International Journal of Innovation and Scientific Research
ISSN 2351-8014 Vol. 23 No. 2 May 2016, pp. 372-379
© 2015 Innovative Space of Scientific Research Journals
http://www.ijisr.issr-journals.org/

An Innovative approach of EM Algorithm for Restoration of Noisy Video Frame Images in a Video Sequence

A. Shenbagarajan[1], P. Elamparithi[1], and C. Karuppasamy[2]

[1]Department of CSE, AAA College of Engineering & Technology, Sivakasi, Tamilnadu India

[2]Department of EEE, AAA College of Engineering & Technology, Sivakasi, Tamilnadu India

ABSTRACT: In this paper, the method was proposed a solution for the problem of an image inpainting method for missing parts or corrupted by noise of a video sequence recorded by a moving or stationary camera. The region to be inpainted may be still or moving, in the background or in the foreground, it may occlude one object or may be occluded by some other objects. This method was approached by a simple preprocessing stage and two steps of video inpainting. In the preprocessing stage, the corrupted video sequence images is extracted into multiple frames, then roughly segment each frame into foreground and background using Expectation Maximization algorithm. In segmentation, it builds three image mosaics that help to produce time consistent results and also improve the performance of the algorithm by reducing the search space. In the first video inpainting step, it reconstructs the corrupted video images of the moving objects in the foreground that are occluded. At the end of this first video inpainting, fill the gap as much as possible by copying information from the moving foreground in other frames, using a priority-based scheme. In the second step, the remaining regions are inpainted with the background. To accomplish this, first align the frames and directly copy when possible. The remaining pixels are filled in by extending spatial texture synthesis techniques to the spatiotemporal domain. This proposed framework has several advantages such as, it is simple to implement, fast and does not require statistical models of background or foreground. Works well in the presence of rich and cluttered backgrounds.

KEYWORDS: Camera motion, Expectation-Maximization (EM), optical flow, texture synthesis, video inpainting, digital image, Partial Differential Equations (PDEs), video sequence, blurring operator, space variant.

1 INTRODUCTION

Image Restoration is the process of obtaining the original image from the degraded image given the knowledge of the degrading factors. There are a variety of reasons that could cause degradation of an image and image restoration is one of the key fields in today's digital image processing due to wide area of application. Commonly occurring degradations include blurring, motion and noise. [1]The general model for image degradation phenomenon is given as y = Hf + n, where y is the observed blurred and noisy image, f is the original image, n is additive random noise and H is the blurring operator. The main objective is to estimate the original image from the observed degraded image. Whatever the degraded process, image distortions can fall into two categories, namely, spatially invariant or space invariant and spatially variant or space variant. [3]In a space invariant distortion all pixels have suffered the same form of distortion. This is generally caused by problems with the imaging system such as distortions in optical system, global lack of focus, or camera motion. In a space variant distortion, the degradation suffered by a pixel in the image depends upon its location in the image. This can be caused by internal factors, such as distortions in the optical system, or by external factors, such as object motion.

The problem of automatic video restoration in general and automatic object removal and modification in particular, is beginning to attract the attention of many researchers. This proposed method is able to inpaint objects that move in any fashion but do not change size. A number of algorithms for automatic still image completion have been proposed to restore the video sequence corrupted or occluded images, but these cannot be generalized in a straightforward manner to address the problem of the given video sequences to be restored. [5]There has also been some preliminary work on frame-by-frame Partial Differential Equations (PDEs) based video inpainting. The PDE is applied spatially and completes the video frame-by-frame. This does not take into account the temporal information that a video provides and its application is thereby limited. Also, the PDEs based methods interpolate edges in a smooth manner, but temporal edges are often more abrupt than spatial edges. [2] In spacetime completion of damaged areas in a video sequence paper, they proposed the problem of video completion as a global optimization problem, which is inherently computationally very expensive. The work extends to space time the pioneering technique of nonparametric sampling developed for still images by [4] Efros and Leung in the "Texture synthesis by non-parametric Sampling". This implies the assumption that objects move in a periodic manner and also they do not significantly change scale, because otherwise the "copy and paste" approach of this "Texture Synthesis by non parametric sampling" would fail. Although the results are good, they suffer from several shortcomings. Only low-resolution videos are shown and over smoothing are often observed. This is due in part to the fact that pixels are synthesized by a weighted average of the best candidates and this averaging produces blurring. Also, the camera is always static in all the examples in that study. Though the reason for this is not discussed, it is probably due to the fact that the authors use a very simple motion estimation procedure involving the temporal derivative. [9] An interesting probabilistic video modeling technique has been proposed in video epitomes, with application to video inpainting. Epitomes as patch based probability models that are learnt by compiling together a large number of examples of patches from input images. These epitomes are used to synthesize data in the areas of video damage or object removal.

Key contribution: Our approach is fundamentally related to the nonparametric sampling method proposed in "Texture Synthesis by non-parametric sampling" for the problem of 2-D texture synthesis. [6] [7] This method was further improved upon by using a priority and confidence based synthesis in "Region filling and object removal by exemplar-based inpainting", [10] This technique was adapted and extended for video inpainting for the static camera case in "Video inpainting of occluding and occluded objects", This proposed method introduced foreground, background and optical- flow mosaics, which not only help to produce good quality results, but also reduce the search space and lead to a faster implementation. Although the copy and synthesis components of the proposed framework are basically 2-D, the whole search and metric distances fully exploit the spatiotemporal.

2 Material And Methods

Our proposed method consists of three stages preprocessing, motion inpainting and background filling. This method needs several assumptions on the kind of video sequences that are able to restore. As mentioned below, these assumptions are implicitly or explicitly shared by most state of the art works on the subject, often in an even more restrictive fashion. These assumptions are used to compute a rough motion confidence mask "Mc" for each frame by comparing it with the following frame using block-matching. In preprocessing, the given video sequence is segmented into background and moving foreground, then Using Expectation maximization algorithm the background was subtracted. Video inpainting consists of motion inpainting and foreground inpainting. At end of this method perform background filling to restore the original video sequence. The Block diagram of this method was shown in Fig. 1. This study consists of the following methods:

- Basic assumptions
- Preprocessing
- Motion Inpainting
- Background filling

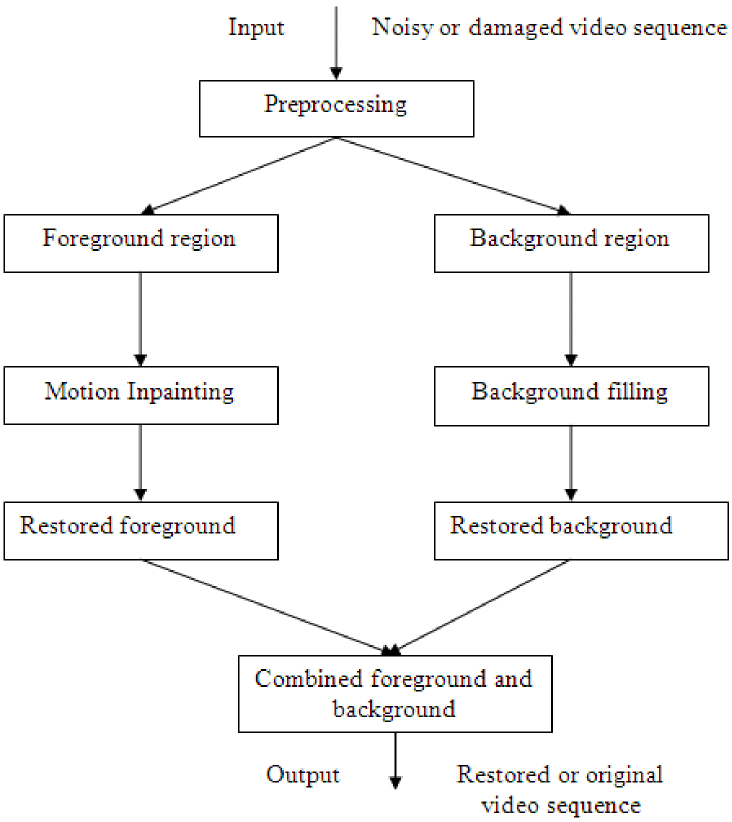

Fig. 1: Block diagram of the proposed method

Basic assumptions: The following are the basic assumptions made to restore the video sequence:

- The scene essentially consists of stationary background with some moving foreground
- Camera motion is approximately parallel to the plane of image projection. This restriction ensures that background objects will not (significantly) change size, allowing for texture synthesis in the spirit, which cannot deal with changes in size nor perspective
- Moving objects do not significantly change size. Again, this restriction is imposed by the use of the non-parametric texture synthesis. This constraint can be removed by using a multi-scale matching algorithm which can address the change in size when the object moves away from or towards the camera
- Foreground objects move in a repetitive fashion. In order to recover occluded or damaged foreground and without the use of probabilistic models or libraries, the vital information must be present in the video itself. Hence this “periodicity” assumption

All the examples in this study are taken with a hand-held camera, thereby complying with these assumptions only partially, while still producing very satisfactory results.

Preprocessing: In this method the first stage is doing some processing over the given video sequence before applying to the video inpainting process. The simple assumptions allow us to compute a rough motion confidence mask “Mc” for each frame just by comparing it with the following frame using block-matching. The preprocessing steps are given below:

Step 1: Segment each frame into static background and moving foreground.
Step 2: Background Subtraction.

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. Background Subtraction done by Expectation-Maximization (EM) algorithm. It will separate the video into background and foreground.

Expectation Maximization (EM) algorithm: EM algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an Expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed and Maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step and the process is repeated.

Let the observed variables be known as y and the latent variables as *z*. Together, y and z form the complete data. Assume that p is a joint model of the complete data with parameters θ: p(y,z | θ). An EM algorithm will then iteratively improve an initial estimate θ_0 and construct new estimates θ_1 through θ_N. An individual re-estimation step that derives θ_{n+1} from θ_n takes the following form (shown for the discrete case; the continuous case is similar):

$$\theta_{n+1} = \arg\max_{\theta} \sum_{z} p(z \mid y, \theta_n) \log p(y, z \mid \theta)$$

In other words, θ_{n+1} is the value that Maximizes (M) the Expectation (E) of the complete data log-likelihood with respect to the conditional distribution of the latent data under the previous parameter value. This expectation is usually denoted as Q (θ):

$$Q(\theta) = \sum_{z} p(z \mid y, \theta_n) \log p(y, z \mid \theta)$$

Speaking of an Expectation (E) step is a bit of a misnomer. What is calculated in the first step are the fixed, data-dependent parameters of the function Q. Once the parameters of Q are known, it is fully determined and is Maximized in the second (M) step of an EM algorithm It can be shown that EM iteration does not decrease the observed data likelihood function and that the only stationary points of the iteration are the stationary points of the observed data likelihood function. In practice, this means that an EM algorithm will converge to a local maximum of the observed data likelihood function. EM is particularly useful when maximum likelihood estimation of a complete data model is easy. If closed-form estimators exist, the M step is often trivial. A classic example is maximum likelihood estimation of a finite mixture of Gaussians, where each component of the mixture can be estimated trivially if the mixing distribution is known. Expectation-maximization" is a description of a class of related algorithms, not a specific algorithm; EM is a recipe or meta-algorithm which is used to devise particular algorithms. The Baum-Welch algorithm is an example of an EM algorithm applied to hidden Markov models. Another example is the EM algorithm for fitting a mixture density model. An EM algorithm can also find Maximum A Posteriori (MAP) estimates, by performing MAP estimation in the M step, rather than maximum likelihood. There are other methods for finding maximum likelihood estimates, such as gradient descent, conjugate gradients or variations of the Gauss-Newton method. Unlike EM, such methods typically require the evaluation of first and/or second derivatives of the likelihood function.

Mosaic: A Mosaic is a panoramic image obtained by stitching a number of frames together. In the pre-processing stage this method build three mosaics: a background mosaic, a foreground mosaic and an optical flow mosaic. The computation of motion confidence mask “Mc” gives us a segmentation of the sequence into foreground and background layers, as well as a good estimate of the camera shift for each frame. The optical flow mosaic, which contains data, used for the Sum of Squared Difference (SSD) computations as shown below, this method use a 2-channel image to store the horizontal and vertical components of the residual optical flow, that is, the motion vectors from which subtracted the camera shift. In Fig. 2 shows the color coding to represent the direction of this 2D vectors: green tones indicate horizontal motion and red tones indicate vertical motion.

Motion inpainting: This consists of two stages to inpaint the damaged frame (i) Motion inpainting (ii) foreground filling, The following steps describe the motion inpainting process.

Step 1: Finding the highest priority pixel: For any given pixel P, its priority Pr(P) is the product of two terms: a confidence term C(P) and a data term D(P): Pr(P) = C(P)D(P). The confidence term C(P) is proportional to the number of undamaged and reliable pixels surrounding P. The data term D(P) is high if there is an image edge arriving at P.

It is important to note that data from the mosaics is not used to fill-in the damaged frames. The mosaics are only used to search for the “candidate-undamaged-frames”, from where the information copy into the damaged frames:

$$D(k) = \frac{|(\nabla M_c \perp)_k . n_k|}{\alpha}$$

Where:

α = A normalizing constant (usually 255)
nk = The normal to the hole-boundary. The inner product of the rotated gradient of Mc, $(\nabla M_c \perp)$ and the normal
nk = Computed using central differences

Step 2: To calculate candidate frame: Candidate frames, i.e., a small subset of frames where the information will look for the best match. This “initial guess search” is implemented using the following steps:

- In the current damaged frame under consideration, find the highest priority location P and its surrounding patch ψ_p
- Using the already available camera shifts are computed during the pre-processing step, find the corresponding location Pm for P and also its surrounding patch ($\psi_{p.}$) in the foreground mosaic
- Using ($\psi_{p.}$) as a template perform a search in the foreground mosaic to find the matching patch (es) ($\psi_{p....}$)
- Now, using the camera shifts and the motion confidence masks for each frame, identify the frames that have motion at the location corresponding to the mosaic area specified by the matching patch (es) ($\psi_{p....}$) These frames are the candidate frames for searching a matching patch for the highest priority location P in the current frame

Copying: Candidate frame be found, then the main process of inpainting can be perform, Then search each candidate frame for a best matching patch ψ_{qm}, the patch with minimum distance to our target patch ($\psi_{p.}$) Once the matching patch ψ_q is found, instead of fully copying it onto the target ψ_p, then do the following. Look at Mc(motion confidence mask)and copy from ψ_q only the pixels that correspond to the moving foreground. The remaining unfilled pixels of ($\psi_{p.}$) must correspond to the background, so don’t want to fill them at this Motion Inpainting stage. For this reason we mark them to have zero priority (i.e., “disable” them from any future motion-filling-in).

Fig. 2: optical flow mosaic

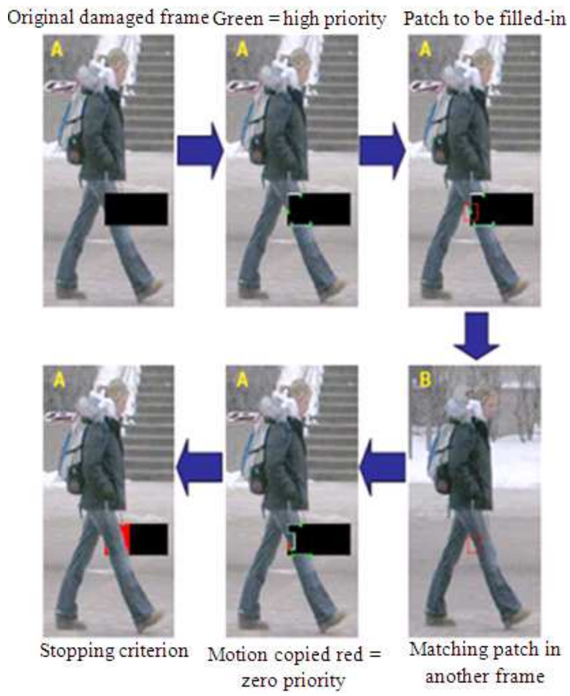

Fig. 3: Motion inpainting

The separation of background and foreground is essential if the background is rich and inhomogeneous. If copied the whole patch ψ_q instead of only its foreground pixels, it would be assuming that whenever foreground matches foreground, their surrounding background matches as well. Such an assumption would imply that the background is more or less the same all along the trajectory of the moving foreground object(s).

Update: After inpainting ψ_p, the Mc values at ψ_p are updated to the Mc values at ψ_q. Next update the confidence C(p) at each newly inpainted pixel as follows:

$$C(p) = \frac{\sum q \in \psi_{p \cap (Mc \setminus \Omega} C(q)}{|\psi_p|}$$

Where:

ψ_p = The area of the patch
Ω = The region of inpainting

The Fig. 3 shows the motion inpainting schemes, in fig Green dots indicate highest priority, Red squares indicate the patch to be inpainted (frame A) and the corresponding best match (frame B). Areas filled with red are constrained to have zero priority.

Foreground inpainting: In this foreground inpainting, repeat the above steps (find candidate frame, copying and update) until all the pixels in the inpainting area are either filled-in or have zero priority for motion inpainting (i.e., are "disabled" as explained above). This is precisely our indication that moving objects have been fully inpainted in the current frame. Now repeat this process for all the frames that require motion inpainting. This gives us a sequence with only moving objects filled-in and the rest of the missing region needs to be filled-in with background.

Background filling: After finished the stage of Motion Inpainting, this method enter the stage where inpaint the background. To accomplish this first align the frames and directly copy whenever possible, while the remaining pixels are filled-in by extending spatial texture synthesis techniques.

Texture synthesis: In image processing, every digital image composed of repeated elements is called texture. Texture Synthesis here comes to help restore these larger regions of damage. The inpainting algorithm is still valuable since it performs quite well for small scratches and runs relatively fast. The texture synthesis method comes with extra overhead and runs in time proportional to the size of the image (not the mask). It is, however, categorically better in just about all tests tried (with inpainting performing better in only rather artificial setups designed to highlight its abilities The algorithm can effectively be broken into 2 stages:

- Calculating the information content of pixels in relation to neighboring pixels (pixels will be chosen based on this weighting)
- Finding a pixel with matching neighbors by evaluating some distance measure

This algorithm goes like this, while there exist undetermined pixels:

- Choose the highest priority one based on weighting in calculating the information content of pixels in relation to neighboring pixels
- Choose the closest approximation by distance measure in finding a pixel with matching neighbors by evaluating some distance measure
- Color the pixel and update the weightings of neighbors

Calculating weighting: The first stage of the repair process (which incurs the time proportional to image size cost) loops through all offsets of a pixel within some set radius. For each offset, loop through all possible base pixels (pixels we're looking at the offsets for-entire image minus some radius sized boundary) and look at the most significant bits of that base pixel. The most significant bits from each channel (for example, RGB) are grouped together so that it can consider them together. Then decide to split the distribution of pixel values at the offset pixels into partitions based on the most significant bits of the base pixel. How many partitions to create is decided using some information theory justification. First, guess that the distribution is unimodal, then want to find the number of bins that would allow us to encode the pixel value distribution using the least number of bits. This corresponds to choosing a high information content for the pixels in relation to their neighbors. Once this number of partitions is calculated, it can assign weightings based on the information content of an offset pixel and normalize to reduce the reproduction of the same highly weighted pixels.

Choosing a match: The distance function is simply the sum of the absolute value of differences in each channel. So the neighborhood of known pixels around a base pixel is compared (using any searching) against other neighborhoods to choose a match. The distance function is subject to various weighting parameters as well. A random channel is also used to add some seeming non-determinism to the process.

[Distance is measure d (2nd step of the algorithm), Sum up flows from neighboring pixels, weighting each term with the strength of flow (found by taking dot product of flow direction with neighbor vector)]. Then, after the sum is found, force each neighbor pixel to be that averaged color achieved from the mixture of all incoming flows. That way a pixel would have to be pretty whatever that color is. Unfortunately, this method allows noise or single pixels that differ by a lot to dominate the process, getting copied multiple times as the flow progresses. So these distinct pixels get too great a share. In the inpainting algorithm, the anisotropic diffusion mitigated this. Considered input with high resolution video sequence with multiple frames.

3 DISCUSSION

In this study, the expected result is restore the damaged video frames. In edge based Image restoration approach the restoration resolution up to 84.22 is obtained for damaged images. In fuzzy logic restoration percentage is 89.73. In EM algorithm the accuracy was 96 percentages and computation time was very high with good resolution. Figure 4 shows expecting result.

(a)

(b)

(c)

(d)

Fig. 4: Expecting result (a) original sequence with a moving occluder (b) sequence with occluder removed (c) Moving person is filled (d) Area of occlusion is filled

4 CONCLUSION

In this study, the model presented a simple framework for filling the missing parts in the video sequences using EM algorithm. The static background filling is consistent throughout the video sequence. Moving foreground filling is independent of changing background from one frame to another frame. This method is combined of Motion based Inpainting with spatial Inpainitng. Obtaing three image mosaics in the preprocessing process which is used to fill the information in the corrupted or damaged regions in the foreground and background. If there are moving objects to be restored, they are filled-in first, independently of the changing background from one frame to another. Then the background is filled-in by extending spatial texture synthesis techniques to the spatio-temporal domain.

REFERENCES

[1] Anguelov, D., P. Srinivasan, D. Koller, S. Thrun and J. Rodgers *et al*., 2005. Scape: Shape completion and animation of people. Presented at the ACM Siggraph.

[2] Baker, S., R. Szeliski and P. Anandan, 1998. A layered approach to stereo reconstruction. Comput. Vis. Patt. Recog., 434.

[3] Ballester, C., V. Caselles and J. Verdera, 2003. Dissoclusion by joint interpolation of vector fields and gray levels. SIAM Multiscale Model. Simul., 2: 80-123.

[4] Bankam, M.R. and A.K. Katsaggelos, 1997. Digital image restoration. Presented IEEE signal Process. Maga.

[5] Bertozzi, A.L., S.E. Glu and A. Gillette, 2007. Inpainting of binary images using the cahn-hilliard equation. Presented Image Process., 16.

[6] Bornard, R., E. Lecan, L. Laborelli and J.H. Chenot, 0000. Missing data correction in still images and image sequences.

[7] Oliveira, M.M., B. Bowen, R. McKenna and Y.-S. Chang, 2001. Fast digital image inpainting. Presented at the International Conference on Visualization, Imaging and Image Processing (VIIP), Sept. 3-5. Marbella, Spain.

[8] Patwardhan, K.A., G. Sapiro and M. Bertalmío, 2007. Video inpainting under constrained camera motion. Presented at the IEEE Conference. Image Processing.

[9] Wang, H., H. Li and B. Li, 2007. Video inpainitng for largely occluded moving human. IEEE.

[10] Zhou, C. and S. Lin, 2007. Removal of image artifacts due to sensor dust. Presented at the Microsoft Research Asia in IEEE.