New-Generation Scalable Motion Processing from Mobile to 4K and Beyond

White Paper

Today’s broadcast video content is being viewed on the widest range of display devices ever known, from small phone screens and legacy SD TV sets to enormous 4K and 8K UHDTV displays. The growth in size and resolution is happening alongside many other improvements, in grey-scale resolution, colorimetry, 3D and, especially, higher frame rates.

These developments mean that the requirements for very high quality, artefact-free conversion in resolution and frame rate have become more important than ever. The challenge is given a further dimension by the wider range of content that can appear on large screens, from upconverted archive footage to the much more detailed, wider window on the world made possible by the new large formats.

This paper presents cutting-edge algorithms for motion compensated processing to meet these challenges in both live TV and file-based operation. One size no longer fits all, so this paper also discusses how to achieve a balance across the range of processing complexity and performance, showing how the trade-offs can be managed gracefully and optimally.

Introduction

How will you watch your next TV programme? Will it be on a small phone screen, a head-up display, a tablet, an old CRT TV, a PC monitor, a modern HDTV display, a projector, a 4K or an 8K UHDTV display? And where will the content have come from? A mobile phone video, an old SD TV drama, an HDTV production studio, a digital film master, or a 4K or 8K camera? We rightly expect seamless transfer of content from all those sources to all those destinations, and for differences in colorimetry, dynamic range, resolution, interlace, aspect ratio and frame rate to be dealt with efficiently, without loss of image quality or visual impact.

In previous IBC papers, we have looked at HDTV standards conversion [1], interlaced and progressive signals [2] and novel ways of processing material for smaller and different-shaped displays [3]. Those technologies and algorithms continue to be relevant. However, in recent years the question of field or frame rate has become increasingly important, as interest has grown in conversion not only between the standard field rate has become increasingly important, as interest has grown in conversion not only between the standard field rate but also from and to 24Hz and newer film frame rates such as 48 Hz, and higher frame rates in cameras and displays such as 100Hz, 120Hz, 300Hz and beyond. One particular example of interest is conversion from 24Hz film to 50Hz and 59.94Hz frame rates, in a world that is becoming increasingly intolerant to the motion judder resulting from conventional 2:2 and 3:2 pulldown methods of conversion.

Motion compensated processing has long been considered essential for high-quality frame-rate conversion. However, the massive increases in screen size, resolution and display brightness have all put pressure on the previous generation of motion compensated algorithms. A step change in motion compensation technology is required to meet these new demands. At the same time, cost pressures on programme production and distribution in multiple formats are bringing a requirement for greater flexibility in allocation of resources to tasks such as conversion in both live and file-based applications.

This paper presents a new generation of algorithms for motion compensated processing. First, we look at a particular problem that has emerged as the range of source and display resolutions increases, to describe which we have adapted the term “wow factor”, and which the new algorithms are particularly suited to address. We then look at developments in the two main components of motion compensated processing: motion estimation and picture building. Finally, we introduce the concept of a single “knob” which can be used to control the trade-off between processing speed and conversion quality, and discuss how to perform scalable load balancing using available processing resources across varied input picture content.

Window on the World

Range of Resolutions

The range of source and display resolutions we might encounter is now very wide. A small mobile phone might have as few as 0.1 megapixels, while with 8K UHDTV we have 32 Mpixels, a ratio of 320:1. At any display resolution, it is important to ask ourselves where the source has come from, in particular what resolution it was captured at, and also what production techniques were used. We shall now look at these questions with particular reference to pictures that are displayed at high resolution, taking 8K as an example.

Low-Resolution Source

A source at a low resolution, for example standard definition, will normally be upconverted if it is to be displayed at high resolution. Typical SD camera techniques involve zooming-in quite close to the subject. Any motion in the source will be, in pixel terms, faster in proportion to the degree of upconversion in each dimension, and the large picture will cover a fairly small viewing angle in the original scene and will be relatively soft.

High-Resolution Source

If the source is at high resolution, it will be displayed unchanged on the high-resolution display, and the characteristics of what is displayed depend on the...
production technique. If, on the one hand, the camera is used as if it were a low-resolution camera, the picture will have the same characteristics as one from the low-resolution source. On the other hand, the viewing angle of the camera could be widened so that the high display resolution is fully exploited, in which case the picture will typically contain more detail, smaller objects and lower motion speeds.

The “Wow Factor”
We propose a simple rule of thumb for expressing the different possible picture characteristics seen in high-resolution displays. The “wow factor” (window on the world) indicates the degree to which increased display resolution is exploited to give the viewer a wider view of the scene. An example showing the relationship between display format, upconversion ratio and wow factor is shown in Figure 1. The diagram shows that, as the display format grows, the range of possible wow factors increases. Table 1 summarises qualitatively the effect of the wow factor on parameters relating to motion compensated processing.

<table>
<thead>
<tr>
<th>Wow factor</th>
<th>low</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharpness</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Object size</td>
<td>large</td>
<td>small</td>
</tr>
<tr>
<td>Motion speed</td>
<td>fast</td>
<td>slow</td>
</tr>
<tr>
<td>Motion variation</td>
<td>narrow</td>
<td>wide</td>
</tr>
</tbody>
</table>

Table 1 – Effects of “wow factor”

Motion Compensation
This analysis has unveiled a problem that occurs when it comes to scaling up a motion compensated processing algorithm for larger display formats. If the wow factor remains low, the processing will have to cope with fast motion of blurred objects. If it is increased, the processing will have to cope with small, detailed objects. Of course, in reality we have to cope with the full range of wow factors, which doubles for every doubling of the display resolution. This means that scalability of motion compensated processing becomes a multi-dimensional affair and will not be handled satisfactorily by any single scaling up of an SD or HD system.

We now discuss the effect of these observations on motion estimation (the analysis part of motion compensated processing) and on picture building (the synthesis part) in turn.

New-Generation Motion Estimation
So how do we design an improved and scalable motion estimator? Here we introduce some of the new approaches we have made towards a fully scalable algorithm.

After several decades of research, methods of motion estimation [4] still largely fall into the categories of block matching, gradient or optical flow methods [5], frequency-domain methods such as phase correlation [6], and feature-based methods. The new suite of algorithms presented here, code-named “Mensa”, makes extensive use of the first three categories, while work is proceeding on introducing the fourth category into the mix.

Multi-Scale Candidate Vector Generation
Our existing motion estimation technology makes use of phase correlation to analyze a scene and to generate candidate motion vectors for subsequent assignment to individual pixels. The phase correlation is based on large blocks, whose size is a trade-off between motion range and ability to handle small or detailed objects. We have seen that both are required, so Mensa uses multiple block sizes in parallel to generate candidate motion vectors of both kinds.

Gradient-Based Refinement
One disadvantage of phase correlation is its fundamental inability to handle smooth variations of motion within objects, such as zooms, rotations and perspective in a receding landscape. Where the “wow factor” is low, this does not pose too great a problem, because the degree of motion variation is also low. But for high wow factors these variations can become quite large. We have solved this problem by allowing the candidate motion vectors to vary slowly from pixel to pixel, using gradient based techniques to refine the vectors from initial constant values.
One of the drawbacks of gradient-based vector refinement is that it fails at motion boundaries. We overcome this problem by using overlapping blocks, with a weighting system to encourage refinement to work hardest in areas where the vector field is already performing well. This has the effect that one overlapping block covering a motion boundary will refine a vector field suitable for one side of the boundary but not necessarily the other, while another overlapping block will refine a field for the other side.

Vector Assignment
The final step in motion estimation is to assign a motion vector to every pixel from the set of refined candidates. The classic way to assign motion vectors is to calculate an error surface for each candidate, usually a displaced frame difference (DFD). This surface is then filtered spatially so that the error generated by a candidate for a pixel contains information about the neighbourhood of the pixel. It is difficult to choose the right size for this filter: too small, and the vector field is noisy; too large, and small objects can be missed and behaviour at motion boundaries is poor. For Mensa, we have developed a nonlinear DFD filter based on splitting the neighbourhood into octants, as shown in Figure 2, and applying a minimax approach that allows a motion boundary to pass through the neighbourhood while retaining the stability of a large filter.

Wavelet Picture Building
It is possible to manage the appearance of these artefacts and to reduce their overall visibility by employing a “wavelet picture builder” in which the output picture is built up in sub-bands with suitably scaled and downconverted motion vectors at each stage. A simplified example of one layer of the Mensa wavelet picture builder is shown in Figure 3. A feature of this approach is that holes in the projections are automatically filled from coarser layers.

Temporal Phase Control
There are some kinds of picture material that will defeat even the most reliable motion estimators and the most benign picture builders. It is prudent to have recourse to some kind of “fallback mode” which is applied when such picture material is encountered. Crucial to the usefulness of a fallback mode is a reliable metric that will determine when and to what extent it should be applied. Our metric is based on the assignment errors, and the principle of our fallback mode is to build pictures that are closer in time to the input pictures.
The Mensa Knob
In this section, we turn to some work on optimizing the cost/performance trade-off of a complex machine such as the standards converter described above. When a standards converter is implemented in hardware, the full resources of a complex algorithm can be applied without inefficiency (except possibly in electrical power) to both demanding and easy picture material. But in an implementation based on software, processing time and the number of processors required are directly measurable as a processing cost, and it becomes beneficial to tailor the processing to the content.

Streaming and File-Based Processing
Some applications of video processing are designed for real-time streaming, usually with a limit to the permitted latency. Others are file-based and may work faster or slower than real-time. In both cases there is scope for optimization of the performance/cost trade-off, though the possibilities are greater in the case of file-based processing. For a given set of content, there may on the one hand be a limit to the time and processing resources available, and the goal is to maximize the quality of the output pictures. On the other hand, there may be a required minimum quality level, and the goal is to minimize the processing time or number of processors used in order to save time and money. But even for live streaming, it may be possible to concentrate resources on locally more demanding parts of a video stream.

The Efficiency Cloud
A conversion algorithm such as Mensa is controlled by a multitude of parameters. Some of them, such as thresholds or gain factors, will typically only affect performance and have no impact on processing time. These can generally be optimized in a straightforward manner, given a suitable performance metric, though it may be worth repeating the optimization process for different genres of input material, for example sport or news. Other parameters, such as numbers of candidate motion vectors or of vector refinement iterations, will generally affect both the performance and the processing time. The interactions between these parameters can be bewilderingly complicated, making it very difficult to control the performance/cost trade-off. Figure 4 shows the results of processing a test sequence with hundreds of combinations of control parameters. The x-axis represents the processing time (the scale is arbitrary) and the y-axis represents a performance error measure, in this case the RMS error between the output sequence and a known “ground truth” sequence. Note the false origin on the y-axis, highlighting the fact that small (though visible) performance improvements are generally only obtained at the cost of substantial increases in processing time. A few points extend above and to the right of the cloud shown.

It would be highly desirable to reduce the set of adjustable parameters to just one: a single controller or “knob” which could be adjusted between relatively high

errors but low processing cost, and low errors but high
processing cost. This could be achieved by selecting a subset of points in the cloud that span the range of performance and processing time but which are in some sense optimal. Looking at Figure 4, it becomes clear that some parameter selections are less efficient than others. For example, point A has both a higher processing time and RMS error than point 5, so within the assumptions we have made, point A would be of no use in a knob. Suitable points would be those that are on the approximately hyperbolic envelope of the left and bottom of the cloud.

A Knob That Goes To 11
Figure 4 shows a labelled subset of points that follow the envelope and which would therefore make good candidates for a performance knob. Point 0, which is non-motion-compensated conversion, and point 1, a very simple motion compensated algorithm, fall well above the top of the plot. The fact that the knob settings extend to 11 is a serendipity, echoing the scene in the cult 1984 film This is Spinal Tap in which a joke is made about amplifier knobs that “go to 11” rather than the standard 10.

Each knob setting maps to a selection of parameter choices, and it is now possible to make adjustments between high performance and high speed, knowing that each setting is performing at optimum efficiency.
Scalable Load Balancing

The above analysis is based on an ensemble of test material of varying degrees of difficulty. In practice, the performance of a particular knob setting will depend on the source material. Whether our goal is to minimize overall error given a processing time limit, or to minimize processing time given a maximum acceptable error, we need an algorithm that links some measure of source difficulty to the knob setting.

If we repeat the analysis for different sources, we would obtain a set of different “knob” curves, as illustrated in blue in Figure 5. Note that the y-axis now represents mean square error, so that errors can be added across all the sources.

Suppose for each source i the mean square error $e$ is linked to processing time $t$ by a function $e = f_i(t)$.

If each source has $M_i$ frames, then the total error is

$$ E = \sum_{i=1}^{N} M_i f_i(t) $$

and we wish to choose $t_i$ the processing time per frame for each source, to minimize $E$ subject to a total processing time constraint:

$$ T = \sum_{i=1}^{N} M_i t_i $$

Using the method of Lagrange multipliers, the equations to solve are:

$$ \frac{\partial E}{\partial t_i} + \lambda \frac{\partial T}{\partial t_i} = M_i \left( \frac{\partial f_i}{\partial t_i} + \lambda \right) = 0 \quad \text{so} \quad \frac{\partial f_i}{\partial t_i} = -\lambda $$

which just means that we have to choose points on each function where all the gradients are the same, as shown by the red lines, and the choice of gradient will be such as to meet the total processing time limit. Similar reasoning would apply to meet an error constraint.

The remaining problem is to find out, given real picture material, which curve is appropriate for each source segment. We no longer have ground truth, and we certainly cannot afford to try out different knob settings, so we have to gather evidence by taking measurements on the source pictures. For example, we can calculate the average frame-to-frame difference of each segment. It turns out that there is a reasonable correlation between such a simple measure and the “knob function”. This allows us to choose an appropriate knob setting for each segment in order to optimize the overall cost/performance trade-off.

Figure 6 shows a comparison between this load-balancing approach and a fixed knob setting with the same overall processing cost. The graphs show the RMS error for a three-minute section of a 1960s spaghetti western film when converted from 24 to 60 Hz using knob settings at the lower end of the processing quality scale. For the purposes of this illustration, the “ground truth” is taken to be the output of knob setting 11, a technique which turns out to be remarkably useful when evaluating the lower-quality settings.

Conclusions

In this paper, we have introduced a new generation of motion compensated processing algorithms suitable for the very wide range of source and display resolutions now encountered, and have described how they can be controlled in such a way as to optimize the performance/cost trade-off in both streaming and file-based processing.

References


Acknowledgements

The author would like to thank the Directors of Snell Ltd. for their permission to publish this paper, and his Snell Technology Development Algorithms team colleagues for their valuable contributions, suggestions and support. Intellectual property disclosed in this paper is the subject of patent applications and granted patents in the UK and elsewhere.