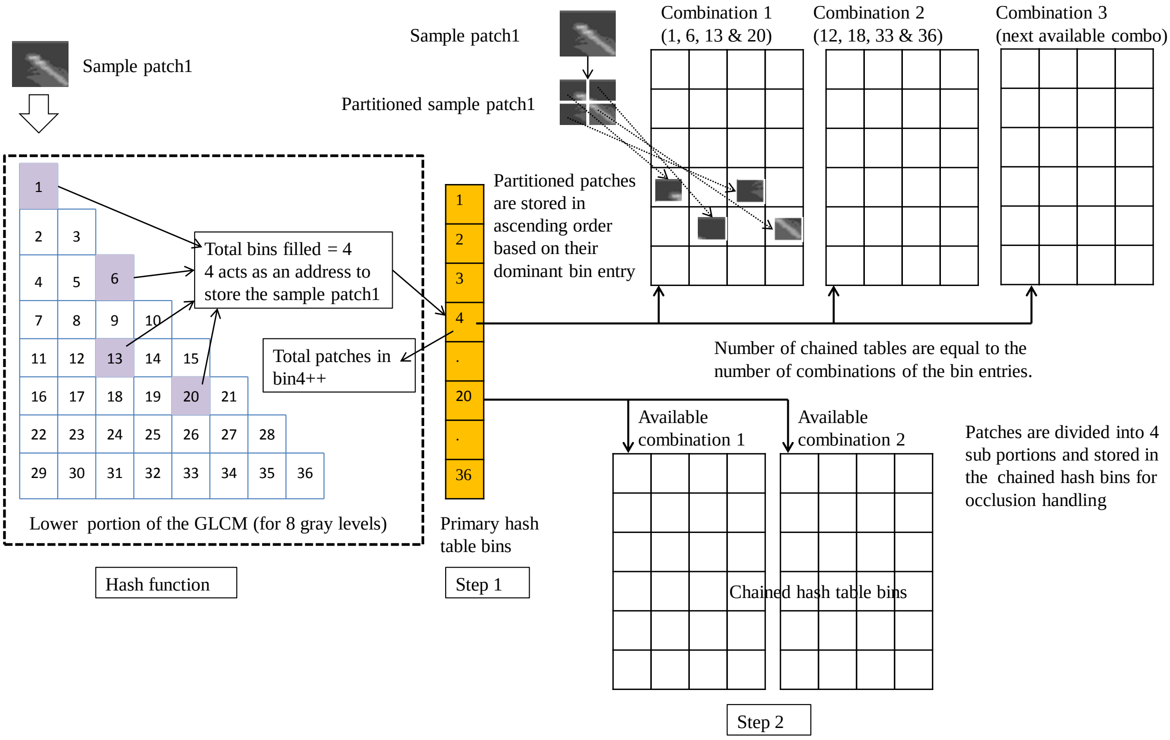


Figure 2. Making of the hash table (Storing the patches in the hash bin)

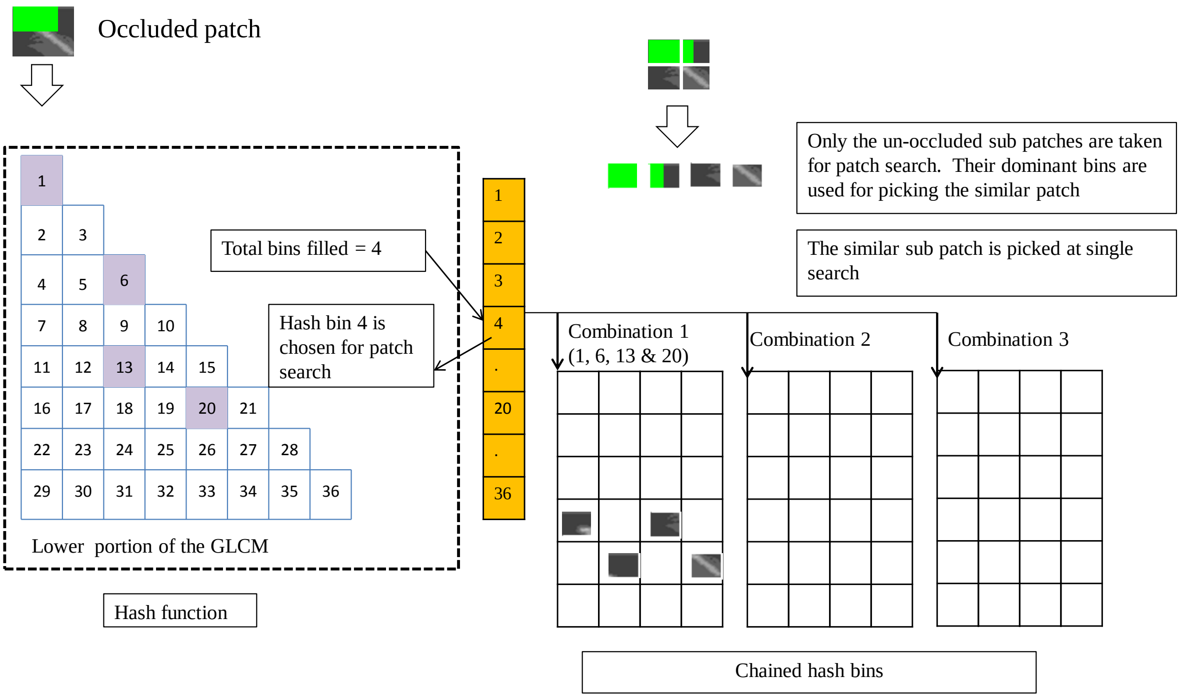


Figure 3. Texture patch query from the hash table

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. During every iteration, the occluded patch is compared with all other patches in the image which counts for thousands and thousands of comparisons. This step counts for the increase in the inpainting time. There comes the necessity to develop an alternative method which finds the location of the needed patch at a single attempt, avoiding the tedious patch searching process. Hash tables are well known for the quick access of data elements in an array. The following section explains the design of hash function.

4. Related works- Improvements in the Exemplar-Based inpainting

There are many contributions for the improvements in the structure propagation part of the Exemplar-Based inpainting. But there are few references on the improvements of the patch searching stage. Fast query for Exemplar-based image completion [4] provides the remedy for the exhaustive search in the Exemplar-Based inpainting method. The authors decompose the exemplars into the frequency coefficients and select fewer coefficients which are the most significant to evaluate the matching score. The authors also developed a local gradient-based algorithm to fill the unknown pixels in a query image block. These two techniques bring the ability of input with varied dimensions to the fast query of similar image exemplars. The fast query is based on a search-array data structure, and can be conducted very efficiently.

Fast and enhanced algorithm for Exemplar-Based Image Inpainting proposed by Anupam et al. [5] suggests the solution for the problem where the same SSD values are calculated for more than one patch. The authors proposed the coordinates for finding the possible location of the patches. By reducing the area of patch search, the computation time can be reduced.

5. Hash function design

The hash function should be designed in such a way that it allots an unique address for every patch. The principle of hash function [6] is that to convert the input to an address of the corresponding hash bin where the input can be stored. So the hash function should be capable of handling the texture patches in terms of numerical values. Textures are effectively represented by the Haralick features [7] such as the energy, entropy and other subsequent parameters. GLCM is the basic gray level matrix behind the Haralick features calculation. There are contributions in the area of quick GLCM calculation [8,9].

Figure 2 shows the design of hash function with the GLCM. The well-known calculation part of the GLCM is omitted in this paper. Due to the similarity in the GLCM structure, only the lower triangular portion of the GLCM is considered. Figure 2 shows the GLCM structure for 8 gray levels forming the 8 rows and 8 columns. Every bin is given a virtual number and the total number of filled bins forms the address for the input patch. The sample patch1 is the quantized image patch (reduced to 8 gray levels), fills the GLCM bins numbered as 1, 6, 13 and 20, in total, 4 bins are filled. So the sample patch1 is stored in hash bin whose address is 4. This step groups the similar gray level patches into the single bin. The effective design would be storing every patch in an unique address which is an advancement from our previous contributions [10].

The whole patch is divided into 4 sub quarters and they are stored in ascending/descending order of their dominant bin entry. For every group of patches in the particular hash bin, there will be one gray level that holds the highest value and the bin that holds this gray level is called the dominant bin. There may be different combinations for every hash bin entries. The number of chained hash entries are equal to the number of different GLCM combinations. The sub quarters are stored in the subsequent chains of every hash bin and the linked list serves the best connection between the chained combinations. This stage helps to handle the occluded patch comparison.

6. Hash table query

Figure 3 explains the query process with the hash table. The hash function maps the hash bin for the occluded patch. The occluded patch is divided into four quarters and the un-occluded quarter is considered for the search over the chained hash entries. According to the GLCM combination of the un-occluded patch, the search is transferred to the corresponding chained entry. Then the address nearer to the dominant bin count of the un-occluded quarter is considered as the needed patch. Once this sub quarter is found out, the corresponding whole patch is retrieved for the patch filling process in the Exemplar-Based inpainting method.

7. Blending the hash works with the Exemplar-Based method

Figure 4 shows the normal Exemplar-Based method and the proposed fast hash-based inpainting method which is nothing but the Exemplar-Based method along with the hash tables. The position of blending the hash works with the Exemplar-Based inpainting is shown in detail in Fig. 4. The hash table is made before the inpainting begins. Once the occluded patch is found out as the result of the priority mapping, the occluded patch queries the hash table for the matching texture patch. By the inclusion of hash table, the patch search is limited to single/minimum search. This is the step which counts for the drastic reduction of the computation time. The normal Exemplar-Based inpainting takes N searches for every patch fill, where N represents the eligible texture patches which are free from occlusion.

8. Results and discussion

The proposed hash-based inpainting is tested successfully in the planes of the virtual reality model. The before and after effects of inpainting for some of the indoor models are shown in Fig. 5. The inpainting time is the main criterion that is to be observed in the hash based inpainting. The quantitative analysis is given in table 1. The indoor models of our office, a Japanese restaurant and the ISMAR2009 site are taken for testing the hash based inpainting. The time taken for inpainting the planes with the Exemplar-Based inpainting is compared against the hash table included Exemplar-Based method. The time shown in the last column of the table 1 is the total time that includes the hash table making and the inpainting process. Usually the hash table making is the preprocessing step before starting the inpainting. From the table, it is evident that the computation time is drastically reduced with the help of the hash works. The GLCM is tested with various gray levels such as 8, 16, 32 etc., and the inpainted results shows only trivial difference between the gray levels. This property shows the robustness of the designed hash function.

9. Conclusion and future work

The hash table blended Exemplar-Based inpainting method is tested on different indoor models. The future work will extend the hash table to the next dimension which will hold the entire texture patches from the 3D model. The extended hash table would be helpful in inpainting the planes that have limited or distorted textures. The plane orientation is considered for separating and grouping the textures in the extended hash table. The hash table may be designed to hold the texture patches with different orientations to handle the curved structures. These necessities and challenges encourage the future works.

ACKNOWLEDGEMENTS 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”, HCII2009, San Diego, CA, LNCS, Vol. 5622, pp. 454-464, 2009.

[2] T. Kurata, M. Kourogi, T. Ishikawa, J. Hyun and A. Park, “Service cooperation and co-creative intelligence cycles based on mixed-reality technology”, Proceedings of INDIN 2010, pp. 967-972, 2010.

[3] A. Criminisi, P. Perez and K. Toyama, “Object removal by Exemplar-Based inpainting”, IEEE CVPR, Vol. 2, pp. 721-728, 2003.

[4] Tsz-Ho Kwok, Hoi Sheung and Charlie C.L. Wang, “Fast query for exemplar-based image completion”, IEEE Transactions on Image Processing, Vol. 19, No. 12, pp. 3106-3115, 2010.

[5] Anupam, Pulkit Goyal and Sapan Diwakar, “Fast and enhanced algorithm for exemplar based image inpainting”, Proceedings of 4th pacific-rim symposium on image and video technology (PSIVT2010), 2010.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to algorithms”, MIT press, second edition.

[7] 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, pp. 610-621, 1973.

[8] Alice Perebski, Nicolas Vandenbroucke and Ludovic Macaire, “Haralick feature extraction from LBP images for color texture classification”, IEEE Image Processing Theory, Tools and Applications, 2008.

[9] Eizan Miyamoto and Thomas Merryman Jr , “Fast calculation of Haralick texture features”, Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA.

[10] Kalaivani Thangamani, Tomoya Ishikawa, Koji Makita and Takeshi Kurata, “Fast and texture-structure preservable inpainting and its application for creating virtualized-reality indoor models”, International Journal of Computer Science and Informatics, ISSN: 2231-5292, Vol. 1. Issue 3, pp. 85-101, 2012.

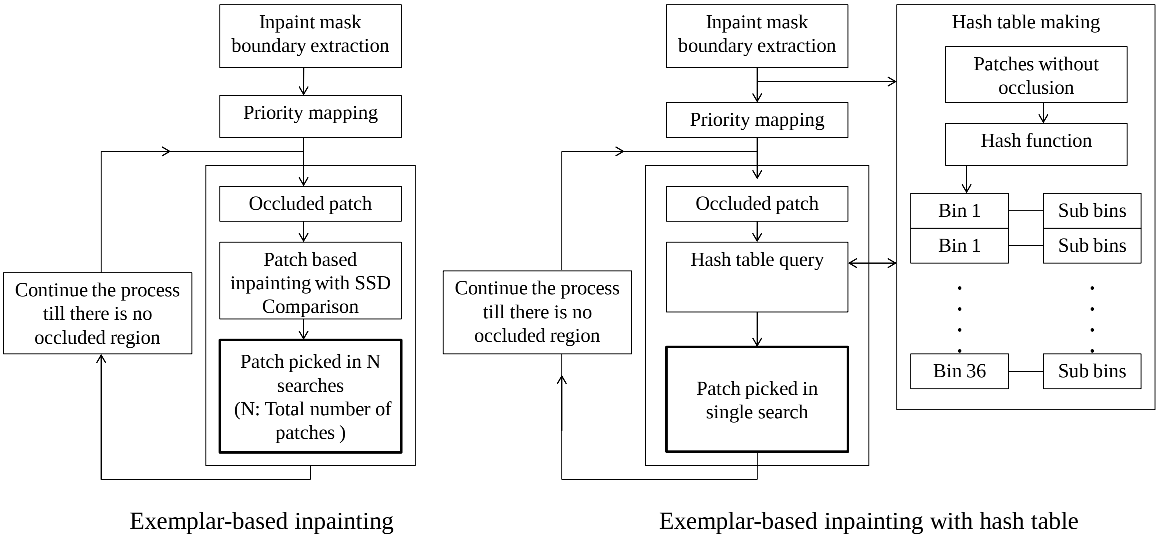
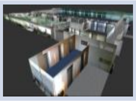


Figure 4. Comparison between the Exemplar-Based inpainting and the proposed method.

Table 1. Quantitative analysis

Virtualized-Reality indoor model	# of photos used	# of 3D planes used	# of 3D planes that need inpainting	Time taken for Exemplar-Based inpainting	Time taken for hash based inpainting (time includes the making of hash table)
office 	68	331	102	3.2 hrs	12.5 min
Japanese restaurant 	68	425	89	2.8 hrs	7.2 min
ISMAR2009 site 	57	333	62	1.2 hrs	5.3 min

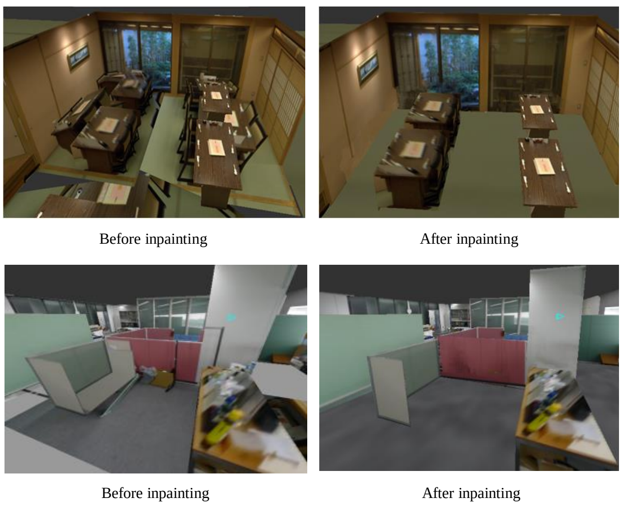


Figure 5. Qualitative analysis