A BAYESIAN FRAMEWORK FOR RECURSIVE OBJECT REMOVAL IN MOVIE POST-PRODUCTION

Anil Kokaram

Electronic and Electrical Engineering Department,
University of Dublin, Trinity College, Ireland

Bill Collis and Simon Robinson[†]

The Foundry, London UK
www.thefoundry.co.uk

ABSTRACT

Some of the most convincing film and video effects are created in digital post-production by removing apparatus that supports or manipulates actors and objects. Wires and people, for instance, can be removed by digitally painting them out of the scene provided some 'clean plate' image is available for pasting in the missing regions. This paper addresses the problem when no such plate is available. Object removal requires the estimation of the motion of the hidden material and then the reconstruction of the missing image data. Using the notion of temporal motion smoothness, this paper articulates the two problems using a Bayesian framework and so develops a unique tool for automated object removal. The tool is currently being tested in the film effects industry and initial feedback is very positive.

1. INTRODUCTION

Some of the most convincing film and video effects are created in digital post-production by removing apparatus that manipulates actors and objects. The undesired apparatus e.g. wires, cranes; will be termed *rig*s in the rest of this paper. Of course some of the undesired 'rig' material may also be objects in the scene itself, for example: undesired people. A simple procedure for removing the undesired apparatus is to generate a 'clean plate' image containing only the background image data for instance. That data can then be pasted into the region covered by the rig. However, arranging clean plate image capture can be a tedious exercise outside a studio and it is useful to consider whether it is possible to remove rigs without the need for a clean plate. It is assumed here that the user has roughly outlined the region to be reconstructed in each frame.

Figure 3 shows a sequence in which the rig to be removed is delineated by a red overlay. As the rig traverses the scene, it uncovers and reveals image material. Intuitively then, it would seem sensible that the removal of the rig can be achieved by collating the uncovered and revealed data throughout the sequence to reconstruct the image in the region of the rig. This is only possible because the rig is *moving*. If it were stationary, then the problem of reconstructing the hidden image is one of image regeneration, or image synthesis. In this latter case, the methods of Efros et al and Bertalmio et al [1, 2] would be more suitable, although the success of those techniques would depend on the *scale* of the underlying 'texture'.

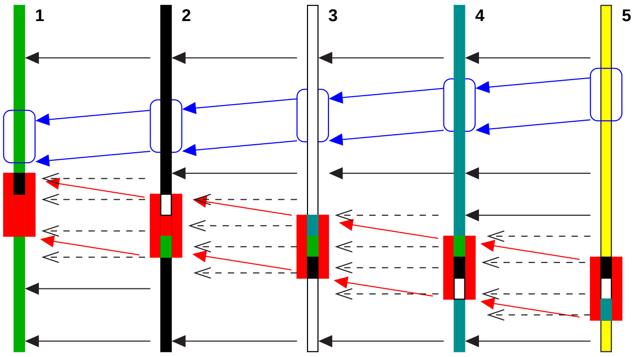

Fig. 1. A view of five frames with two objects showing simple motion. The motion in the area of the rig (red) is to be removed. The ideal interpolated motion in the rig area is shown as dashed arrows. The rig can be totally removed by frame 3 (no hatched area left).

It is possible to take an object based approach to this problem. The motion of the scene can be used to segment the sequence into a number of interacting layers, and the estimation problem is to synthesise a complete layer for each image frame. However it is clear that motion based image sequence segmentation is a difficult problem, particularly when realistic motion is complex due to fast moving objects, motion blur and non- rigid bodies. Instead, a more pragmatic approach in the medium term is to employ local measures of motion and reconstruct the image data using a recursive picture building process.

To illustrate the basic idea, Figure 1 shows a view of five one-dimensional image frames containing two moving objects. The rig is the lower object. The motion of the top object, background and rig is indicated with blue, black and red arrows respectively. For simplicity The diagrams show only motion in one direction. The dashed motion in the region of the rig shows the situation if it were possible to reconstruct the motion of the sequence, without the presence of the rig. Using this motion information it is possible to reconstruct the data hidden by the rig by recursively propagating data from *non-rig* regions into the rig obscured region in each frame.

Figure 1 also shows how this propagation can take place. In frame 2, motion information that maps rig data onto non-rig data in frame 1 can be used to *pull* image material into a small portion of the rig. Thus the bottom of the rig in frame 2 can be filled in with

[*] anil.kokaram@tcd.ie www.mee.tcd.ie/∼ack
[†] bill@thefoundry.co.uk, simon@thefoundry.co.uk

0-7803-7750-8/03/$17.00 ©2003 IEEE.

a bit of frame 1 (this is shown as a green (right diagonal patterned) patch). A similar situation exists between frame 2 and frame 3, allowing a part of frame 3 to patch the top of the rig (indicated by the brown (left diagonal) patch). In frame 3 the same concept allows more of the rig to be removed. After just 3 frames in this case (depending in general on the amount of motion the rig is undergoing) completely reconstructed images, without any rig, can be generated. After this process, the first few frames still contain part of the rig as shown by the presence of the large hatched region in the rig area of the first frame (also see Figure 2). However a backward recursive pass will allow propagation of reconstructed data from the future frames into these partially rebuilt past frames to complete the picture building process. Note in addition, that both forward and backward motion can be used simultaneously to reconstruct rig data in each frame.

In summary, the essential idea is to reconstruct motion in the rig area, then to use that motion to reconstruct the picture. The main problem with this idea is the interpolation of motion in the region of the rig while handling occlusion and uncovering. By using a Bayesian approach to the problem, a suitable spatio-temporal scheme can be built. This is one of the main contributions of the paper and is discussed next.

2. MOTION RECONSTRUCTION

A basic translational motion image sequence model is used as follows.

$$I_n(\mathbf{x}) = I_{n-1}(\mathbf{x} + \mathbf{d}_{n,n-1}(\mathbf{x})) + e(\mathbf{x}) \tag{1}$$

Where $\mathbf{x}$ indicates the location of a pixel $\mathbf{x} = [i, j]$, the intensity at that site in frame n is denoted by $I_n(\mathbf{x})$ and the two component motion vector mapping that site into the previous frame is given by $\mathbf{d}_{n,n-1}(\mathbf{x})$. $e(\mathbf{x})$ accounts for uncertainty in the model and is assumed to follow a Gaussian distribution $\mathcal{N}(0, \sigma_e^2)$. Although the model accounts for translation motion only, it is used at the pixel resolution. In the framework that follows, this implies that more complex motion fields can be handled.

The problem is to reconstruct the motion $\mathbf{d}^h_{n,n-1}(\cdot)$ (backward) and $\mathbf{d}^h_{n,n+1}(\cdot)$ (forward) at sites covered by the rig[1]. The rig sites are denoted by $\mathbf{x}_r$. The motion of the rig itself is denoted by $\mathbf{d}^r_{n,n-1}(\cdot)$. Also important is to configure an occlusion field $o_{n,n-1}$, $o_{n,n+1}$ that indicates temporal discontinuities at the boundaries of moving objects. To simplify the arguments that follow, only the backward motion $\mathbf{d}^h_{n,n-1}(\cdot)$ will be considered. A similar situation exists in the forward direction. Consider for the moment that motion fields for the entire sequence have been obtained, excepting at the rig locations denoted as the sites $\mathbf{x}_{-r}$. Thus $\ldots, \mathbf{d}_{n-1,n-2}(\mathbf{x}_{-r}), \mathbf{d}_{n,n-1}(\mathbf{x}_{-r})$ have all been obtained.

Proceeding in a probabilistic fashion it is necessary to manipulate the distribution $p(\mathbf{d}^h_{n,n-1}(\mathbf{x}_r)|\mathbf{d}^r_{n,n-1}(\mathbf{x}_r), \mathbf{d}(\mathbf{x}_{-r}), \mathbf{I})$, where $\mathbf{I}$ denotes all previous and next frames. The best estimate for $\mathbf{d}^h_{n,n-1}(\cdot)$ is that which maximises this probability. To continue, Bayes' law allows the distribution to be decomposed as follows.

$$\begin{gathered}p(\mathbf{d}^h_{n,n-1}(\mathbf{x}_r), o_{n,n-1}(\mathbf{x}_r)|\mathbf{d}_{n,n-1}(*\mathbf{x}_r), \mathbf{I}) = \\ p_l(I(\mathbf{x}_r)|\mathbf{I}_{-\mathbf{r}}, \mathbf{D})p_t(\mathbf{d}^h_{n,n-1}(\mathbf{x}_r)|\mathbf{D}_{n-1,n-2}, o_{n,n-1}) \\ p_s(\mathbf{d}^h_{n,n-1}(\mathbf{x}_r)|\mathbf{D}_{n,n-1}(*\mathbf{x}_r))p_s(o_{n,n-1}(\mathbf{x}_r)|o_{n,n-1}(*\mathbf{x}_r))\end{gathered} \tag{2}$$

$p_l(\cdot)$ denotes the *likelihood* of the image data *given* all the required motion information at each pixel site $\mathbf{D}$. $p_t(\cdot)$ is the prior probability of a particular choice of hidden motion in the current frame given the motion in the *previous* frame $\mathbf{D}_{n-1,n-2}$, and $*\mathbf{x}_r$ indicates all sites not including $\mathbf{x}_r$. This encourages temporal motion smoothness. $p_s(\cdot)$ is a spatial smoothness constraint on the interpolated motion field. To design a suitable algorithm, meaningful expressions must be attached to these concepts.

The image data likelihood The model in equation 1 is used to impose the constraint that the image data matched by motion vectors between frames should be roughly the same. Because the observed image sequence is only partially observed i.e. obscured by the rig, it becomes useful to attach weights to each pixel in each frame. This weight field, $w_n(\mathbf{x})$ is a continuous variable between 1 (for data available) and 0 for data missing. In the rig area, $w(\mathbf{x}_r) = 0$. The image data likelihood can then be defined as follows (dropping the argument $\mathbf{x}_r$ for brevity.

$$p_l(I_n|\cdot) \propto \exp - \frac{1}{2\sigma_e^2} w_n w_{n-1}(\mathbf{x}'_r)(I_n - I_{n-1}(\mathbf{x}'_r))^2 \tag{3}$$

where $\mathbf{x}'_r$ denotes the motion compensated site $\mathbf{x}_r + \mathbf{d}^h_{n,n-1}(\mathbf{x}_r)$. Thus the likelihood is proportional to the weighted image matching error between frames. However, since in the rig area the weight is zero, the image data likelihood has no effect on the motion interpolation problem and can be ignored. Note that a *product* of weights is used here since this temporal likelihood is only useful when *both* motion compensated pixels contain known data.

Temporal smoothness It is through temporal smoothness that the motion in the rig area can be interpolated as shown in Figure 1. Assuming little acceleration between frames the distribution can be written as follows.

$$\begin{aligned}p_t(\mathbf{d}^h_{n,n-1}|\cdot) \propto \exp - & \frac{1}{\sigma_v^2}(1 - o_{n,n-1})w_{n-1}(\mathbf{x}'_r) \\ & \times|\mathbf{d}^h_{n,n-1} - \mathbf{d}_{n-1,n-2}(\mathbf{x}'_r)|^2\Big]\end{aligned} \tag{4}$$

where $|\cdot|$ denotes the Euclidean vector difference. This prior penalises vectors that do not match well with their motion compensated counterpart in the previous frame. The occlusion variable $o_{n,n-1}$ at site $\mathbf{x}_r$ allows large mismatch between motion compensated vectors to indicate a motion discontinuity. The error is weighted only by the previous image weights so that the motion smoothness term is only valid when the previous motion compensated image data is not at a rig location. σ_v^2 represents the amount of acceleration that is allowed. Small values < 1 penalise acceleration heavily, while large values allow poor temporal vector matches. A value of 0.01 is used here to encourage low acceleration.

Spatial Smoothness This is a common concept in any consideration of motion estimation. The idea is to ensure that in a local region the motion and occlusion field is smooth since objects tend to be locally well connected. The prior adopted for motion is a Gibbs Energy prior (e.g. Konrad and Dubois [3]) as follows.

$$p_s(\mathbf{d}^h_{n,n-1}|\cdot) \propto \exp - \left(\sum_{\mathbf{s} \in \mathbf{S}_n(\mathbf{x})} \lambda(\mathbf{s})|\mathbf{d}^h_{n,n-1} - \mathbf{d}(\mathbf{s})|^2 \right) \tag{5}$$

[1] Superscript h is used to indicate the underlying *hidden* image motion

where $\mathbf{s}$ is each motion vector in the 8 connected neighbourhood represented by $\mathbf{S}_n(\mathbf{x})$, and $\lambda(\mathbf{s})$ is the weight associated with each clique. The neighbourhood $\mathbf{S}_n(\mathbf{x})$ is the 8 nearest neighbour. The occlusion prior uses the Ising model (similar to equation 5), with the addition of a penalty term $\exp-(\alpha o_{n,n-1}(\cdot))$ to prevent $o_{n,n-1}$ set to 1 everywhere. $\Lambda = 2.0$ in the results presented later.

3. A PRACTICAL SOLUTION

Solving equation 2 for $\mathbf{d}^h(\cdot)$ given the various component expressions 3, 4, 5 is not straightforward if the problem is treated as direct estimation. This is because the arguments of the various motion compensation actions required are also unknowns. The first step in simplifying the solution is to use the notion of ICM or local conditional maximisation [4]. A solution is therefore generated at each pixel site conditioned on the state of the sites around. Each site is visited in turn and after a number of passes over the image, the motion field converges to some overall state. The second step in designing a simple solution is to recognise that it is possible to generate a number of reasonably good *initial* estimates for $\mathbf{d}^h(\cdot)$ using straightforward, deterministic ideas. These estimates can then be used as candidate solutions. Each candidate is evaluated according to the probability criterion in equation 2, and the best candidate selected at each site. There are two stages in generating possible candidate solutions discussed next.

Weighted motion estimation Using the image weights $w(\mathbf{x})$ described previously, it is possible to define an error criterion for motion estimation that ignores the rig area as follows

$$\epsilon(\mathbf{x})^2 = w_n(\mathbf{x})w_{n-1}(\mathbf{x}')(I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} + \mathbf{d}(\mathbf{x})))^2 \qquad (6)$$

Many different approaches can be used to estimate motion with this criterion and here a block based version of the Wiener estimation approach [5] is used. The weighted approach ensures that the error in image matching at the rig sites does not affect the motion estimation process at the edges of the rigs. Obviously, within the rig area, no motion estimates can be generated with this method as the weights of all pixels are zero. The final weighted gradient based motion estimator is applied on a multiresolution pyramid. 4 levels are used, with block sizes of $9,\ 9,\ 5,\ 5$ at each level of the pyramid.

Spatially interpolated Candidates The previous process will allow motion to be estimated for those blocks that overlap the rig region. In order to create candidate motion estimates inside the rig, a simple idea is to spatially interpolate the motion field within the gap. There are two simple methods that perform well. The first is to interpolate the vector field using the motion smoothness prior in equation 5, using instead a weighted energy. Thus a pairwise term is removed if the neighbourhood vector concerned has not yet been interpolated. Using ICM, each vector is interpolated in turn starting from the outside of the rig and moving inwards.

A second useful method is to assume that the material hidden by the rig is moving with only one single motion that is the same as the region immediately surrounding the rig. This would be the case if the rig is moving against a large rigid body background for instance. A rectangular area that encases the rig could then be used for estimation with the weighted criterion shown above. This is similar to the global motion estimation ideas previously presented in [6].

Temporally interpolated candidates In the case of low acceleration, motion between $n-1, n-2$, suitably motion compensated, is a good estimate for the motion between frames $n, n-1$. Vectors $\mathbf{d}_{n-1,n-2}$ can therefore be used as candidates for $\mathbf{d}^h_{n,n-1}$ at locations in frame n indicated by $\mathbf{x} - \mathbf{d}_{n-1,n-2}(\mathbf{x})$. These motion candidates are assigned to the nearest integer pixel site in n. Only vectors that 'hit' in rig locations need to be recorded.

THE FINAL ALGORITHM Consider a site $\mathbf{x}_r$ and the backward motion $\mathbf{d}^h_{n,n-1}(\mathbf{x}_r)$. At each site in the rig of frame n, a list of motion candidates can be collected using the temporally projected set and any of the eight nearest neighbours that have already been assigned. Denote the ith vector in this list of N vectors as $\mathbf{d}^c_i$. For each $\mathbf{d}^c_i$ two possible occlusion states are associated. $o_{n,n-1}(\mathbf{x}_r) = 0, 1$. This creates $2N$ motion candidates. For each such motion/occlusion candidate the log posterior density is evaluated from equation 2. This amounts to summing a spatial smoothness error (for both motion and occlusion), a temporal smoothness error and a DFD term for each motion/occlusion candidate. The candidate with the smallest error is selected as the interpolated vector. This process is iteratively repeated over the rig region, and again for the forward motion.

Finally, the hidden data in the rig region, is estimated from $p_l(\cdot)$ (across three frames) as $\hat{I}^h_n$ using weighted interpolation as follows.

$$\hat{I}^h_n = \frac{w_n I_n + w'_{n-1} I'_{n-1} + w'_{n+1} I'_{n+1}}{w_n + w'_{n-1} + w'_{n+1}} \qquad (7)$$

where w', I' denotes motion compensation. To recursively reconstruct the rig, the weight image w_n is updated by performing the same interpolation process on the weight image sequence. Thus the reconstructed portions in frame n are automatically used in reconstructing the image in frame $n+1$.

4. PICTURES

Figure 3 shows results from removing a moving motorcycle in a real scene with PAL resolution frames. The original data is shown together with the user defined matte in a red overlay. There is substantial camera motion, and the user defined matte is only a rough outline that does not allow for objects moving behind or infront of the rig. The sequence was processed using 5 iterations for motion interpolation with the algorithm above, $\alpha = 2.76$, and σ_e^2 measured from non-rig parts of the image. A forward and backward pass of the algorithm was used in order to complete the 'fill in' operation in the early frames of recursion. Figure 2 shows an example of the motion interpolation action as well as the effect of recursion on rig removal. For longer video examples see `www.mee.tcd.ie/~sigmedia/postpro`.

5. FINAL COMMENTS

This paper has introduced a novel mechanism for the automated removal of rigs in image sequences. The use of motion interpolation is important for the success of the algorithm, and it allows a recursive approach to fill in the region as it moves across the background. The use of a candidate selection strategy for motion interpolation allows a straightforward implementation of the algorithm. The process presented here is currently being tested by The Foundry (a London based film effects software house) and the user feedback is already positive. The pelwise constraint strategy

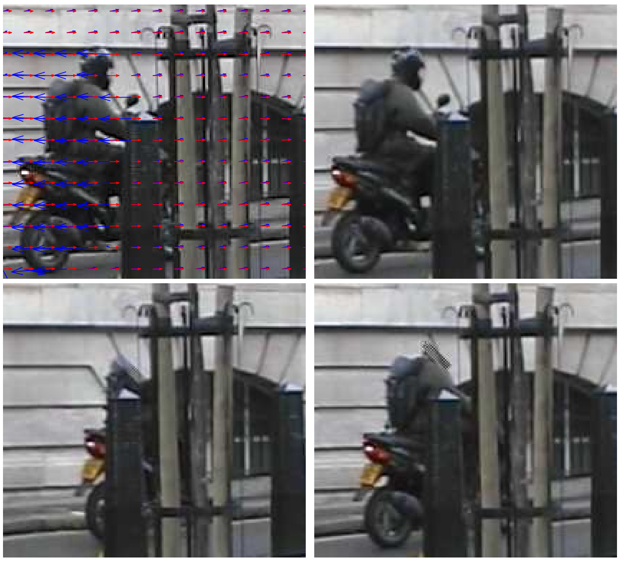

Fig. 2. Top Left: Frame 2 with original motion (blue) and interpolated motion (red). Clockwise from top right: First three consecutive rig removed frames from a forward pass of the algorithm. Note progressively more of the 'rig' is removed with time.

potentially allows complex motion to be handled, but effects like motion blur and self-occlusion are still not well modelled by the occlusion framework presented. That is the subject of future work.

6. REFERENCES

[1] Alexei A. Efros and Thomas K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, September 1999, vol. 2, pp. 1033–1038.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings SIGGRAPH*, 2000.

[3] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, September 1992.

[4] J. Besag, "On the statistical analysis of dirty pictures.," *Journal of the Royal Statistical Society B*, vol. 48, pp. 259–302, 1986.

[5] J. Biemond, L. Looijenga, D. E. Boekee, and R.H.J.M. Plompen, "A pel–recursive Wiener based displacement estimation algorithm.," *Signal Processing*, vol. 13, pp. 399–412, 1987.

[6] J-M. Odobez and P. Bouthémy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, pp. 348–365, 1995.

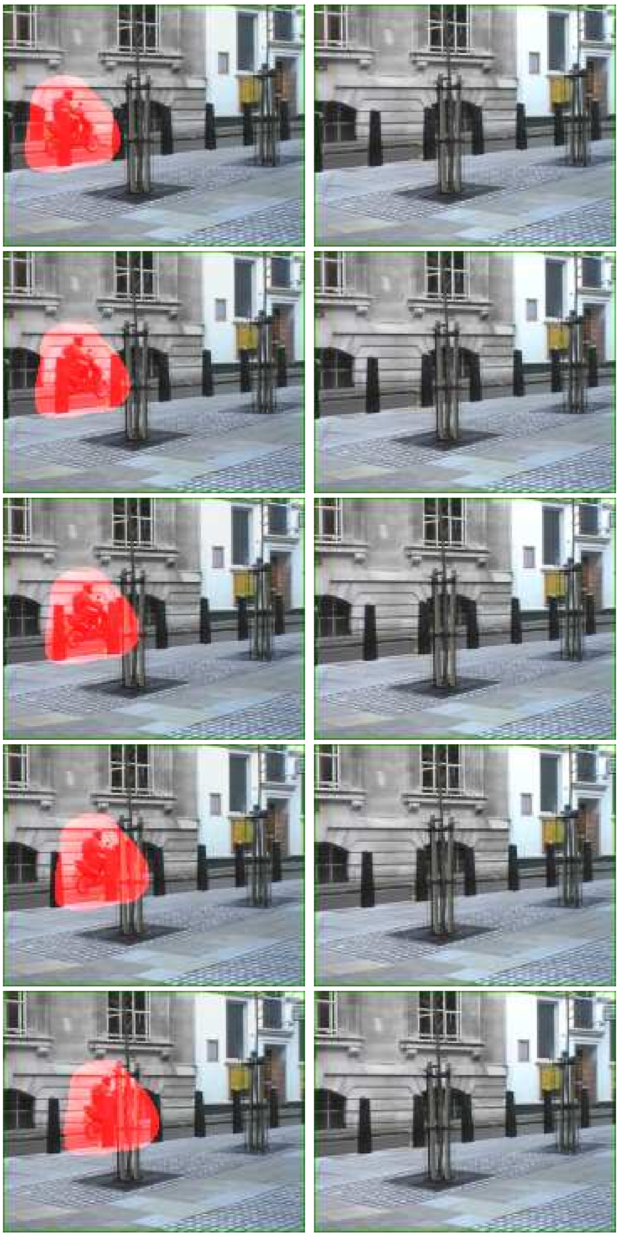

Fig. 3. Left column: The user defined matte that delineates the rig is overlaid in red. Right column: The results of rig removal. See `www.mee.tcd.ie/~sigmedia/postpro`