ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

VIDEO INPAINTING

Chayanica Das[1], Sachet Sharma[2], Gaurav Mishra[3]

JSPM's Wagholi ,ICOER(Computer Engineering, Pune, India) [1, 2, 3]

Abstract: This Video inpainting is an important video enhancement technique used to facilitate the repair or editing of digital videos. In this paper, we propose a video inpainting algorithm for repairing damaged content in digitized video films, focusing on maintaining good spatiotemporal continuity. The proposed algorithm utilizes key techniques.

Keywords: Frame Processor, Extraction, Masking, AVI splitter

I. INTRODUCTION

In recent years, transforming cultural and historical artifacts such as photographs and vintage films/videos into digital format has become an important trend. However, because of their age, the visual quality of such images and videos after digitization is usually very poor and often contain unstable luminance and damaged content.

It is the most desirable techniques which helps to repair the content .Video inpainting is the method by which the noisy or damaged frames are extracted from the video and those frames are replaced by the new frames. Thus video inpainting removes the damaged frame from the video and finally gives the good quality video as output.

II. AVI SPLITTER

It is an easy way to split AVI file. AVI Splitter allows you to split a large AVI file into smaller AVI files, or to extract manually selected segments of an AVI file to new AVI files.

A. Image Extraction

Extraction of frame is always required when it comes to image processing.It is an key feature of AVI splitter that it can select segment of AVI video file and extract it into smaller AVI video file

Extracted image is saved as BMP image. The BMP file format is capable of storing 2D digital images of arbitrary width, height, and resolution

B.AVI Key Features

1) It splits large AVI video file into equal-size parts AVI video.

2)It selects segment of AVI video file and extract it into smaller AVI video file.

3)It captures a frame of AVI video file and export it as an image file. The image can be bmp, gif, jpg, tif,png.

4)Can export AVI video with specified video frame size, video frame rate, video codec and audio format.

III. NEAREST NEIGHBOUR ALGORITHM

The core of our system is the algorithm for computing patch correspondences. We define a nearest-neighbor field (NNF) as a function f : A 7-> R^2 of offsets, defined over all possible patch coordinates (locations of patch centers) in image A, for some distance function of two patches D.

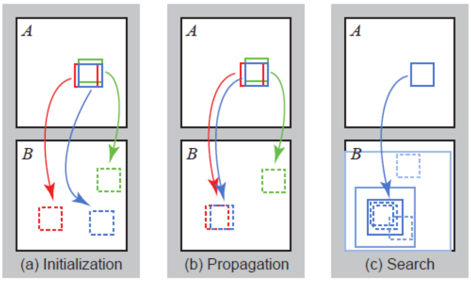
Given patch coordinate a in image A and its corresponding nearest neighbor b in image B, f(a) is simply b-a. We refer to the values of *f*as offsets, and they are stored in an array whose dimensions are those of A

A. Randomized Algorithm

This section presents a randomized algorithm for computing an approximate NNF. As a reminder, the key insights that motivate this algorithm are that we search in the space of possible offsets, that adjacent offsets search cooperatively, and that even a random offset is likely to be a good guess for many patches over a large image.

B. Iterative Update

Next, an iterative update process is applied to the NNF, in which good patch offsets are propagated to adjacent pixels, followed by random search in the neighborhood of the best offset found so far.



C. Iitialising

The nearest-neighbor field can be initialized either by assigning random values to the field, or by using prior information. When initializing with random offsets, we use independent uniform samples across the full range of image B.We use a coarse-to-fine gradual resizing process, so we have the option to use an initial guess upscaled from the previous level in the pyramid. However, if we use only this initial guess, the algorithm can sometimes get trapped in suboptimal local minima.

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 random initialization, then merge with the upsampled initialization only at patches where D is smaller, and then perform the remaining iterations.

D. 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 patch offsets are likely to be the same. For example, 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 fD(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 below and to the right of (x;y) will be filled with the correct mapping. Moreover, on even iterations we 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 v_0 = f (x;y). We attempt to improve f (x;y) by testing a sequence of candidate offsets at an exponentially decreasing distance from v_0:

$$\mathbf{u}_i = \mathbf{v}_0 + w\alpha^i \mathbf{R}_i$$

(1)

where Ri is a uniform random in [-1;1] x [-1;1], w is a large maximum search “radius”, and $\boldsymbol{\alpha}$ is a fixed ratio between search window sizes. We examine patches for i = 0;1;2; ::: until the current search radius $w\boldsymbol{\alpha}^i$ is below 1 pixel. In our applications w is the maximum image dimension, and $\boldsymbol{\alpha}$= 1/2, except where noted.Note the search window must be clamped to the bounds of B.

IV. MASKING OF IMAGE

Masking image is calculated as a form of Boolean matrix and mask it to the original

```
initialize mask as boolean matrix........1
initialize origimg as pixel marix..........2
load maskimg and origimg.................3
loop maskimg ...................................4
a.              read mask pixel (x,y)
b.              extract for rgb
c.              if  rgb == 0xffffffff
i.                       assign x,y to mask=true
ii.                      end if
end loop
loop origimg
d.              read pixel x,y
e.              if mask(x,y)==true
i.                       set origimg(x,y)= 0xffff0000
end if
     end loop
     display masked image
```

V. INPAINTING

```
INPUT IMG, MASK,RADIUS..............................1

CREATE MASK IMAGE...................................2

INITIALIZE PYRAMID ARRAY..........................3

LOOP SRC WIDTH>RADIUS AND SRC HEIGHT>RADIUS
src=downsampling......................................4
add src to pyramid.......................................5
increment pyramid level...............................6
end loop

INITIALIZE LEVEL=MAXLEVEL

FOR LOOP LEVEL>0
if level=maxlevel-1.......................................7
targetimg=sourceimg............................8
initialize targetmask=false....................9
find nearestneighbour(src,target,radius)
else
find nearestneighbour(src,target,radius)
endif
expectationmaximum(level)
          end forloop

RETURN TARGET IMG.....................................10
```

video is always rebuilt after recovering the image.

VI. CONCLUSION

We have proposed a novel video inpainting algorithm for digitized aged films. The algorithm consists frame completion. In addition, a preprocessing procedure constructs a motion map to record the motion information in undamaged source areas. The motion completion procedure restores the motion in each missing area based on the completion order determined by the priority computation step. The completed motion map is used to improve the temporal continuity and find the best-matched result for inpainting damaged areas. The frame completion procedure seamlessly repairs all the damaged areas and reduces the intensity of video flicker. During the frame completion phase, we use a panoramic mosaic to help stabilize the global and local luminance and thereby obtain better restored videos.

ACKNOWLEDGMENT

We would like to thank Prof. Mrs. Bharti for her guidelines in making this paper.

REFERENCES

[1] en.wikipedia.org/wiki/Remote_Installation_Services

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, “Hierarchical model-based motion estimation,” in *Proc. 2nd Eur. Conf. Computer Vision*, 1992, pp. 237–252.

[3]] A. M. Huang and T. Nguyen, “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation,” IEEE Trans. Image Process., vol. 18, no. 4, pp. 740–752, Apr. 2009.

[4] Y. Shen, F. Lu, X. -C. Cao, and H. Foroosh, “Video completion for perspective camera under constrained motion,” in Proc. Int. Conf. Pattern Recognition, 2006, pp. 63–66.

K. M. Gullu, O. Urhan, and S. Erturk, Scratch detection via temporal.