Video Inpainting

Ankita S. Bagul
Computer Department,
Pune University,
GESCOE, Nasik, India

Asha S. Bhalerao
Computer Department,
Pune University,
GESCOE, Nasik, India

Vaishali N. Dhage
Computer Department,
Pune University,
GESCOE, Nasik, India

ABSTRACT
Video inpainting or completion is a vital video improvement technique used to repair or editing of digital videos. In this paper, we are describing masking and nearest neighbor algorithm for repairing damaged frames in video films, focusing on maintaining good spatiotemporal continuity. Masking algorithm is used for detection of scratches or damaged portions in video frames. Nearest neighbor algorithm used for quickly finding approximate nearest neighbor matches between image patches. The given algorithm offers performance improvement, because of that anyone can use it in interactive editing tools. Video completion has been used all over the world to convert cultural artifacts such as vintage videos and films into digital form. However, such types of videos are usually very poor in quality and most of the time it contain unstable luminance and damaged content. Experiment on many frames of videos show the benefits of proposed approach which lead to natural looking videos with less annoying artifacts.

General Terms
Image processing, Frame Extraction, Frame processor, AVI Splitter, AVI Builder.

Keywords
Inpainting, avi, patching, masking, spatiotemporal, luminance.

1. INTRODUCTION
Now days, converting cultural and historical artifacts such as vintage films and videos into digital format have become an important trend. Because of their age visual quality of such frames and videos after digitization is usually low and often contain unstable luminance and damaged content. Video inpainting is one of the most challenging techniques, which helps users to repair frames where content is missing or damaged. Video inpainting is the technique in which the damaged frames are extracted from the entire video and after that the damaged area is selected and this selected damage areas are filled by pasting the best match patch from the neighboring or current frame and these newly repaired frames are replaced by old damaged frames. Thus video completion or video inpainting repair the damaged frame in the video and finally gives the good quality video as output [1]

In video inpainting damaged region may be placed in all the video frames at same location or continuously moving location .The damaged region can be think of as a hole or tunnel in the video. In video inpainting method, the damaged area is repaired by referring the data in neighboring frames in such a way that it has smooth transition of object having a motion [2].

In video inpainting approach entire video Sequence is analyzed, after that masking algorithm is applied. Once the damaged frames detected then nearest neighbor algorithm used to find best match patch for that damaged patch .once best match patch found then patch pasting process is carried out. In this way frames are repaired. Thus the technique increases the quality of the video.

2. LITERATURE SURVEY
The main goal of image inpainting [3] is to inpaint or replace Damaged or missing region by using local information present in the current image [4], [5]. Most of the inpainting techniques find this information in the image itself. Correct reproduction of both textures and geometric structures is one of the main difficulties of this task. The problem raise in video inpainting technique is inpainting a spatio-temporal hole in a video. This resulted into the new technical challenges such as inpainting foreground, background and moving objects due to this execution time becomes a critical aspect, as certain video inpainting algorithms which do not deal with this aspect specifically may take days or even weeks to run [20].In general, video completion algorithms belong to either object based or patch based category. In case of Object-based algorithms the video usually separated into moving foreground and background that is either still or displays simple motion. These separated images sequences are then inpainted using separate algorithms. The background is often inpainted or completed using image inpainting techniques such as [6], whereas moving objects are often copied into the occlusion (damaged area) as smoothly as possible but such methods often include restrictive hypotheses on the moving objects' motion, Such as strict periodicity. A few object-based techniques include [11, 7, 8 and 9].

Patch-based methods are based on the idea of copying and pasting small video frame's patches (rectangular cuboids of video information) into the damaged regions. Wexler et al. in [10] described the first patch-based method to ensure temporal coherency in video inpainting .This is an iterative method, seen as a heuristic to solve a global optimization problem. The algorithm is work very slow due to high dimensionality of problem. It needs up to several days for a few seconds of VGA video. Patwardhan et al. [11] also use a patch-based approach. This is a greedy algorithm, and therefore cannot guarantee global coherency. The high synthesis quality of patch optimization methods requires more search iterations, which is the clear complexity obstruction in all of these methods. The search problem is even more critical in applications such as patch-based inpainting because the input image is typically much larger than the texture synthesis which usually take small image as input. Several technique for this search have been proposed, these techniques involves tree structures such as TSVQ [12], kd-trees [13] ,[10], [15] and VP-trees [16] each of which provide both exact and approximate search (ANN).

In case of synthesis applications, in many problems approximate search is used in combination with dimensionality reduction techniques such as PCA [13],[15],[17] because ANN methods are much more time- and memory-efficient in low dimensions. Ashikhmin [18] proposed a local propagation methods exploiting local

coherence in the synthesis process by restricting a patch search space to the source locations of its neighbors in the exemplar texture. Propagation search step of NNF used same coherence assumption. The k-coherence method [19] integrate the propagation methodology with a precomputation stage in which the k nearest neighbors of each patch are cached, and after that searches take advantage of these recomputed result sets. Although this speed up the search phase, in the input k-coherence still requires a full nearest-neighbor search for all pixels, and has only been demonstrated in the texture synthesis context. It assumes that the initial offsets are close enough that it needs to search only a small number of nearest neighbors. This may be true for small pure texture inputs, but it is found that for large complex images random search phase is required to escape local minima.

If speed and memory utilization of NNF algorithm compare against kd-trees with reduction in dimensionality, it show that it is at least an order of magnitude faster than the best competing combination (ANN+PCA) and uses significantly less memory. This algorithm also provides more generality than kd-trees because it can be applied with random distance metrics, and effortlessly modified to enable local interactions such as constrained completion. This NNF algorithm extracts undamaged information from the entire video sequence and finds reliable reference data to repair missing areas. In addition, experiment results demonstrate that NNF algorithm also significantly improves the temporal continuity of the final result.

3. AVI SPLITTER

It is a simple and easy to split AVI file. We can split a large AVI file into smaller AVI files, by using an AVI Splitter or to extract manually selected segments of an AVI file to new AVI files. For image processing Extraction of frame is always required. Extracted image from AVI video file is saved as BMP image. The BMP file format can store 2D digital images of arbitrary width, height, and resolution [14]. In general, AVI files contain multiple streams of different types of data. Most of the AVI sequences use both audio and video stream contents. For an AVI sequence, there is simple variation which uses video data and does not need an audio stream contents .A control track or MIDI track might include in specialized AVI sequence as an additional data stream. The AVI RIFF form use in AVI files. The four-character code AVI is used to identify AVI RIFF form. Two mandatory LIST chunks include in AVI files. These chunks define the format of the streams and stream contents. There might also include an index chunk in AVI files. The location of data chunks within the file specifies by this optional chunk.

4. SCRATCH DETECTION & MASKING ALGORITHM

4.1 Scratch detection algorithm

This scratch detection algorithm analyze each and every frame of input video after that it check RGB value of each pixel of every frame & when RGB value get as 0 then the frame from which RGB value get as 0 which frame is select as damaged frame.

4.2 Masking algorithm

4.2.1 Step1

First initialize mask as Boolean matrix. Initialize original image as pixel matrix, after that load masking & original image.

4.2.2 Step2

This mask matrix is initially taken a empty, take in loop it.. The algorithm read every pixel (x, y) coordinate value from original image matrix. Then extract RGB value of that pixel. If pixel's RGB value is 0 (means white color) then set Mask = T in the pixel (x, y) coordinate value in mask matrix.

4.2.3 Step3

After that, take in loop original image matrix. Read every pixel's (x, y) coordinate value. Take every pixel's value from original image matrix & compare it with same pixel's coordinate in mask matrix. If pixel's value in mask matrix is founded as Mask = T then change the RGB value of same pixel's coordinate in original image matrix for identification purpose that this area is mask & need to inpaint .[14]

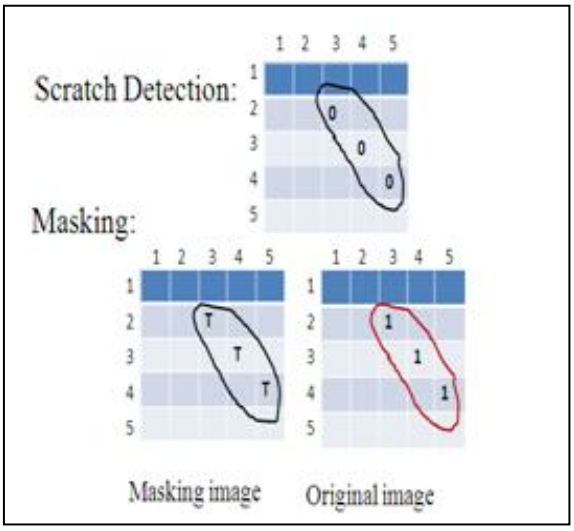


Fig 1: figure demonstrates the scratch detection and masking algorithm. In first matrix 0 value indicate the damaged pixels. In second Boolean matrix, for 0 values in first matrix true value set in Boolean matrix and in third original matrix damaged pixel 's RGB value set to red.

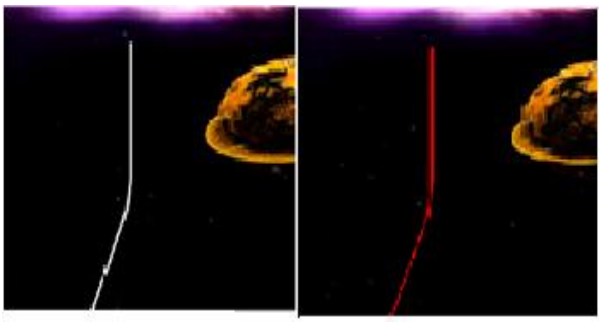
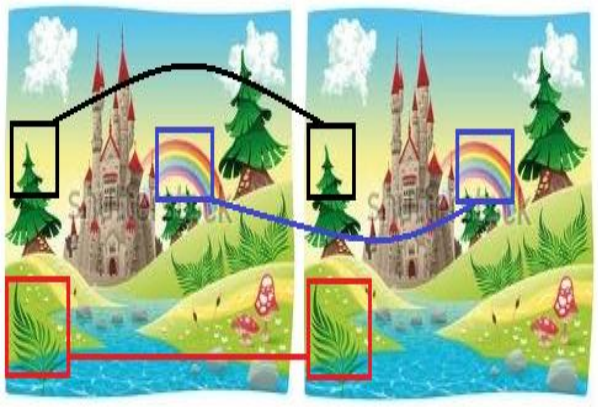


Fig 2: a) Damaged frame b) Masking of damage area

5. NEAREST NEIGHBOR ALGORITHM

Video Inpainting system relies upon the fast randomized algorithm for approximating nearest neighbor correspondence between two images. To understand the algorithm, we consider the main components of these methods: The vital element of nonparametric patch sampling methods is a repeated search of all patches in one image region for the most similar patch in another image region. Given images o A and B, every patch in A find the nearest neighbor in B under a patch distance metric such as Lp. We called this mapping the Nearest-Neighbor Field (NNF)which is illustrated in the figure 3.



(A) **(B)**

Fig 3: Given images A and B, for every patch in A find the Nearest neighbor in B and this mapping called as the Nearest-Neighbor Field (NNF).

This algorithm relies on three key observations about the problem for efficiently compute approximate nearest-neighbor fields:[22]

5.1 Dimensionality Offset Space

even if the dimensionality of the patch space is large (m dimensions), it is sparsely populated (O(M) patches). Many previous technique have speed up the nearest neighbor search by solving the dimensionality of the patch space using tree structures (e.g., kd-tree, which can search in O(mMlogM) time) and dimensionality reduction technique (e.g., PCA). In contrast, algorithm present in the paper searches in the 2-D space of possible patch offsets, which achieve enormous speed and memory efficiency.

5.2 Natural Structures Of Images

The e usual independent search for each pixel avoids the natural shape in images. The output typically contains large contiguous chunks of data from the input in patch-sampling synthesis algorithms (as noticed by Ashikhmin [2001]). Thus using NNF algorithm can improve efficiency by performing searches for adjacent pixels in an interdependent manner.

5.3 The Law Of Large Numbers

Any one random selection patch assignment is very unlikely to be a good guess, a few nontrivial fraction of a large field of random assignments will likely be good guesses. As this field increase larger, the chance that no patch will have a correct offset becomes small.

Based on these three observations randomized algorithm for computing approximate NNFs using incremental updates efficiently compute the NNF.[22]

The algorithm contains three steps: initialization, propagation and random search. The initialization is done by Randomly associating an ANN to each patch Wp, with p = (x;y). During propagation phase, the patches are sequentially scanned from low to high indices, first in the x dimension and then in the y dimension. For a given patch Wx;y, the algorithm considers the following ANNs W(x;y)+f(x-1;y) and W(x;y)+f(x;y-1). If one of these ANNs has a smaller patch distance with respect to W(x;y) than W(x;y)+f(x;y), then W(x;y)+f(x;y) is replaced with the best new ANN. The scanning order is reversed for the next iteration of the propagation (from high to low), and the algorithm test W(x;y)+f(x+1;y) and W(x;y)+f(x;y+1). In the two different scanning orderings, the vital point is obviously to use the neighbors which have already been tested in the current propagation step. The motivation for this step is the idea that objects in images are spatially coherent, and therefore that offsets which lead to good Approximate nearest neighbor for a given patch are likely to produce good ANNs in the proximity of this patch.[20]

5.4 Algorithm of NNF

- With random patch offsets pixels are Initialized.
- Neighbors are for better patch offsets.
- In concentric radius around the current offset search for better patch offsets.
- Go to Step no 2 until converge.[21]

5.4.1 Initialization

By assigning random values to the field ,or by using previous information nearest neighbor field can be initialized. When nearest neighbor field initializing with random offsets, Independent uniform samples use across the full range of image B. The algorithm can sometimes get trapped in suboptimal local minima if we use only this initial guess. To retain the quality of this prior but still preserve some ability to escape from such minima, we perform a few early iterations of the algorithm using a initialization randomly, then merge with only at patches where the up sampled initialization, where D is smaller, and then perform the remaining iterations.[22]

5.4.2 Iterative Update

After initialization, Iterative process of improving the NNF is performing. Each iteration of the algorithm go on as follows: In scan order these offsets are examined (from left to right, top to bottom), and each undergoes propagation followed by random search. At the patch level these operations are interleaved.[22]

5.4.3 Propagation.

We attempt to improve f (x; y) using the known offsets of f (x-1;y) and f (x;y-1), assuming that the offset of patch are likely to be the same. For e.g., if there is a good mapping at (x - 1; y), we try to use the translation of that mapping one pixel to the right for our mapping at (x; y). Let D (v) denote the patch distance (error) between the patch at (x; y) in A and patch (x; y)+v in B. We take the new value for f (x;y) to be the arg min of f D(f (x;y)), D(f (x-1;y)), D(f (x;y-1))g. The effect is that if (x; y) has a correct mapping and is in a coherent region R, then all of R which are below and to the right of (x; y) will be filled with the correct mapping.

Moreover, on even iterations it propagate information up and left by examining offsets in reverse scan order, using f (x+1;y)and f (x;y+1) as our candidate offsets. Random search. Let v0 = f (x; y). We attempt to improve f (x; y) by testing a sequence of candidate offsets at an exponentially decreasing distance from v0. [22]

5.4.4 Random search.

Let v0 = f (x;y). We attempt to improve f (x;y) by testing a sequence of candidate offsets at an exponentially decreasing distance from v0

ui = v0+waiRi …..(1)

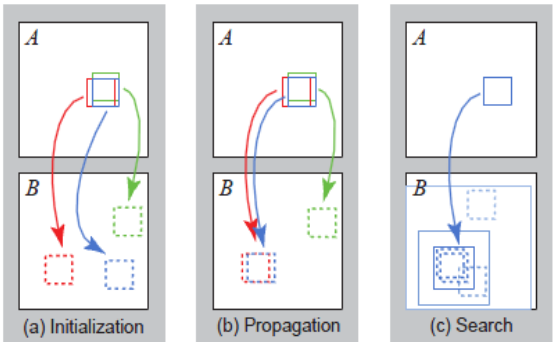


Fig 4 : The randomized NNF algorithm Phases:(a) patches initially have random assignments; (b) the blue patch checks above/green and left/red neighbors to see if they will improve the blue mapping, propagating better matches; (c) the patch searches randomly for improvements in concentric neighborhoods.[2]

Where in Eq.1.Ri is a uniform random in [-1;1] x [-1;1], w is a large maximum search "radius", and α is a fixed ratio between search window sizes. We examine patches for i = 0;1;2; ::: until the current search radius wa^i below 1 pixel. In our applications w is the maximum image dimension, and α= 1/2, except where noted. Note the search window must be clamped to the bounds of B.Consider the example of inpainting the missing region in given image shown in figure (b). Many algorithm address these problem by finding the similar patches in the image and filling the missing region with their context as shown in figure (C) and figure(d) . We make use of two key technique about this search for the similar patches first random guess for the correspondence are likely to be wrong most of the time which is shown in figure(e) But in sufficiently large region a few lucky guesses will be almost the correct correspondence as indicated in figure(f) Once we do find the good guess for any one patch it is likely that many nearby patches have similar correspondence and with this observation we begin with random initialization for patch correspondence and improve result as shown in figure(g).[23]

6. INPAINTING ALGORITHM

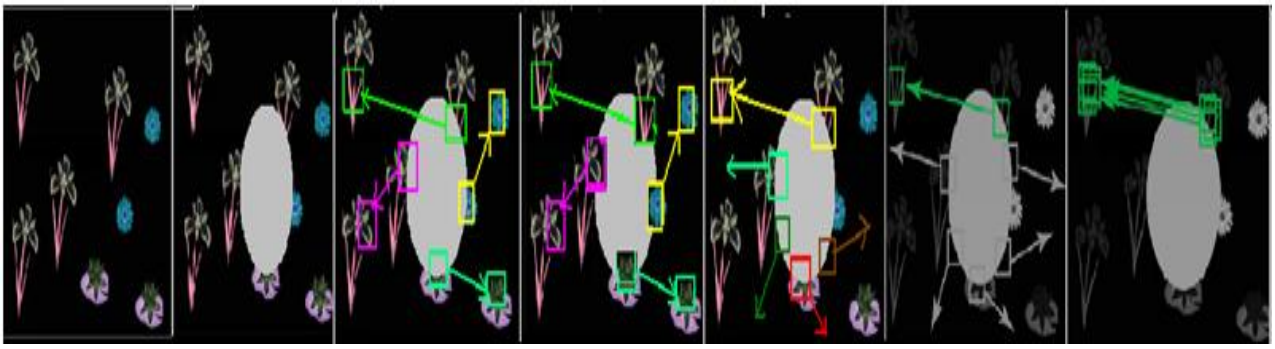
- input_image, mask & radius take as input.
- Create mask image.
- Initialize pyramid array.
- In loop take source width greater than radius & source height greater than radius. In this loop, assign down sampling to source. Add source to pyramid & increment pyramid level.
- Initialize level with maximum level.
- In loop, take level greater than zero. Check that if level is equal to maxlevel-1, if yes then load source image to target image, initialize target mask with false & find nearest neighbor by using source, target & radius.
- Return the target image[14]

7. RESULT AND FUTURE WORK

Figure 5 shows the sequential process of video inpainting. In figure 5.a white region indicate the damaged area in frame and Using the masking algorithm damaged part is detected and indicated with red color in figure 3.c.after that nearest neighbor algorithm used to find best match patch for damaged part .once that patch is found it is pasted on the damaged portion. In this way video inpainting process is completed. Some important points in this domain for future work are GPU implementations, additional speed gains may be realized, create an opportunity for new applications in real-time vision and video processing.

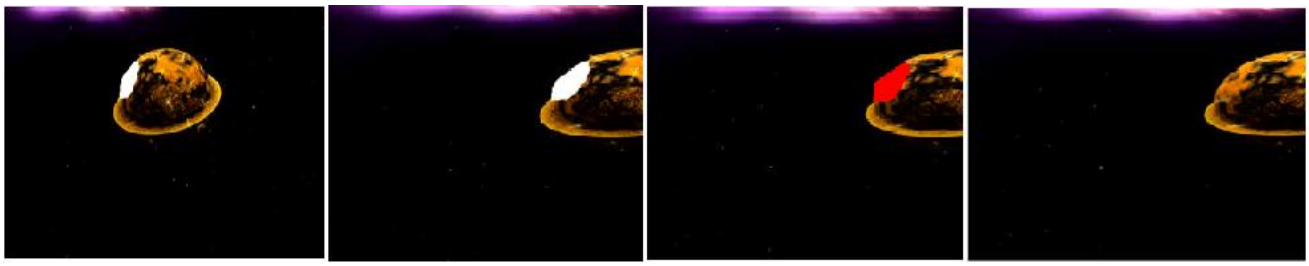
8. CONCLUSION

Video inpainting using NNF algorithm is sufficient for many practical applications like object removal, scratch repairing ,object reshuffling and repairing of old vintage videos. NNF algorithm avoids the costly Computations of the joint patch compatibility term and inference or optimization algorithms. The given algorithm does have some failure cases. Most notably, for pathological inputs, In addition, more edits to video frame can sometimes produce "feathering" artifacts where the algorithm simply can't escape a large local minimum basin. However, the speed of the algorithm makes it feasible to either introduce additional constraints or simply rerun the algorithm with a new random input to obtain a different solution. Although such repeated trials can be a burden with a slower algorithm. Nearest-neighbor algorithm use for inpainting the damaged area in video frame and it quickly computes the approximate nearest-neighbor fields between pairs of images .This algorithm increase the speed of matching nearest neighbor.



(a) (b) (c) (d) (e) (f) (g)

Fig 5:(a) shows original image (b)shows damaged image(c)shows similar patches correspondence (d) pasting of the patch on damaged area(e)random guess for patch correspondence (f)shows correct patch correspondence (g) shows that the nearby patches have similar



(a) Damaged frame . (b) Damage area indicated using white color (c) Masking of damaged area (d) Inpainting of damage area

Fig 6: Shows the result of video inpainting process using masking and nearest neighbor algorithm.

9. ACKNOWLEDGMENT

We would like to thank Prof. Mr.D.M. Deshpande for his Guidelines in making this paper.

REFERENCES

[1] Video Inpainting on Digitized Vintage Films via Maintaining Spatiotemporal Continuity Nick C. Tang, Chiou-Ting Hsu, Member, IEEE, Chih-Wen Su, Timothy K. Shih, Senior Member, IEEE, and Hong-Yuan Mark Liao, Senior Member, IEEE

[2] A fast video inpainting technique Mrinmoy Ghorai, Pulak Purkait, and Bhabatosh Chanda Indian Statistical Institute Kolkata, India

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester,"Image inpainting," SIGGRAPH, pp. 417–424, 2000.

[4] S. Masnou and J.-M. Morel, "Level lines based dissoclusion," ICIP, vol. 3, pp. 259–263, 1998.

[5] S. Masnou and J.-M. Morel, "Disocclusion: a variationalhal-00838927, version 1 - 27 Jun 2013approach using level lines," IEEE Trans. Image Processing,vol. 11, no. 2, 2002.

[6] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," CVPR, pp. 721–728, 2003.

[7] J. Jia, Y-W. Tai, T-P. Wu, and C-K. Tang, "Video repairing under variable illumination using cyclic motions," PAMI, vol.28, no. 5, pp. 832–839, 2006.

[8] M. V. Venkatesh, S-C. S. Cheung, and J. Zhao, "Efficient object based video inpainting," ICIP, vol. 30, pp. 168–179,2006.

[9] C-H. Ling, C-W. Lin, C-W. Su, Y-S. Chen, and H-Y. Liao, "Virtual contour guided video object inpainting using posture mapping and retrieval," IEEE Trans. Multimedia, vol. 13, no. 2, pp. 292–302, 2011.

[10] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," CVPR, vol. 1, pp. 120–127, 2004.

[11] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Videoinpainting of occluding and occluded objects," ICIP, vol.2,pp. 69–72, 2005.

[12] WEI, L. Y., AND LEVOY, M. 2000. Fast texture synthesis using tree structured vector quantization. In ACM SIGGRAPH, 479–488.

[13] HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image analogies. In ACM SIGGRAPH, 327–340.

[14] VIDEO INPAINTING Chayanica Das1, Sachet Sharma2, Gaurav Mishra3JSPM's Wagholi ,ICOER(Computer Engineering, Pune, India) 1, 2,PatchMatch - A Randomized Correspondence Algorithm for Structural Image Editing

[15] KOPF, J., FU, C.-W., COHEN-OR, D., DEUSSEN, O., LISCHINSKI, D.,AND WONG, T.-T. 2007. Solid texture synthesis from 2d exemplars.ACM SIGGRAPH 26, 3, 2:1–2:9..

[16] KUMAR, N., ZHANG, L., AND NAYAR, S. K. 2008. What is a good nearest neighbors algorithm for finding similar patches in images? In European Conference on Computer Vision (ECCV), II: 364–378.

[17] LEFEBVRE, S., AND HOPPE, H. 2005. Parallel controllable texture synthesis. ACM SIGGRAPH 24, 3, 777–78

[18] ASHIKHMIN, M. 2001. Synthesizing natural textures. In I3D '06 Proc., ACM, 217–226

[19] TONG, X., ZHANG, J., LIU, L., WANG, X., GUO, B., AND SHUM, H.-Y. 2002. Synthesis of bidirectional texture functions on arbitrary surfaces. ACM SIGGRAPH 21, 3 (July), 665–672.

[20] Towards fast, generic video inpainting ,Alasdair Newson, Matthieu Fradet,Patrick Pérez ,Andrés Almansa, Yann Gousseau

[21] PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing Connelly Barnes Eli Shechtman Adam Finkelstein Dan B,CS 294-69- Paper Presentation.

[22] PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing Connelly Barnes1 Eli Shechtman2;3 Adam Finkelstein1 Dan B Goldman2 1Princeton University 2 Adobe Systems 3 University of Washington

[23] http://www.youtube.com/watch?v=fMe19oTz6vk(ceffect sstudio)