Problems on Video Coding

Guan-Ju Peng

Graduate Institute of Electronics Engineering, National Taiwan University

1 Problem 1

How to display digital video designed for TV industry on a computer screen with best quality? .. Hint: computer display is 4:3, 1280x1024, 72 Hz, progressive.

Supposing the digital TV display is 4:3 640x480, 30Hz, Interlaced. The problem becomes how to convert the video signal from 4:3 640x480, 60Hz, Interlaced to 4:3, 1280x1024, 72 Hz, progressive.

First, we convert the video stream from interlaced video to progressive signal by a suitable de-interlacing algorithm. Thus, we introduce the de-interlacing algorithms first in the following.

1.1 De-Interlacing

Generally speaking, there are three kinds of methods to perform de-interlacing.

1.1.1 Field Combination Deinterlacing

- Weaving is done by adding consecutive fields together. This is fine when the image hasn't changed between fields, but any change will result in artifacts known as "combing", when the pixels in one frame do not line up with the pixels in the other, forming a jagged edge. This technique retains full vertical resolution at the expense of half the temporal resolution.

- Blending is done by blending, or averaging consecutive fields to be displayed as one frame. Combing is avoided because both of the images are on top of each other. This instead leaves an artifact known as ghosting. The image loses vertical resolution and temporal resolution. This is often combined with a vertical resize so that the output has no numerical loss in vertical resolution. The problem with this is that there is a quality loss, because the image has been downsized then upsized. This loss in detail makes the image look softer. Blending also loses half the temporal resolution since two motion fields are combined into one frame.

- Selective blending, or smart blending or motion adaptive blending, is a combination of weaving and blending. As areas that haven't changed from frame to frame don't need any processing, the frames are weaved and only the areas that need it are blended. This retains full vertical resolution, half the temporal resolution, and has fewer artifacts than weaving or blending because of the combination of them both.

- Inverse Telecine: Telecine is used to convert a motion picture source at 24 frames per second to interlaced TV video in countries that use NTSC video system at 30 frames per second. Countries which use PAL at 25 frames per second do not use Telecine since motion picture sources are sped up 4

- Telecide-style algorithms: If the interlaced footage was generated from progressive frames at a slower frame rate (e.g. "cartoon pulldown"), then the exact original frames can be recovered by copying the missing field from a matching previous/next frame. In cases where there is no match (e.g. brief cartoon sequences with an elevated frame rate), then the filter falls back on another deinterlacing method such as blending or line-doubling. This means that the worst case for Telecide is occasional frames with ghosting or reduced resolution. By contrast, when more sophisticated motion-detection algorithms fail, they can introduce pixel artifacts that are unfaithful to the original material. For telecine video, decimation can be applied as a post-process to reduce the frame rate, and this combination is generally more robust than a simple inverse telecine (which fails when differently interlaced footage is spliced together).

1.1.2 Field Extension Deinterlacing

- Half-sizing displays each interlaced frame on its own, resulting in a video with half the vertical resolution of the original, unscaled. While this method retains all vertical resolution and all temporal resolution it is understandably not used for regular viewing because of its false aspect ratio.

- Line doubling takes the lines of each interlaced field (consisting of only even or odd lines) and doubles them, filling the entire frame. This results in the video having a frame rate identical to the field rate, but each frame having half the vertical resolution, or resolution equal to that of each field that the frame was made from. Line doubling prevents combing artifacts but causes a noticeable reduction in picture quality since each frame displayed is doubled and really only at the original half field resolution. This is noticeable mostly on stationary objects since they bob up and down. This technique is also called bob deinterlacing for this reason. Line doubling retains horizontal and temporal resolution at the expense of vertical resolution and bobbing artifacts on stationary and slower moving objects. A variant of this method discards one field out of each frame, halving temporal resolution.

1.1.3 Motion Compensation

The best deinterlacers combine all of the methods mentioned above, both field combination and frame extension. This technique is often called motion compensation. Deinterlacers that use this technique are often superior because they can use information from many fields, as opposed to just one or two. For example, if two fields had a person's head moving to the left, then if weaving was applied, mouse teeth would appear. If blending was applied, ghosting would appear. Selective blending would also create ghosting. Both of the frame extension methods would have no artifacts and would be the best selection for this motion section of the scene. Advanced motion compensation (ideally) would in addition see that the face in both fields is the same, just transposed, and would combine the face (i.e. through image stacking) to get full detail in both output frames. Doublers as above don't provide combined field resolution in this form. This technology would need to be combined with a scene change detection algorithm, otherwise it will attempt to find motion between two completely different scenes. In the areas that it cannot find a motion match, it could fall back on selective blending. If frame rate was to be preserved it could fall back on doubling. The best de-interlacers, (In the case of NTSC) also determine whether video material source was from film by checking for a 3:2 pulldown Telecine sequence. They automatically do a reverse telecine instead of the above deinterlacing techniques in this case. This operation is more automatic on modern deinterlacers than it used to be.

1.2 Temporal Interpolation

After the de-interlacing process, the video signal becomes 48Hz progressive if the Inverse Telecine style de-interlacing is applied or 60Hz progressive if other de-interlacing algorithm is applied. In order to reach resolution with 72Hz progressive, an interpolation procedure in temporal domain is required. We introduce a zonal algorithms for temporal interpolation in this section.

Generally speaking, zonal based algorithms can achieve a smoother motion vector field, and relatively accurate motion vectors, which are also closer to the true motion. Since adjacent blocks are highly correlated, and thus, in reality, tend to have similar motion vectors. Such a property would suggest that, in most cases, the closer to the true motion field an algorithm gives, the smaller the entropy versus a set of possible predictions would be as well. Such predictors could include the motion vectors of the three spatially adjacent blocks on the left, top, and top right to the current position, their Median, the (0,0) motion vector, and even the motion vector of the collocated block in the previous frame. All these predictors constitute a Predictor Set, and are quite important in the performance of zonal algorithms.

We wish to predict a frame k by using currently available frames (k+2) and k. Let us assume that macroblock B_1 at position (x,y) inside frame (k+2) has a motion vector (dx,dy). We may predict the pixels of the missing frame (k+l) by considering the block at position (x+dx/2, y+dy/2), where x and y are basically the coordinates of each Macroblock (MB) in a frame. What we need to do is to essentially track block B_1 and estimate where this block may be found inside the missing frame. To further improve the performance we may generate block B_3 in the missing frame, as the average of B_1 and B_2. Using this method it is evident that several pixels in the inserted frame are generated by more than one block, whereas there might also be pixels that have no corresponding reference, thus leaving empty gaps inside the inserted field frame. For the former, the average of all corresponding pixels is selected, whereas for the latter, blank pixels can be predicted by either assuming zero motion or by using other predictive techniques. Such a method could be considered a bit complicated but relatively accurate.

A less complicated technique, but nevertheless a not so accurate one, which could also be used for predicting the values of the blank pixels. We may instead assume that the macroblock B_3, at the corresponding (x,y) position inside the missing frame has a motion vector equal this time to (dx/2,dy/2) pointing inside the kth frame (block C_{3B}).

We may also assume that this same block has a second motion vector, this time equal to (-dx/2,-dy/2) which this time points to the (k+2)'h frame (block C_{3F}). Thus B_3 can be reconstructed as the average of blocks C_{3B} and C_{3F}. Thus is rather similar to the way B frames are reconstructed, and its advantage is its relative simplicity, especially for hardware, since we may reuse the standard motion compensation architecture.

1.3 Spatial Interpolation

After the procedure of de-interlacing and temporal interpolation, the only thing left is the spatial interpolation which up-samples all 640x480 frames to 1024x768. In order to preserve the original scale of the content, each 640x480 frame will be first up-sampled by a factor of 8 into a 5120x3840 frame, then the frame is down-sampled by a factor of 5 into 1024x768. Note that we will perform a smooth process before the down-sampling to preserve the high frequency signal and the up and down sample filters should match to preserve the most of the original signal.

1.4 Other Issues and Conclusion

Usually, the format of a DVD/VCD video after deciding is YUV 4:2:0 and encapsulated into YV12 format. If the video card on the computer supports the display of the YV12 format, the video card can direct convert the YUV 4:2:0 to YUV 4:4:4 and RGB32. Otherwise the conversion procedure should be accomplished by the software.

Current high definition digital tv usually supports the resolutions in the following table.

NA	spatial	Temporal	Interlace/Non-Interlace
FHD	1920x1080	30Hz/60Hz	Progressive or Interlace
HD	1280x720	30Hz/60Hz	Progressive or Interlace
SD	720x480	30Hz/60Hz	Progressive or Interlace

Generally, the conversion procedure between TV and VGA video signals follow the similar procedure which are de-interlacing, temporal up-sampling and spatial up-sampling. However, many experiments should be applied to find the best tools for the conversions, which are not included in the article.

Our conclusion is that the major factors that affect the quality of conversion are the methods of de-interlacing, spatial, temporal filters and the smooth kernel. Only the suitable tools are chosen, to achieve the best conversion quality is possible.

2 Problem 2

Design/explore/describe a fast mode decision method to efficiently decide prediction mode for interlace contents in MPEG-2 video. Explain why it is effective, and how much computation can it reduce?

Supposing the coding sequence and the frame type is pre-determined before the mode decision. Thus the mode decision problem occurs when choosing the inter prediction mode of P and B frames. In MPEG2, there are three modes available for interlace contents in the situation. They are direct field prediction, 16x8 motion compensation and dual-prime prediction. We discuss their properties first before further considering the mode decision problem.

The scheme of the field prediction is similar to the frame prediction in MPEG-1. Depending on it belongs to a P or B frame, one or two motion vectors will be found for each 16x16 macroblock. If the motion vector is searched on a 32x32 region, and the field has a size mxn, then the total time consumed by the prediction can be estimated in the following. Note that the distortion of each macroblock is measured by mean square error.

$$\begin{aligned} T(\textit{Field Prediction,P}) &= \frac{mxn}{16x16}(32x32)\{(16x16x(add)) \\ &+ [(16x16x(multiply)) + (16x16x(add))]\}. \end{aligned} \quad (1)$$

If the field belongs to a B frame, it needs double computing time. Thus we have $T(\textit{Field Prediction,B}) = 2T(\textit{Field Prediction,P})$.

Another scheme is the 16x8 motion compensation which partitions each macroblock into upper and lower half, and finds their corresponding motion vectors separately. Supposing the search range of each half-macroblock is also reduced to 32x16, thus the total time consumed by the procedure can be computed in the following.

$$\begin{aligned} T(\textit{16x8 motion compensation,P}) &= 2\frac{mxn}{16x8}(32x16)\{(16x8x(add)) \\ &+ [(16x8x(multiply)) + (16x8x(add))]\}. \end{aligned} \quad (2)$$

Similar to previous results, if the field belongs to a B-frame, the consuming time will be doubled. Thus, we have $T(\textit{16x8 motion compensation,B}) = 2T(\textit{16x8 motion compensation,P})$.

The last scheme is named as dual prime. In this procedure, two predictions are computed preliminarily and then averaged to form the final prediction. Note it can not used while the previously coded picture was a B-picture or it is the first I-field picture of a frame. It is usually executed in P-pictures where there are no B-pictures between the predicted and reference fields or frames. The total time necessary can be estimated in the following.

$$\begin{aligned} T(\textit{Dual Prime,P}) &= 2x\frac{mxn}{16x16}(32x32)\{(16x16x(add)) \\ &+ [(16x16x(multiply)) + (16x16x(add))]\}. \end{aligned} \tag{3}$$

It is similar to the computing time of B-pictures because it also costs two motion vectors for each macroblock.

With the above discussion, we need total $T(\textit{Field Prediction,P}) + T(\textit{16x8 motion compensation,P}) + T(\textit{Dual Prime,P})$ to archive the best mode for inter prediction. The exhausted method costs lots of time and power thus a simpler and efficient prediction method is required.

The main idea of our fast mode decision is based on whether the object in the pictures transforms or not. If the objects in the pictures move but not transform, the dual prime mode or the traditional field prediction is chosen. If their shapes are transformed, the 16x8 motion compensation motion compensation mode is selected. Since the mode selection procedures are similar in P and B frames, we discuss P frame only and B frame can easily follow the same idea.

First we examine the objects transform or not by generating a roughly predicting frame with the motion vectors of the previous field. Let the pixel of the predicting field $pred_i$ which can be computed by $pred_i(x - mvx_{i-1}(x, y), y - mvy_{i-1}(x, y)) = ref_{i-1}(x, y)$. Some pixels in the $pred_i$ may be predicted repeatedly because some motion vectors refer to the same region. In this case, only the last predicting value is used. Some pixels which may not be assigned a value because some region is not referred in the previous field. The value of this kind pixel $pred_i(x, y)$ is set to be $ref_{i-1}(x, y)$. Let the prediction error caused by the roughly predicting field to be $\sigma_{rough,i}$. We set a threshold $\alpha\sigma_{i-1}$ where α is a variable designated by the designer and σ_{i-1} is the MSE of previous field. If $\sigma_{rough,i}$ is less than the threshold, the motion vectors to the previous two fields are first computed. Since the computations of dual prime mode and traditional field prediction rely on the same set of motion vectors, we can computer both results with little cost and choose the better one. Otherwise, we assume the object transforms or moves rapidly. In this situation, the 16x8 motion compensation mode is applied.

Comparing to the exhausted method which needs time $T(\textit{Field Prediction,P}) + T(\textit{16x8 motion compensation,P}) + T(\textit{Dual Prime,P})$, the fast mode decision needs only $max(T(\textit{Field Prediction,P}) + T(\textit{16x8 motion compensation,P}) + T(\textit{Dual Prime,P}))$ plus time to generate the predicting fields which is relatively small to the time of motion estimation.