

# AUTOMATED RIG REMOVAL WITH BAYESIAN MOTION INTERPOLATION

*Anil C. Kokaram*

Department of Electronic Engineering,  
Trinity College Dublin, Ireland.

*Bill Collis and Simon Robinson*

The Foundry,  
London 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, cranes and other objects 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. Provided the undesired object (the *rig*) is moving, it is possible to automatically use the motion throughout the sequence to reconstruct the image material that was obscured. The work presented here takes a novel approach that allows the estimation of the motion of the material beneath the rig and then the reconstruction of the missing image material. A Bayesian framework is used to solve the motion reconstruction problem, and a unique tool is developed for automated rig removal. This tool holds great potential for speeding up one of the major tasks performed in the effects industry and is currently being tested in that environment.

Keywords: rotoscoping, image-based rendering, matting and compositing, video processing, motion estimation, Bayesian Inference.

## 1. INTRODUCTION

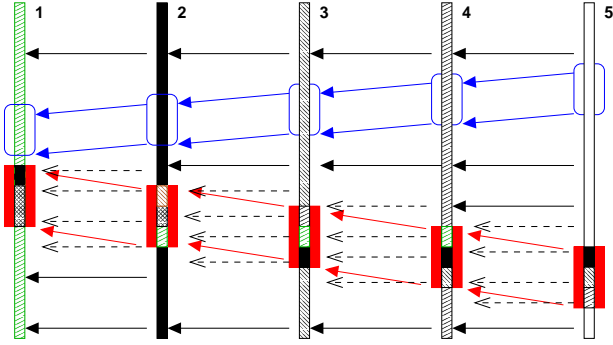
Some of the most convincing film and video effects are created in digital post-production by removing apparatus that supports or manipulates actors and objects. The undesired apparatus e.g. wires, cranes; will be termed *rigs* in the rest of this paper. Of course some of the undesired ‘rig’ material 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

would be interesting to consider whether it is possible to remove rigs without the need for a clean plate.

Figure 6 shows a sequence of several frames in which the rig to be removed is delineated. 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 [7, 6, 1] would be more suitable, although the success of those techniques would depend on the relative difference between the size of the rig and the *scale* of the underlying ‘texture’.

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. It is assumed that the user has roughly outlined the region to be reconstructed in each frame.

In a way, the rig removal problem contains issues similar to that encountered in *Mosaicking* and *Background Estimation* as studied by the Computer Vision [16, 4, 15] and Video Processing community [12], especially with regard to MPEG4. Given a sequence of images taken of some background view, the task there is to estimate the entire background and synthesise it as a single picture. This requires that *all* moving objects be ignored. A typical mechanism for executing this task involves a combination of semi-automated user identification of all foreground regions and estimation of the warping parameters be-



**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).

tween frame pairs. This latter step allows the “common area” between frames to be estimated as well as the geometric warp needed to “stitch” the images together.

In this paper, the problem addressed is not that of estimating the background image or that of creating a single panoramic view. The problem here is to remove in each frame, a **single** moving object specified by the user. Therefore, a clean plate is *not* the result that is sought. Rather, the goal is to create a new sequence in which that single object is removed, revealing the hidden image, regardless of its interaction with other moving objects in the scene.

The novelty in the process presented here is both in its algorithm and the practical implementation. The novel algorithmic aspect is the presentation of motion interpolation with new priors together with a low cost solution. A unique aspect is that the process discussed here is able to deal with multiple overlapping motions. The practical issue of recursive interpolation is also treated.

## 2. THE ESSENTIALS

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. Note that the cross-hatched region in the red object area indicates the amount of remaining object that is to be removed in each frame. 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 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 7). 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.

## 3. MOTION RECONSTRUCTION

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

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

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  $\mathbf{d}_{n,n-1}^h(\cdot)$  (backward) and  $\mathbf{d}_{n,n+1}^f(\cdot)$  (forward) at sites covered

by the rig<sup>1</sup>. The rig sites are denoted by  $\mathbf{x}_r$ . The motion of the rig itself is denoted by  $\mathbf{d}_{n,n-1}^r(\cdot)$ . 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. These variables are binary,  $o(\mathbf{x}) = 1$  indicates a discontinuity at site  $\mathbf{x}$  while  $o(\cdot) = 0$  indicates no discontinuity is present. To simplify the arguments that follow, only the backward motion  $\mathbf{d}_{n,n-1}^h(\cdot)$  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  $\mathbf{d}_{n-1,n-2}(\mathbf{x}_{-r})$ ,  $\mathbf{d}_{n,n-1}(\mathbf{x}_{-r})$  have all been obtained.

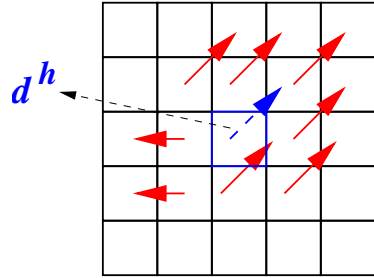
Proceeding in a probabilistic fashion it is necessary to manipulate the distribution  $p(\mathbf{d}_{n,n-1}^h | \mathbf{d}_{n,n-1}^r(\mathbf{x}_r), \mathbf{d}(\mathbf{x}_{-r}), \mathbf{I})$ , where  $\mathbf{I}$  denotes all previous and next frames. The best estimate for  $\mathbf{d}_{n,n-1}^h(\cdot)$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

$$\begin{aligned} p(\mathbf{d}_{n,n-1}^h(\mathbf{x}_r), o_{n,n-1}(\mathbf{x}_r) | \mathbf{d}_{n,n-1}(*\mathbf{x}_r), \mathbf{I}) = \\ p_l(I(\mathbf{x}_r) | \mathbf{I}_{-r}, \mathbf{D}) p_t(\mathbf{d}_{n,n-1}^h(\mathbf{x}_r) | \mathbf{D}_{n-1,n-2}, o_{n,n-1}) \\ \times p_s(\mathbf{d}_{n,n-1}^h(\mathbf{x}_r) | \mathbf{D}_{n,n-1}(*\mathbf{x}_r)) \\ \times p_{so}(o_{n,n-1}(\mathbf{x}_r) | o_{n,n-1}(*\mathbf{x}_r)) \end{aligned} \quad (2)$$

where  $*\mathbf{x}_r$  indicates all sites not including  $\mathbf{x}_r$ .  $p_l(\cdot)$  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  $\mathbf{D}_{n-1,n-2}$ . This encourages temporal motion smoothness.  $p_s(\cdot), p_{so}$  contain spatial smoothness constraints on the interpolated motion and occlusion fields. To design a suitable algorithm, meaningful expressions must be attached to these concepts.

### 3.1. 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  $\mathbf{x}_r$



**Fig. 2.** A  $5 \times 5$  pixel grid is shown. The vector (dashed) at the centre site is the focus of interest here. Vectors (solid) at the neighbourhood set of 8 sites,  $\mathbf{x} \in \mathbf{S}_n(\mathbf{x})$ , are also indicated.

for brevity.

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

where  $\mathbf{x}'_r$  denotes the motion compensated site  $\mathbf{x}_r + \mathbf{d}_{n,n-1}^h(\mathbf{x}_r)$ . 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.

### 3.2. 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.

$$\begin{aligned} p_t(\mathbf{d}_{n,n-1}^h | \cdot) \propto \exp - \left[ \frac{1}{\sigma_v^2} (1 - o_{n,n-1}) w_{n-1}(\mathbf{x}'_r) \right. \\ \left. \times |\mathbf{d}_{n,n-1}^h - \mathbf{d}_{n-1,n-2}(\mathbf{x}'_r)|^2 \right] \end{aligned} \quad (4)$$

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.  $\sigma_v^2$  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.

<sup>1</sup>Superscript  $h$  is used to indicate the underlying *hidden* image motion

### 3.3. Spatial Motion 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 [9]) as follows.

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

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. Figure 2 shows this neighbourhood of 8 sites.

The driving force behind the specification of this prior is to penalise the creation of motion vector fields that have a high local gradient. Thus the vector most likely to be appropriate is that which is aligned with the same directions as most of the other vectors. In the case shown in Figure 2 that direction is to the right and up. In order to discourage ‘smoothness’ over too large a range,  $\lambda(\mathbf{s})$  is defined as  $\lambda(\mathbf{s}) = \Lambda/|\mathbf{s} - \mathbf{x}_r|$ .  $\Lambda = 2.0$  in the results presented later.

### 3.4. Spatial Occlusion Smoothness

The occlusion prior uses the Ising model (similar to equation 5), with the addition of a penalty term.

$$p_{so}(o_{n,n-1}|\cdot) \propto \left[ \exp - \left( \sum_{\mathbf{s} \in \mathbf{S}_n(\mathbf{x})} \lambda_o(\mathbf{s}) |o_{n,n-1} - o_{n,n-1}(\mathbf{s})| \right) \right] \left[ \exp - (\alpha o_{n,n-1}(\cdot)) \right] \quad (6)$$

The penalty,  $\alpha$  is set to  $3.31^2$ . Without this penalty, setting  $o = 1$  everywhere will maximise the probability of any vector in the temporal motion prior specified in equation 4. The penalty energy therefore acts in balance with the temporal motion prior. Choosing this value of  $\alpha$  ensures that the temporal energy must be larger than about  $3\sigma_v$  before occlusion is set. As the temporal distribution is Gaussian, the probability of this happening is  $< 10\%$ . To be more ‘certain’ that occlusion should be set,  $\alpha$  can be increased further. This argument is not strictly true in the broader scheme of spatial and temporal interactions, but it does allow some logic to be applied in choosing the value of this hyperparameter,  $\alpha$ .

## 4. A PRACTICAL SOLUTION

Solving equation 2 for  $\mathbf{d}^h(\cdot)$  given the various component expressions 3, 5, 4 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 local conditional maximisation [2] is used. In this approach, a solution is 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. Thus at each site the probability distribution, exactly as stated in equation 2, is manipulated.

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. Firstly, a weighted error criterion can be used to generate motion estimates at every pixel site outside the rig area. Then motion vectors from previous and next frames can be propagated toward the current frame  $n$  along their motion trajectories. At each site in frame  $n$  these propagated vectors are recorded. By collating the temporal propagated vectors with spatially collected neighbours, a list of candidate solutions is created. One vector from this list is then selected according to 2. These various stages are discussed next.

### 4.1. 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

$$\epsilon(\mathbf{x})^2 = w_n(\mathbf{x})w_{n-1}(\mathbf{x}')(I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} + \mathbf{d}(\mathbf{x})))^2 \quad (7)$$

This weighted criterion ensures that the error in image matching at the rig sites does not affect the motion estimation process at the edges of the rigs. Many different approaches (developed in the Computer Vision and Video Processing communities) can be used to estimate motion with this criterion e.g. optic flow [14, 11], block matching [8], pel-recursion [3]. Although these previous works do not consider *weighted* motion estimation, work in robust methods for global and camera motion estimation [13, 5] shows how those ideas can be adapted for use in a related context. The important aspect of generating *candidate solu-*

tions however, is that the motion estimation process used can be rough, as long as motion is correctly estimated over *some portion of the image*. The candidate selection process that is described later, will ideally propagate the correct solution throughout each image. A suitably fast, and fairly reasonable motion estimation process can therefore be adopted.

A block based version of the Wiener motion estimation approach [3] is used, adapted for weighted motion estimation. A single vector for each  $B \times B$  block of pixels can be generated by minimising  $\sum_{\mathbf{x} \in \mathcal{B}} \epsilon(\mathbf{x})^2$  with respect to the motion  $\mathbf{d}(\mathbf{x})$ . The principal idea in using the weighted approach is to ensure 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. A multiresolution coarse to fine refinement process allows this gradient based approach to yield useful motion estimates for real image sequences. See [8] for further details. Four resolution levels are used with the coarsest level being 8 times smaller than the original image size in each direction.

## 4.2. Spatially interpolated Candidates

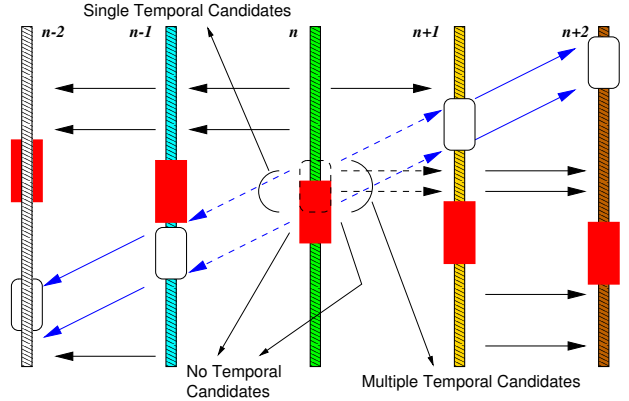
The previous process will allow motion to be estimated for those blocks that overlap the rig region. In order to create candidate motion estimates inside the rig, a simple idea is to spatially interpolate the motion field within the gap. There are two simple methods that perform well.

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. Denoting the sites  $\mathbf{x}_r$  that have been assigned a vector as  $a(\mathbf{x}_r) = 1$  and 0 otherwise, the algorithm for interpolation is as follows.

1. Visit each pixel site within the rig
2. At each site collect the eight nearest neighbour motion vectors that have been previously assigned. Denote these vectors as  $\mathbf{v}_c$ , possible candidates for interpolation.
3. Evaluate the following spatial smoothness criterion for each vector in the list.

$$E(\mathbf{v}_c) = \lambda \sum_k a(\mathbf{x}_k) |\mathbf{v}_c - \mathbf{s}(\mathbf{x}_k)| \quad (8)$$

$\mathbf{s}(\mathbf{x}_k)$  are the vectors in the neighbourhood, and the spatial energy of each pairwise interaction



**Fig. 3.** To illustrate how motion estimates between frames  $n + 1, n + 2$ , and  $n - 1, n - 2$ , estimated using non-rig data, can be projected back along their trajectories to yield candidates for motion from frame  $n$  into  $n + 1$  and  $n - 1$  respectively. The red patch in frame  $n$  is to be removed to reveal the other moving object behind it. At the interface between moving objects there are usually multiple candidates (multiple hits) or no candidates (holes) in frame  $n$ . Single hits occur when there is only a single moving object.

is only considered if the neighbourhood vector has been previously assigned,  $a(\mathbf{x}_k) = 1$ .

4. Select the interpolated vector as that which yields the lowest spatial smoothness error.
5. Move to the next site and repeat from step 2.

Note that this process is far from optimal, a vector based inpainting approach [1] may be more appropriate. However, it is not necessary that this interpolated field be accurate. It is only necessary for it to be *reasonable*, since these vectors will only be used as further candidates in the larger spatiotemporal rig removal problem.

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 and a small portion of surrounding information could then be used for estimation with the weighted criterion shown above. This would be similar to the global motion estimation ideas previously presented in [5, 13].

## 4.3. 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  $n, n - 1$ . Vec-

tors  $\mathbf{d}_{n-1,n-2}$  can therefore be used as candidates for  $\mathbf{d}_{n,n-1}^h$  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 process of generating candidates by *motion compensated temporal propagation* is shown in figure 3. The figure shows at frame  $n$  that there are two objects interacting. Between frames  $n, n + 1$  there are multiple candidates for motion because the white object has moved into the rig region which is itself moving through the scene. Between  $n, n - 1$  there are single hits where just one motion dominates, and no hits because the moving object moves past the rig region in frame  $n - 1$ . No hits are made between  $n, n + 1$  in the middle of the rig region because that coincides directly with rig region in frame  $n + 1$  hence no initial motion candidates can be generated at that location between frames  $n + 1$  and  $n + 2$ . The weights  $w(\mathbf{x})$  automatically allow for this in the specification of the motion priors. That figure also illustrates that in the case of no acceleration, the temporal motion candidates can be very good estimates for the underlying object motion.

#### 4.4. The overall motion reconstruction algorithm: Candidate Evaluation

Consider a site  $\mathbf{x}_r$  and the backward motion  $\mathbf{d}_{n,n-1}^h(\mathbf{x}_r)$ . 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  $i$ th vector in this list of  $N$  vectors as  $\mathbf{d}_i^c$ . For each  $\mathbf{d}_i^c$  two possible occlusion states are associated.  $o_{n,n-1}(\mathbf{x}_r) = 0, 1$ . This creates  $2N$  motion candidates:  $[\mathbf{d}_0^c, 0], [\mathbf{d}_0^c, 1], [\mathbf{d}_1^c, 0], [\mathbf{d}_1^c, 1], \dots, [\mathbf{d}_{N-1}^c, 0], [\mathbf{d}_{N-1}^c, 1]$ . For each such motion/occlusion candidate the log posterior density is evaluated from equation 2 using the expressions in equations 3, 4, 5. 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.

The algorithm is enumerated as follows, for motion between  $n, n - 1$ . At each site  $\mathbf{x}_r$  DO:

1. Collect candidate vectors  $\mathbf{d}_i^c, i = 1 \dots N$

2. For each vector evaluate the following energies.

$$E_l(i) = \frac{1}{2\sigma_e^2} w_n w_{n-1}(\mathbf{x}'_r) (I_n - I_{n-1}(\mathbf{x}'_r))^2 \quad (9)$$

$$E_t^0(i) = \frac{1}{\sigma_v^2} w_{n-1}(\mathbf{x}'_r) \times |\mathbf{d}_{n,n-1}^h - \mathbf{d}_{n-1,n-2}(\mathbf{x}'_r)|^2$$

$$E_t^1(i) = \alpha$$

$$E_s(i) = \sum_{\mathbf{s} \in \mathbf{S}_n(\mathbf{x})} \lambda(\mathbf{s}) |\mathbf{d}_{n,n-1}^h - \mathbf{d}(\mathbf{s})|^2$$

$$E_o^0(i) = \sum_{\mathbf{s} \in \mathbf{S}_n(\mathbf{x})} \lambda(\mathbf{s}) |o(\mathbf{s})|^2$$

$$E_o^1(i) = \sum_{\mathbf{s} \in \mathbf{S}_n(\mathbf{x})} \lambda(\mathbf{s}) |1 - o(\mathbf{s})|^2$$

3. To each candidate, associate two energies  $E_0(i) = E_l(i) + E_t^0(i) + E_s(i) + E_o^0(i)$  and  $E_1(i) = E_l(i) + E_t^1(i) + E_s(i) + E_o^1(i)$ .
4. Find the candidate energy that is minimum,  $i_m$ . If for  $i_m, E_0$  is minimum, then assign  $o_{n,n-1} = 0$  else assign  $o_{n,n-1} = 1$ .
5. Move to the next site.

#### 4.5. Image Reconstruction

Given the reconstructed motion fields, the hidden data in the rig region, is estimated from  $p_l(\cdot)$  (across three frames) as  $\hat{I}_n^h$  using weighted interpolation as follows.

$$\hat{I}_n^h = \frac{w_n I_n + w'_{n-1} I'_{n-1} + w'_{n+1} I'_{n+1}}{w_n + w'_{n-1} + w'_{n+1}} \quad (10)$$

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 assigned higher weights and therefore are automatically used in reconstructing the image in frame  $n + 1$ .

##### 4.5.1. Coping with Recursive Interpolation

Bilinear interpolation is too poor to be useful for repeated motion compensation of the progressively reconstructed rig region motion. After a few frames the image in the region of the rig rapidly loses its detail. Windowed sinc interpolation [10], or bicubic interpolation, for instance, gives much better fidelity. However, better image interpolators necessarily employ a larger filter support, for example a window of  $9 \times 9$  pixels. At the rig boundary, it can occur that very

small image perturbations occur, leading to weights that are numerically small, but unstable. To prevent this problem from propagating, weights that are less than some threshold are assumed to imply a low pixel ‘confidence’ and are set to 0. A threshold of 0.3 was used here.

## 5. PICTURES

Figure 4 shows a sequence constructed by overlaying two rectangular objects against some textured background. The flame object is moving to the right, while the other object is moving to the left. The situation is the two-dimensional equivalent to that illustrated in Figure 1. The task is to remove the *flame* ‘rig’, revealing the background and other moving object as necessary. The sequence was processed using 5 iterations for motion interpolation with the algorithm above,  $\sigma_e^2$  measured from non-rig parts of the image, and  $\sigma_v^2 = 0.01$ . The figure shows the results from a forward pass through a few frames, and more of the rig is removed with each frame as expected.

Figure 5 shows the behaviour of the motion interpolation process. It correctly fills in the missing rig region with the correct motion. The forward motion in this case is not shown since rig is always present in the next frame and the weights therefore disallow the assignment of motion. The top row shows the initial motion estimate and the bottom row shows the reconstruction. The behaviour of the algorithm at the edges of the moving non-rig object, is interesting. The top left corner and the left edge shows a few pixels in error. This is because at a few of those locations the algorithm is unable to decide on which object is background/foreground since there is no edge process that shuts down motion smoothness at moving edges. Omitting this edge process was a practical choice since it would increase the variables to be estimated substantially. Nevertheless, this is a small price to pay since it does not affect the reconstruction heavily. To our knowledge, most of the previous work on Mosaicing that could be related to rig-removal, is unable to deal with the problem of multiple overlapping motions as in this example. This is because most of that work assumes that the moving regions are all to be removed and therefore they exist against a background showing cohesive motion.

Another interesting point to note is that the vector interpolation process is only used in the region of the rig. However, outside of that region, there may be some areas that need treatment simply because the initial motion estimate could be wrong for all the usual reasons of aperture effect, large motion over periodic structures etc. The last column in figure 5

shows this phenomenon at the right hand edge of the rig. There are two vectors that are clearly wrongly assigned, but as they are not part of the reconstruction process, the current implementation does not alter these vectors. This is purely a practical point, but it is sensible to avoid this problem simply by reconstructing a region which is slightly larger than the actual object to be removed. This also frees the user from having to specify exact masks or mattes for the rig. Instead, defining a rough matte, or garbage matte would be more practical and convenient.

Figure 6 shows results, using such a garbage matte to remove 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 in front of the rig. Nevertheless, the result is convincing. 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 7 shows an example of the motion interpolation action as well as the effect of recursion on rig removal. Note again, that more and more of the rig is removed in each consecutive frame. Until by frame 4 there is no rig left. In this example, the spatially interpolated candidates were generated by fitting an affine motion model to the known region around the garbage matte. For longer video examples see [www.mee.tcd.ie/~sigmedia/postpro](http://www.mee.tcd.ie/~sigmedia/postpro).

The current implementation (operating with 5 iterations) takes about 1 sec to reconstruct 2000 missing pixels on a 800 MHz PIII. This does not include the time spent on initial motion estimation, as that depends on the motion estimation process used. The implementation currently being tested in an industry environment includes a number of simplifying approximations. For instance, in many cases, the spatially interpolated vectors may be ‘good enough’ and using those alone allows the algorithm to improve in speed dramatically.

## 6. 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 candidate selection strategy for motion interpolation allows a straightforward implementation of the algorithm, and many simple motion estimation processes can be used to generate solution candidates. Although the pixelwise constraint strategy potentially allows complex motion to be handled, ef-

fects like motion blur and self-occlusion are still not well modelled by the occlusion framework presented. That is the subject of future work. 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.

## 7. REFERENCES

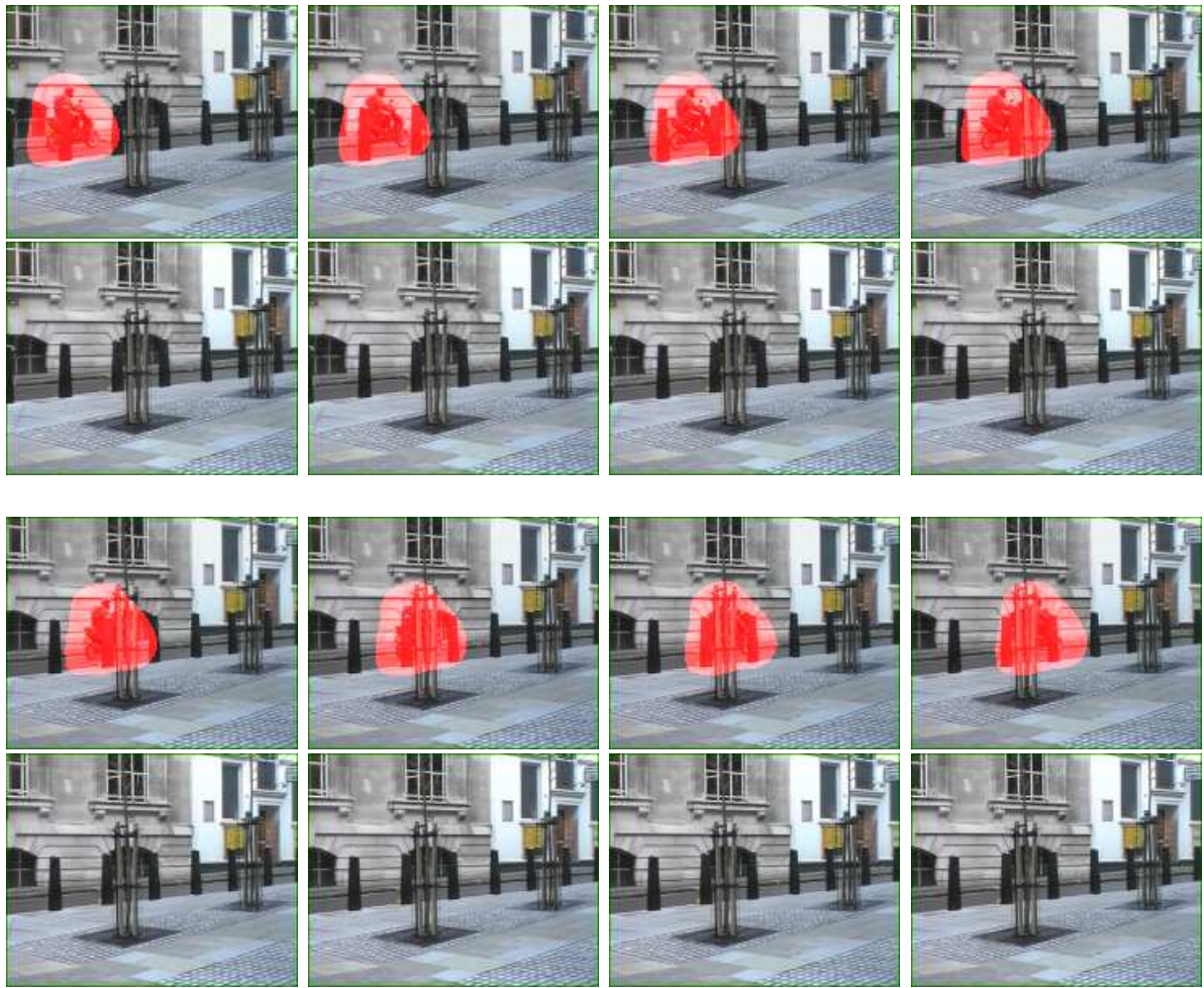
- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings SIGGRAPH*, 2000.
- [2] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48:259–302, 1986.
- [3] J. Biemond, L. Looijenga, D. E. Boeke, and R.H.J.M. Plompen. A pel-recursive Wiener based displacement estimation algorithm. *Signal Processing*, 13:399–412, 1987.
- [4] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video matting of complex scenes. In *Proceedings of ACM SIGGRAPH*, 2002.
- [5] F. Dufaux and J. Konrad. Efficient, robust and fast global motion estimation for video coding. *IEEE Transactions on Image Processing*, 9:497–501, 2000.
- [6] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings SIGGRAPH*, pages 341–346, 2001.
- [7] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038, September 1999.
- [8] A. C. Kokaram. *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*. Springer Verlag, ISBN 3-540-76040-7, 1998.
- [9] J. Konrad and E. Dubois. Bayesian estimation of motion vector fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9), September 1992.
- [10] Jae S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, 1990.
- [11] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector field from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–592, September 1986.
- [12] H. Nicolas. New methods for dynamic mosaicking. *IEEE Trans. Image Processing*, 10(8):1239–1250, August 2001.
- [13] J-M. Odobez and P. Bouthémy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6:348–365, 1995.
- [14] B. Schunck. The image flow constraint equation. *Computer Vision, Graphics and Image Processing*, 35:20–46, 1986.
- [15] Richard Szeliski and H.Y. Shum. Creating full view panoramic mosaics and environment maps. In *Proceedings of ACM SIGGRAPH*, pages 251–258, 1997.
- [16] Y. Wexler and A. W. Fitzgibbon and A. Zisserman. Image-based environment matting. In *Eurographics Workshop on Rendering*, pages 1–9, 2002.



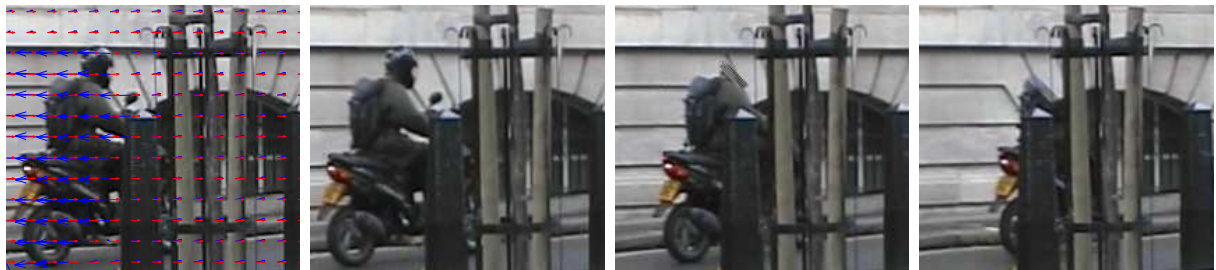
**Fig. 4.** Top Row: Artificially constructed sequence showing two moving objects. The flame object is moving to the right, while crowd is moving to the left. The blue background is stationary. The flame object is to be removed and it is assumed that the mask coincides directly with this object. Bottom Row: The recursive evolution of the rigremoval algorithm described in this paper. As the objects move, more and more of the rig can be removed. Note that the motion of the objects is not sufficient in these four frames to show the complete removal of the rig. The rig obscures the underlying object in the last three frames of this sequence, yet the edge reconstruction of the remaining object is still sharp. The situation here is similar to that illustrated in Figure 3.



**Fig. 5.** Illustrating the motion reconstruction performance of the algorithm. Only backward motion is shown here since in this example, the forward motion can never provide image information as the rig in the next frame always obscures the data to be interpolated. The pictures show a zoom on a portion of the images in Figure 4. The top row shows the initial motion estimate. Note that there are no motion vectors assigned to the rig region, since there is no image data there to generate initial motion estimates. The bottom row shows the reconstructed motion field after 5 iterations of the rigremoval algorithm. These motion vectors are superimposed on the reconstructed image. Interpolated vectors now cover the rig region, and are mostly correctly estimated. There are some small pixel errors at the edge of the reconstructed object due to the difficulties in imposing smoothness constraints at that sharp edge. Nevertheless, this does not strongly affect the reconstruction.



**Fig. 6.** Odd Rows: The user defined matte that delineates the rig is overlaid in red. Even rows: The corresponding results of rig removal. See [www.mee.tcd.ie/~sigmedia/postpro](http://www.mee.tcd.ie/~sigmedia/postpro)



**Fig. 7.** Leftmost: Frame 2 with original motion (blue) and interpolated motion (red). Next three pictures: First three consecutive rig removed frames from a forward pass of the algorithm. Note progressively more of the 'rig' is removed with time.