

Filling in the gaps

by Bill Collis and Anil Kokaram

The common view of film and video special effects is that it involves the use of computer graphics to synthesise virtual scenes or new objects in a film or video production. The viewer can be presented with scenes that could never be filmed in this world, such scenes of rampaging robots in 'Terminator 3', or scenes that are apparently real but contain objects or features or physics that could never exist, for instance, the fight scenes in 'The Matrix'.

What is surprising is that, in the film and video postproduction community, there exists a host of regularly executed tasks that do not fall into this domain. Some of the most convincing film and video effects are created in digital post-production by removing the apparatus, or rigs, that support actors or manipulate objects from view.

Another, now accepted, task is view synthesis, or in-betweening. It was popularised by the makers of 'The Matrix', but was being used by TimeSlice Films, a British post-production company, for some time before that. The idea is to use a host of synchronised cameras to film a scene. During playback of that scene, it is then possible to change the viewing angle instantaneously and sweep through several intermediate angles while doing so.

The effect is as if the camera were able to change position instantaneously with infinite velocity. To create a convincing in-betweened shot, enough views must be used for the shot to look smooth. As cameras can rarely be positioned close enough to allow this, new images must be synthesised from viewpoints in-between the existing camera views.

Synthesis of entirely new images from an existing sequence is also required for slow-motion effects. This is a matter of converting the frame rate from typically 24 frames per second to something much higher. Synthesis is also used for standards conversion, such as when film running at 24 frames per second needs to be converted to video at 30 frames per second.

MISSING DATA

All of the tasks considered above are instances of dealing with missing data. The solution to these problems is to use motion interpolation. For instance, in the case of rig removal, as the rig traverses the scene, it uncovers and reveals image material. If it were possible to estimate the motion of the scene covered by the rig then it would be possible to paste in the revealed and occluded areas from previous and next frames as the rig moves. Using in-betweening, given a motion field mapping every pixel in

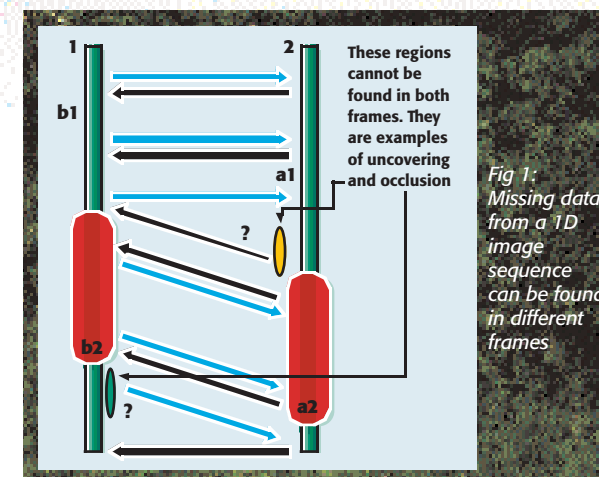
the missing image to pixels from the surrounding views, it is possible to cut and paste pixels from those images and so build the missing frame.

Where the information is known to exist somewhere in the sequence, motion interpolation will normally provide the most reliable reconstruction method. Typically, in an image sequence, nothing much changes between consecutive frames, except that objects can move around. This idea can

be used to design a model that describes how pictures are related to each other. In simple terms, the model states that an image can be built by rearranging the locations of pixels in the previous frame. This model is illustrated by the 1D image sequence shown in Fig 1. Here two frames are shown, and it is clear that the points a1, a2 in frame 2, for instance, can be found at locations b1, b2 in frame 1. The estimation of the displacement given a pair of images is what is known as motion estimation.

Unfortunately, motion estimation cannot work for individual pixels. There are two variables – the horizontal and vertical components of motion – and just one equation at each pixel. The equation cannot be solved using one pixel site alone. Perhaps the simplest mechanism for getting around this problem is to assume that the motion is the same over small blocks in the image. A block size of 8x8 pixels is typical. Within such a block therefore, there are 64 equations and two variables. Several techniques can be applied to estimate the motion.

The simplest technique for estimating motion is block matching. Up to an estimated maximum displacement, the block is tested against blocks at each possible displacement to see how well it fits the model. Although computationally intensive, especially for fractionally accurate motion, →



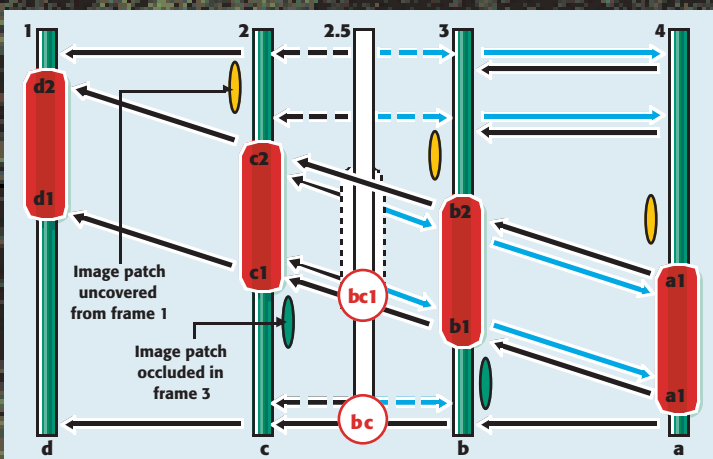


Fig 2: In-betweening synthesises data using adjacent frames

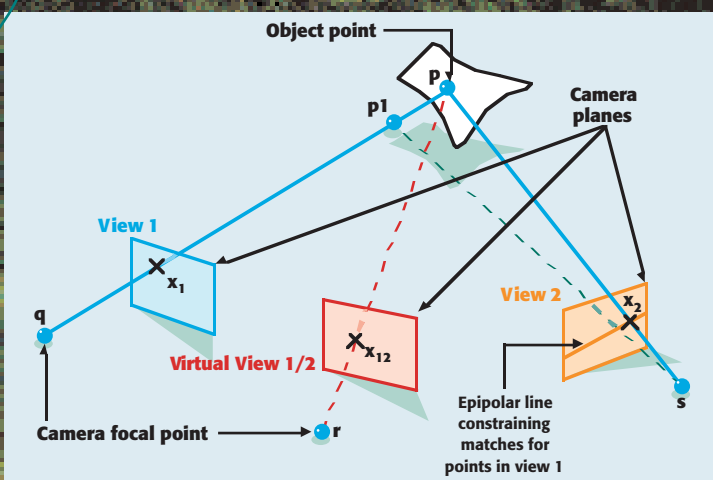


Fig 3: In-betweening is also used to interpolate between different camera views

Bayesian Inference generally yields more flexible solutions.

Regardless of the framework, encoding notions such as discontinuities remains difficult. To introduce occlusion into the motion framework, typically a new variable to represent occlusion is introduced at each pixel site. It is set to 1 if the data at that site in frame n cannot be found – that is, occluded – in frame $n-1$ and it is set to 0 otherwise.

In-betweening covers two distinct uses: where frames are synthesised to increase the frame rate (Fig 2) or are used to interpolate between views from different cameras (Fig 3). At first glance, the situations shown in Fig 2 and Fig 3 are only related by the trivial observation that they involve the creation of new image data where none existed before. That is, they both involve missing data. However, Fig 2 focuses on an image-centric problem and Fig 3 is certainly a 3D problem. In fact, the complete solution to the view interpolation problem requires recognising that, by identifying correspondences between pixels in the two views – for example, identifying that the pixels x_1 and x_2 match, the position in 3D space of a point can be estimated. Thus, given the orientation of the virtual viewpoint required, the position of the interpolated point can be estimated.

CHOOSING THE RIGHT DIMENSIONS

Identifying image point correspondences is a tricky problem but the situation is simplified by observing that, given a point, the camera geometry constrains the object point to lie along a line in 3D space. This means that the corresponding image point in the second view must lie along a line in that image. This line is called the epipolar line. Estimating this line orientation has occupied the computer vision community for quite some time. However, we find that regardless of the view geometry, in practical circumstances, the differences between the view orientations are not large.

In fact, regardless of the differing orientations, the result of acknowledging camera geometry is only to confirm that the position of each pixel in each view is related to the position of every other corresponding pixel in other views by some not-necessarily linear transformation. Once that is accepted, then it follows that the view interpolation situation in Fig 3 is identical to the situation in Fig 2. There is much to gain from operating somewhere inbetween the image-centric 2D world and the vision-centric 3D world. The computer vision community has labelled this kind of activity ‘image-based rendering’.

Both motion and occlusion information needs to be estimated at each site of the in-betweened image. If the pixel is not occluded in either direction, the interpolated pixel is just the average of the motion compensated pixel in the previous and next frames. If the pixel is occluded in one or the other direction, then the interpolated pixel comes from either the previous or the next frames. A simple heuristic for generating the missing motion information is to first

this algorithm is very simple and easily mapped into hardware, and is therefore the choice for the video compression community.

There are problems with an approach based on the displacement of pixel blocks. It cannot be expected that two different objects should undergo the same motion transformation from frame to frame, unless they are connected in some way. Therefore real world motion fields would tend to have discontinuities in space typically coinciding with object boundaries. Blocks that lie across boundaries cannot yield useful motion estimates.

Further, Fig 1 shows that not all of the points in consecutive images can be associated with each other. This occurs at the boundaries of moving regions and is caused as objects interact and move in front of or behind each other. These regions are called regions of occlusion in this paper. Estimating the location of these regions is important for creating believable pictures. In most cases, if a motion-based special effect does not allow for occlusion, the result is that background gets pulled behind foreground moving objects and generally results in a visible artifact.

Building motion estimation processes that cope with these realities results in better motion fields, and is the subject of much current research. An approach based on



“An approach based on Bayesian Inference generally yields more flexible solutions”

Fig 4 (far left) and Fig 5 (left): Two examples of frames from a multiple-view camera rig. The odd rows are original frames, and the even rows are interpolated frames.

Fig 6 (below) The first two images are frames from a multiple view camera rig. The third frame has been created by blending the original frames. The last was created using motion interpolation. Note the doubling imaging and blurring of the blended frame as compared with the sharp motion-interpolated result.

estimate motion between the existing image frames, and then divide each vector by two to assign that motion to some halfway point in the inbetweened image. Unfortunately, this heuristic is unable to generate a realistic motion field that obeys other constraints.

A Bayesian approach is better able to articulate all aspects of this complex problem. By operating in a probabilistic sense it is possible to express this problem in terms of solving for each piece of motion information at each pixel site iteratively. Consider a site in frame 2.5; and assume that there already exists a current guess for the solution at all other image sites. The problem is then to manipulate the probability density function for our variables

at that site, both backward and forward in time, given the image data and the current state of the motion information in the immediate neighbourhood.

There is some effort in massaging these concepts into efficient algorithms and there are a number of solutions that the Bayesian approach yields. However, the basic principle is to kick start the estimation process using a simple heuristic and then use the Bayesian framework to refine this initial estimate.

Figs 4 to 6 show results from inbetweening images using the methods discussed, and the interpolated images are convincing. What is particularly encouraging is that, without any 3D information, it is possible to treat both →



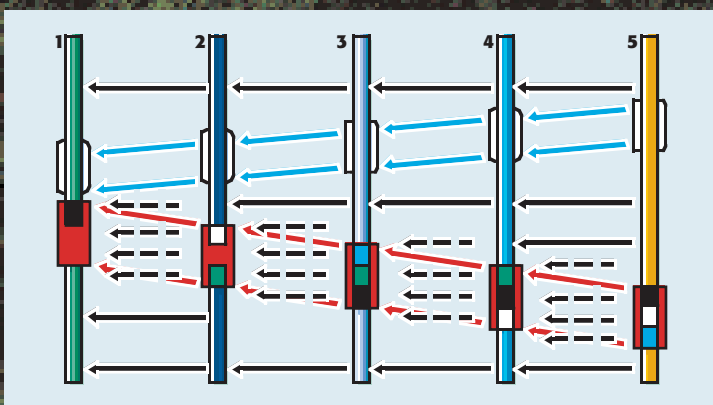
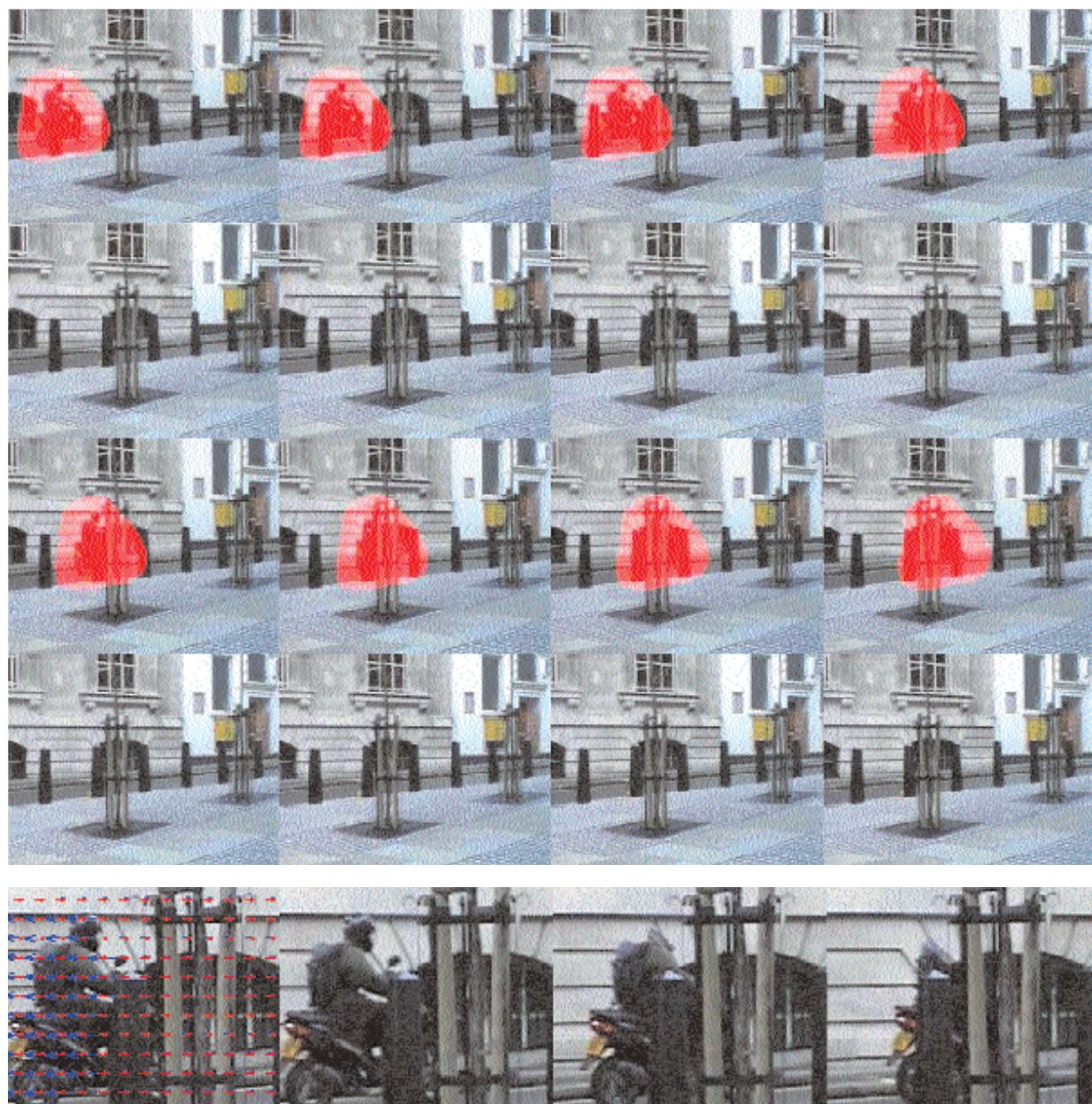


Fig 7: Rig removal process, with the red area being the rig that needs to be removed

the view interpolation and frame rate conversion problems within the same image-based framework.

Fig 7 shows the basic idea of rig removal. It shows 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

Fig 8 (below): The first and third rows show a matte used to define the rig. The second and fourth rows show the result. Fig 9 (base of page): Progressive removal of a rig over time from frame 2 of fig 8.



rig obscured region in each frame.

Fig 7 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 a bit of frame 1 (this is shown as a green 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 white patch). In frame 3 the same concept allows more of the rig to be removed.

After just three frames in this case, 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. 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.

VALIDATION OF PIXELS

The key to rig removal is to recursively reconstruct the motion field. To do so requires knowledge about what part of the motion field in previous and next frames is valid. This means that a weight is attached to every pixel in the image. It is set to 1 where the pixel is not covered by the rig, and it is 0 in the rig region. The user must roughly delineate a garbage matte for this to be achieved. Having done this, a Bayesian approach almost identical to that discussed for inbetweening can be used. The only difference is alterations have to be made to take into account the fact that some data in the previous and next frames is invalid and should not be taken into account.

Fig 8 shows results, using 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. Fig 9 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.

There are occasions where there is no motion that can assist the missing data reconstruction. For instance, Fig 10 shows an example of crowd duplication. Here, we want to reconstruct more crowd to fill the empty parts of the stadium. In situations like this, the only mechanism is to synthesise data spatially. This area of study is known →

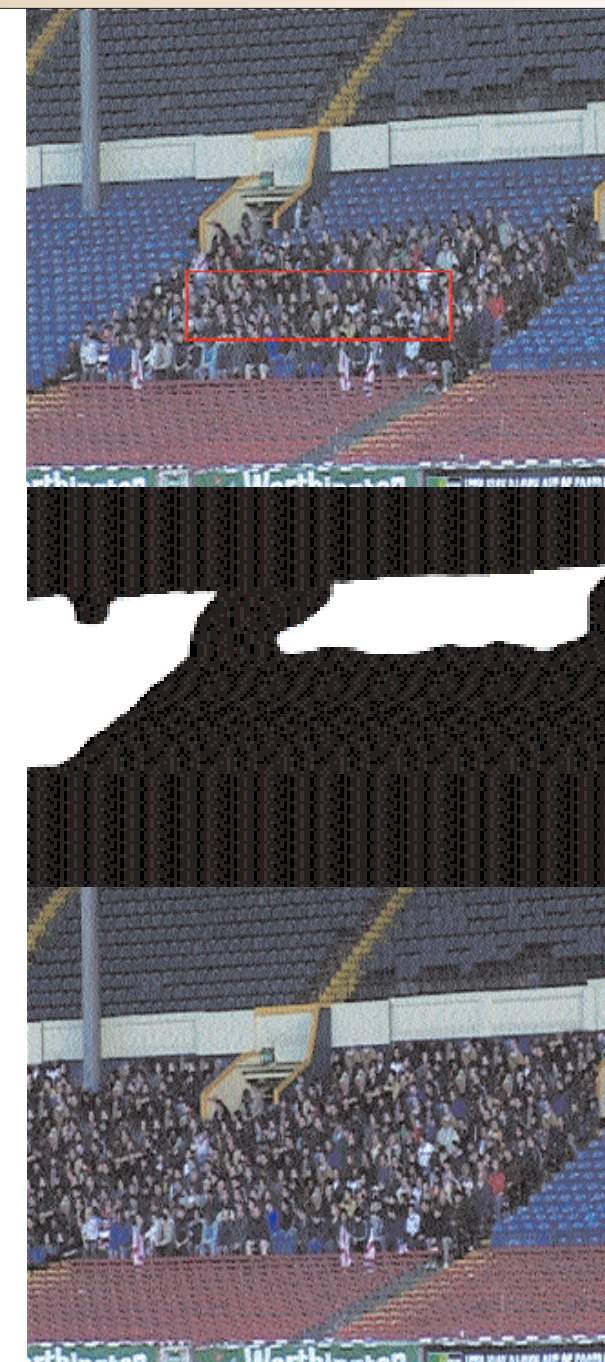


Fig 10: Source crowd (top) with a sampling region defined by the red box, matte defining the region of stadium to be filled (middle), and the resulting image created by inpainting (bottom)

“ The key to rig removal is to recursively reconstruct the motion field ”

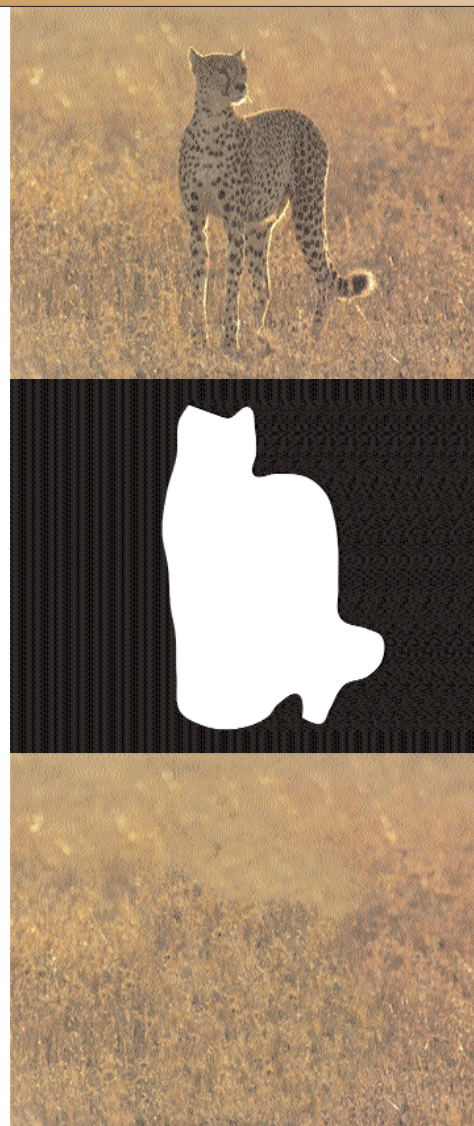


Fig 11: Use of inpainting to remove an animal (top) from view, with a matte (middle) and the final view (bottom)

as texture synthesis or inpainting. The most successful of the inpainting algorithms work by simply sampling and copying pixel values from a source image to create the new texture.

In the simplest form of inpainting, the user defines two regions, a fill region and a sample region. The algorithm proceeds by filling pixels in the fill region in an onion skin scan order until the region is completely filled. The key to the success of the technique is a reliable way of measuring the probability distribution function of the colour values at a pixel. Rather than model this function mathematically the novel step is to measure it empirically. To select a pixel to be filled a small patch around the pixel (the source patch) is compared with similar sized patches (sample patches) in the sample region. Sample patches that match well to the source patch are used to generate the probability distribution function.

Although texture synthesis algorithms work well at duplicating pure texture, most real world scenes involve a mixture of structure and texture. A useful inpainting technique must be able to reconstruct missing structures, for example edges, that cross the region to be filled. A

“The algorithms described are now in widespread use within the post-production community”

number of approaches have been tried to ensure edge completion. The most common are diffusion-based techniques. Their main failing is that the diffusion process tends to produce a soft, blurred result. A possible solution is to attempt to combine texture- and diffusion-based techniques, using diffusion for the low-frequency components of the image and texture replication for the high frequency.

One of the major difficulties with inpainting is to ensure temporal consistency of the filled region through the sequence. This is relatively easy for local motion, but is still an ongoing research topic for large motion.

As we have seen, there are a number of important effects used in the post-production industry that can be unified through an understanding of automated missing data treatment. This framework is flexible and allows the decomposition of an apparently complex problem into a number of easily-manipulated, simpler sub-problems.

The algorithms described in this article have been implemented as plug-ins by the authors and are now in widespread use within the post-production community. Film credits include the ‘Matrix’ trilogy, the ‘Harry Potter’ films, the ‘Lord of the Rings’ trilogy, and many others. However, part of the success of these techniques is that the average viewer is unaware of their use in a film.

As image and video processing researchers, we find it encouraging to note that many of the more recent sophisticated probabilistic ideas have valuable application in as popular an area as film special effects, and the future lies in exploiting more image and vision based constraints for making the impossible possible. ■

Bill Collis is based at film effects specialist The Foundry and Anil Kokaram is based at Trinity College, Dublin. The authors would like to thank Tim Macmillan at TimeSlice Films for the multiple-view camera images. The crowd images are courtesy of The Moving Picture Company from the film ‘Mike Bassett: England Manager’ (2001), Film Council, Hallmark Entertainment and Entertainment Film Distributors.