Video–scaling algorithm based on human perception for spatio–temporal stimuli

Christoph Kuhmünch, Gerald Kühne, Claudia Schremmer and Thomas Haenselmann

Lehrstuhl Praktische Informatik IV, University of Mannheim, Germany

ABSTRACT

Video conferencing and high quality video–on–demand services are very desirable for many Internet users. However, Internet access channels ranging from wireless connections to high–speed ATM networks mean great heterogeneity with respect to bandwidth.

Hierarchical video encoders that scale and distribute video data over different layers enable users to adapt video quality to the capacity of their Internet connection. However, the construction of the layers at the encoder determines the video quality that can be expected at the receiver.

To achieve an optimal configuration of the different layers with respect to visual quality, we propose a hybrid scaling algorithm that scales video data both in spatial and temporal dimension. Using a quality metric based on properties of the human visual system our algorithm calculates an optimal ratio between spatial and temporal information. Additionally, we present experimental results that demonstrate the capabilities of our approach.

Keywords: Hierarchical video encoding, Internet video streaming, video quality metric

1. INTRODUCTION

Due to the constantly improving bit rates of the Internet and the availability of efficient compression techniques, applications like real–time video conferencing and high quality video–on–demand services are within reach for the average user. However, unlike television broadcasting, video transmission via the Internet has the disadvantage that various access channels ranging from wireless connections to high–speed ATM networks mean great heterogeneity with respect to the available bandwidth.

This constraint underlines the development of hierarchical video encoders that scale the video in the dimensions of temporal and spatial resolution. Hierarchical video encoders distribute the video data over several layers. According to the bandwidth available, a user chooses the number of layers he/she wants to receive. Clearly, the visual quality of the video received depends on the construction of the different layers at the encoder. To maximize the visual quality, the scaling procedure should exploit spatial as well as temporal resolution.

While there already exist a number of techniques for scaling a video in either dimension, the authors are not aware of any approach to compute the optimal ratio between spatial and temporal scaling at a given bandwidth at a specific moment. Keeping in mind that the data sink for video transmission via the Internet is a human observer, the question is to find an algorithm choosing the appropriate trade–off between temporal and spatial resolution to maximize the perceived quality of the video for the observer.

The algorithm presented in this work realizes intelligent spatio–temporal video scaling for a given bandwidth. It is thus eligible to portray a pre–processing step for all encoding standards that provide structural elements for hybrid video scaling without specifying implementation details (e.g. MPEG–2 video[1]).

The paper is organized as follows: Section 2 summarizes related work. Section 3 briefly reviews the fundamentals of layered video coding. Detailed design requests of our algorithm, different possible approaches, and an implementation overview are presented in Section 4. The perceptive quality measure underlying our approach and held abstract until then is demonstrated in Section 5. Our experimental results are pointed out in Section 6 and the paper ends by denominating open issues in Section 7.

*Send correspondence to:
{kuhmuench|kuehne|schremmer|haenselmann}@informatik.uni-mannheim.de

2. RELATED WORK

Our work relies on research conducted in the areas of (1) human perception modeling and (2) layered encoding of video streams.

The modeling of human perception of visual stimuli is still an open field of research. One of the first books to present a broad overview of how human observers see is [Frisby].[2] More recent research is detailed in [Wandell][3] where the human visual system is explained with respect to the MPEG compression standard. Psycho–physiological research has been carried out in order to measure the sensitivity of the human visual system in the three dimensions: color resolution, spatial resolution, and temporal resolution. Recapitulative these research projects proved the following attributes of the human visual system: (1) Human visual perception is based less on absolute (luminance) values but more on contrast.[4] (2) Contrast sensitivity is much higher for luminance than for chrominance.[5] (3) Contrast sensitivity is highly correlated to the spatial frequency of the perceived stimulus and decreases if spatial frequency increases.[6] (4) An important aspect of temporal perception is the critical flicker frequency, i.e., the minimum number of frames per time unit that make a video appear "fluid". This critical flicker frequency is highly correlated to luminance and motion energy.[7]

Based on these results, a number of mathematical models have been designed that simulate the human visual system. One model of still images based on wavelet transforms is presented in [Bock].[8] Modeling the human visual system by imprecise data sets is presented in [Steudel].[9]

A first attempt to widen the models of human visual perception into the spatio–temporal dimension and thus to adapt them to digital videos is called ITS metric and has been elaborated in [Webster].[10] The quantitative measure proposed in this work relies upon two quantities. The first one measures spatial distortions by comparing edge enhanced copies of the original and their corresponding approximating frames. The latter measures the loss of temporal information by comparing the motion energy between the original and the approximating frame sequences. These two units of information are post–processed by three measures whose weighted linear combination conforms highly with the results of subjective testing, a scale ranging from the school mark 1 (very bad quality) to 5 (excellent quality).

In the area of layered video coding several techniques has been developed which scale and compress a frame sequence in either temporal or spatial dimension. In [Pennebaker, Amir, McCanne][11–13] a spatial approach is described which relies on layered quantization. Each 8×8 block of each image is transformed into the frequency domain. The bits of the DCT coefficients are distributed over several layers. This corresponds to applying different quantization factors ranging from coarse to fine to the coefficients. Another approach in the context of spatial scalability based on pyramid encoding[14] is used in the MPEG–2 video standard.[1] In [Merz][15] temporal scaling is achieved by spreading consecutive frames on a video sequence over a number of layers.

3. HIERARCHICAL VIDEO CODING

Video can be interpreted as a vector consisting of the three dimensions: color resolution, spatial resolution, and temporal resolution. The *color dimension* is defined by the number of bits that represent the color value of each pixel. The *spatial dimension* describes the horizontal and the vertical resolution of each picture the video consists of. The *temporal dimension* describes the temporal resolution of the video, in other words, the number of frames per second.[16] More formally, we give the following

Definition. A color video V consists of a sequence of frames

$$V = \{(F_1, F_2, ...) \mid F_i \in [0, 255]^{w \times h \times 3}\},$$

where $w \times h$ denotes the spatial resolution and each pixel typically is represented by a triple $(Y, C_B, C_R) \in [0, 255]^3$ for the luminance, chrominance, and color hue of the pixel. ⌟

Hierarchical encoding techniques scale the video quality in at least one of the three dimensions. The idea is to encode video signals not only into one but into several output streams. Each stream S_i depends on all lower streams $S_0, ..., S_{i-1}$, can only be decoded together with these lower streams, and each stream adds to the quality of the video transmitted. In the following, we give a generalized definition of the one found in [McCanne].[13]

Definition. Let $V_{i,k}$ be a sub–sequence of V with the length k starting at frame i:

$$V_{i,k} = (F_i, \ldots, F_{i+k}). \tag{1}$$

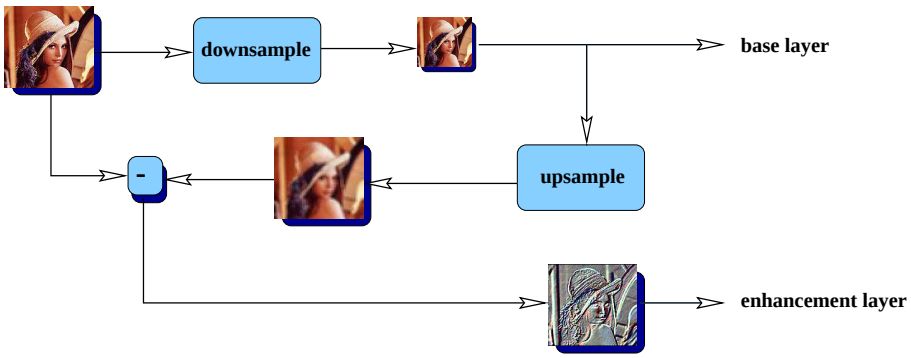


Figure 1. Data flow of a (spatial) Laplacian Encoder.[13]

A hierarchical encoder E encodes a sequence of k frames into L output codes $C^1, \ldots, C^L$. Therefore, E is a mapping

$$E : V_{i,k} \rightarrow \{C^1_{i,k}, \ldots, C^L_{i,k}\}. \tag{2}$$

In order to reassemble the video at the receiver side we need a decoder D that reverses codes $C^1_{i,k}, \ldots, C^l_{i,k}$ into a sequence of frames:

$$D : \{C^1_{i,k}, ..., C^l_{i,k}\} \rightarrow (\hat{F}_{i,1}, \ldots, \hat{F}_{i,k}) = \hat{V}_{i,k}, \qquad l \leq L. \tag{3}$$

The difference between the original sub–sequence $V_{i,k}$ and the reassembled sequence $\hat{V}_{i,k}$ shortens the more codes l are taken into account at this inversion. According to this definition, the elementary task of a hierarchical encoder E is to define encoding schemes that split (and compress) a given frame sequence into a set of codes $\{C^l\}$. A number of hierarchical video compression techniques have been developed that scale and compress a frame sequence in its three dimensions: time, size, and color depth. Color scaling is beyond the scope of this publication but in the following, we summarize the most common approaches to spatial and temporal scaling. A more detailed overview can be found in [Kuhmünch].[17]

Spatial Scaling. This scaling approach splits each video frame into its spatial frequencies. Since lower spatial frequencies are better perceived by human observers,[18] the lower layers of spatial scaling approaches concentrate on the lower frequencies, while higher layers provide information about higher spatial frequencies. Implementations either scale the coefficients of the discrete cosine transform (DCT) often used in video compression standards[12] or they produce a set of low–pass filtered copies of each video. The central idea of the latter approach, called pyramid encoding,[14] is described in Figure 1. The encoder first down samples the image, compresses it according to the chosen encoding technique, and then transmits it in the base layer stream. When the image is decompressed and up sampled, a much coarser copy of the original arises. To compensate for the difference, the decoder subtracts the resulting copy from the original image and sends the encoded differential picture in the enhancement layer stream.

Temporal Scaling. Temporal scaling approaches work very similarly to spatial approaches, but they distribute the *frames* of a video sequence over the different layers. Figure 2 visualizes a possible approach with three layers, where a subsample of the image sequence is transmitted on each layer.[15]

4. A PRE–PROCESSING ALGORITHM FOR LAYERED VIDEO ENCODERS

As described in Section 3, a layered video encoder compresses a video into multiple layers $S_0, S_1, \ldots, S_n$. In general, the encoder performs three steps:

(1) Each video frame is decomposed into atomic information units (U). For example, a single unit can consist of a number of DCT coefficients or a layer from the Laplacian pyramid of the frame. Consider a YC_bC_r–frame of size 352×288. Performing the DCT on 8×8 blocks of the frame results in $352 \times 288/(8 \times 8) = 1584$ DC coefficients and $352 \times 288/(8 \times 8) \times 63 = 99792$ AC coefficients for the luminance (Y) component. One way to form the unit is to combine all DC coefficients into one spatial atomic information unit, while fractions of the AC coefficients can be grouped into additional units.

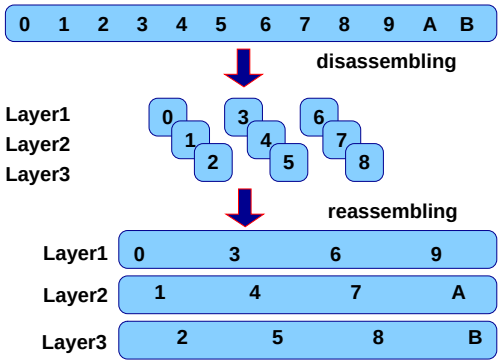


Figure 2. Temporal scaling of a video stream.

(2) The atomic information units of all frames are grouped into subsets of different visual importance. Figure 3 depicts this process. A single subset may contain units from either one frame or many frames. Consequently, information of a single frame may be encoded in either a single subset or in many subsets. If the scaling procedure is confined to the temporal dimension, information about a certain frame can only be found in one subset. If the scaling process is restricted to the spatial dimension, information about a certain frame can be found in all subsets. Accordingly, if spatio–temporal scaling is employed, information of a single frame can either be encoded in a single subset or in many subsets.

(3) Finally, each subset is encoded on a different layer considering the corresponding bandwidth limitation.

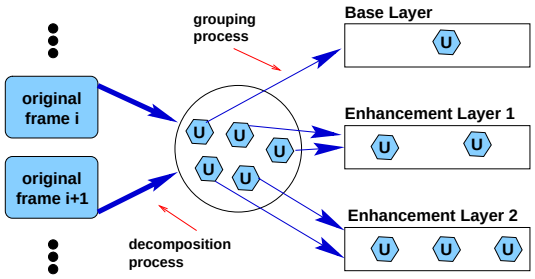


Figure 3. Each frame in the sequence is decomposed into atomic information units. The units are depicted as hexagons marked with the letter "U". It is the task of a layered encoder to code the units on the different layers. Pre–processing algorithms decide which information should be coded in which layer.

The subset that contains the most important information is coded into the base layer S_0. The other subsets are assigned to higher layers $S_1, \ldots, S_n$ according to their level of importance.

Obviously, the construction procedure of the different subsets determines the visual quality that can be achieved at the decoder. When using hybrid (i.e. spatio–temporal) layering algorithms, it is necessary to find an optimal ratio between spatial and temporal scaling. In the following we describe a pre–processing algorithm which solves this problem for a sequence of N frames.

The algorithm is designed upon the following assumptions: (1) There exists a metric that measures the *quality perceived by a human observer* of a decoded video by calculating the perceived distance between the original video and the decoded one (see Section 5). The metric can be given as a function of the used information of the frame sequence. (2) This function develops monotonously, i.e., each additional atomic information unit increases the quality of the video (this assumption must be fulfilled by the function anyway in order to fulfill the conditions of a mathematical metric). (3) There exists an algorithm that splits each video into atomic information units, e.g., an approach comparable to JPEG that produces runs of DCT coefficients.

The algorithm determines how much information from each frame of the sequence should be transmitted on each layer. The amount of information from each frame can be anything between no information at all (i.e., the frame is skipped) and the complete information from the frame.

Since the algorithm calculates each subset in the same manner, we confine the description of our algorithm to the base layer subset. Starting from an initial subset configuration, the algorithm calculates an optimal subset of information units that fit into the given bandwidth limitation and produces an optimal visual quality, i.e. the distance between the original and the decoded sequence is minimal.

4.1. Initial subset configuration

As mentioned above, the quality metric behaves monotonously, therefore we assume that an optimal combination exhausts the complete bandwidth. In our implementation the start point is the maximal temporal information that can be transmitted within the given bandwidth, i.e., we localize our search to a pure spatial scaling approach. As an alternative one could start with a certain frame in maximal spatial resolution.

The initial subset configuration is performed as follows (see Algorithm 1): The frames are sorted in reverse order according to their motion energy (see Section 5). The number of atomic information units for each frame is set to zero. Then, the number of atomic information units is incremented for each frame with respect to the sort order until the bandwidth limitation is reached.

Consequently, if the bandwidth limitation of the subset is too narrow to contain even a single information unit for each frame, frames with low motion energy are skipped.

4.2. Iterative refinement

From the initial subset configuration, our algorithm searches for the optimal combination by trading temporal information for spatial information (see Algorithm 2).

In each iteration the frame with the lowest motion energy is removed completely from the subset. The bandwidth gained is distributed over the remaining frames, i.e. the spatial quality of the video is increased, while the temporal information is reduced. In order to find out which distribution results in the best visual quality, information units are added to the remaining frames until the bandwidth gained has been consumed. In each step the quality achieved is calculated and the best subset configuration is stored.

After each iteration the quality of the decoded video is measured. The optimal combination has been found when the quality ceases to increase (see Algorithm 3).

4.3. Bandwidth calculation

Within each algorithm quality improvement and bandwidth calculation determine whether additional information units are added to the subset configuration. While the measurement of quality is described in Section 5, the question arises how to perform bandwidth calculation.

In each step of the algorithm the subset configuration is known. Consider, for instance, a frame sequence $F = \{f_0, f_1, \ldots, f_7\}$ with eight frames and the subset configuration $S = \{0, 4, 0, 0, 4, 0, 0, 0\}$. Each element in S contains the number of atomic information units for the corresponding frame: In our example only the frames f_1 and f_4 ($S_1 > 0, S_4 > 0$) are currently important. Proceeding from this subset configuration, the necessary bandwidth has to be calculated. However, the complexity of this calculation depends on the compression method chosen.

If atomic information units consist of DCT coefficients and a JPEG compression method is used, the calculation is straightforward: One has to compute the runs of the coefficients and estimate the Huffman compression. However, if a compression method is used that employs motion compensation, additional steps are necessary before bandwidth can be calculated. Consequently, the computational overhead for bandwidth calculation can be reduced by choosing atomic information units appropriate to the compression method.

```
funct init_subset (FrameSequence F, integer max_b) ≡
                    /* F = frame sequence to optimize, max_b = available bandwidth of this layer */
  begin
    subset := {};
    while there is still bandwidth left do
          foreach Frame i ∈ F sorted by descending motion energy do
            if bandwidth exhausted then break; fi
            increase_quality(subset, i);
          od
    od
    return (subset);
  end.
```

Algorithm 1: Initial subset configuration.

```
funct spread_spatial_info (Configuration current_subset, FrameSequence F, Frame f) ≡
                                        /* current_subset = current spatio–temporal combination */
          /* F = frame sequence to optimize, f = frame which spatial information is distributed */
  begin
    max_quality := calc_quality(current_subset);          /* calculate current quality */
    best_subset := current_subset;
    remove_from_subset(current_subset, f);                /* reduce temporal resolution */
    while there is still bandwidth left do                /* spread bandwidth */
          foreach Frame i ∈ F do                          /* identify the best frame */
            increase_quality(current_subset, i);          /* add one spatial information unit */
            q := calc_quality(current_subset);
            if q > max_quality                            /* check for better quality */
              then best_subset := current_subset;         /* store the best configuration */
                   max_quality := q;
            fi
            decrease_quality(current_subset, i);          /* reset frame information */
          od
          current_subset := best_subset;
    od
    return (max_quality, best_subset);
  end.
```

Algorithm 2: Spreading of spatial information.

```
funct spatio_temporal_trading (FrameSequence F, integer max_b) ≡
                    /* F = frame sequence to optimize, max_b = available bandwidth of this layer */
  begin
    subset := init_subset(F, max_b);                      /* calculate initial subset configuration */
    do
       f_min_motion := search_frame_with_min_motion_energy(F);
       (q, subset) := spread_spatial_info(subset, F, f_min_motion);
    while q increases ∧ bandwidth available;
    return (subset);
  end.
```

Algorithm 3: Spatio–temporal trading.

5. VIDEO QUALITY METRICS

The algorithm described in the previous section relies on a metric that measures the quality of a video sequence. The performance of the algorithm depends heavily on the quality of the metric, i.e., we need a metric that measures the video distortion perceived by the human observer as precisely as possible. We define the quality of a (decoded) video sequence relative to the original sequence. In order to get a fair measurement of the difference between the original image sequence and the decoded sequence we assume that both contain the same number of frames and that the frames are all of the same spatial size. A quality measure operates on a sequence of n original images $O_1, ..., O_n$ and compares it to the decoded sequence of n images $A_1, ..., A_n$ that approximates the original sequence by decoding a video stream.

One measure widely used in the context of video quality is the signal-to-noise ratio (SNR). It describes the energy of an undistorted signal in relation to the noise introduced by processing the signal (e. g. compression and decompression of the signal). However, high SNR values do not always correspond to signals with perceptually high quality[19] (see Figure 4). Therefore, it is necessary to create measures that are more appropriate with respect to the human visual system.

In a first attempt, we used the ITS metric[10] in our algorithm, but it turned out that the metric varies to inertly for our application: unfortunately, the output indeed varied only in the range 4.55 to 4.77, no matter of what the distorted video was like.

Therefore, we have defined our own metric based on the concept of the ITS metric, but which is better adapted to our video scaling application. The metric is described in the following subsection.

5.1. Adaption of the ITS metric

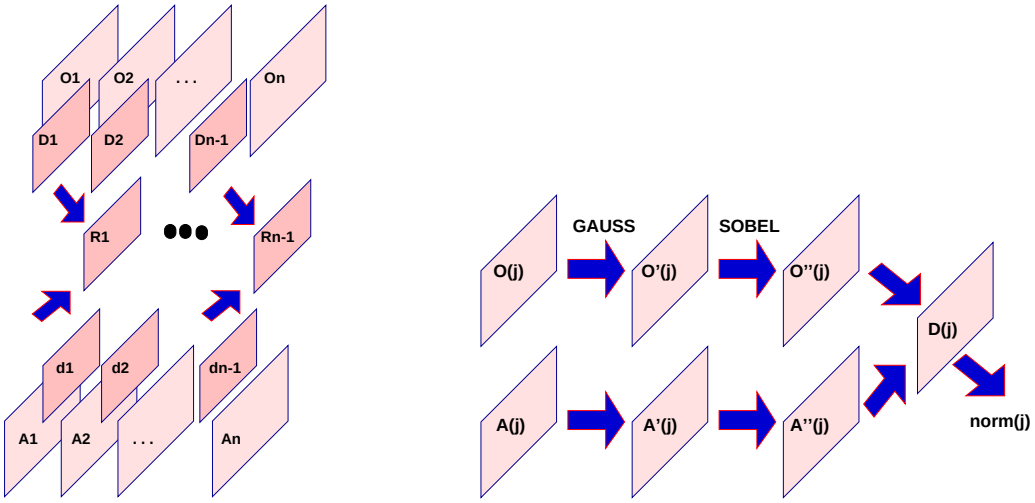
Since the metric is going to be used to measure the quality of video scaled in its temporal and spatial dimensions, the metric concentrates on these two dimensions. Hence, the quality measure is divided into two components, namely one that reflects the sole picture quality and a second one that is focused on the dynamics within the sequence. It is an important feature of the metric that the two components strictly separate image quality from sequence dynamics since this allows us to focus on the distortions and artifacts produced by spatial and temporal scaling.

5.1.1. Spatial quality measure

Because edges provide important information to the human visual system[3] our spatial quality measure operates—similar to the ITS metric—on edge–enhanced image sequences. Each frame of the original and the approximated video sequence is processed in turn (see Figure 5).

Each step applies a Gauss filter on the original picture O_j and the approximating picture A_j in order to remove high–frequency noise from the images. Additionally, the low–pass filtering with the Gauss operator removes "thin" edges from the video, which are less important to the visual system. Afterwards the frames are filtered with the Sobel operator in order to carry out edge detection. The Euclidean norm of those two images O'' and A'' is then calculated on a pixel basis (Equation 4) . Note that $O_j(x, y)$ addresses the gray–values of picture O_j in column x and row y. The same holds for $A_j(x, y)$.

Figure 4. SNR vs. perceptual quality. From left to right: (a) original image, (b) image distorted by JPEG block artefacts (SNR $\approx$ 20 dB), (c) image distorted by random noise (SNR $\approx$ 20 dB).



(a) Spatial quality measure.

(b) Temporal quality measure.

Figure 5. In the spatial quality measure (a), each original and each approximating frame is Gauss and Sobel filtered. The difference frame D leads to the calculation of the Euclidean norm. In the temporal quality measure (b), first, difference pictures are calculated for the original and the approximating sequence. Then the difference pictures are compared.

$$NORM_j = \frac{1}{height \cdot width} \sum_{y,x} (O''_j(x,y) - A''_j(x,y))^2 \tag{4}$$

The norm expresses how well the edges in the approximated video frames correspond to the edges in the original sequence. Note, that in contrast to the SNR our spatial quality measure clearly distinguishes the distorted images shown in Figure 4.

Remark. Since it is our goal to separate image quality from sequence dynamics, we only take pictures into account where spatial information is available. More formally, if no spatial atomic information units from the frame O_j is received then no spatial distortion is calculated. One might ask how frame A_j can be calculated anyway if no spatial information is available. In that case, the decoder can conceal the missing frame by using concealment strategies. A simple approach is to redisplay the previous frame. In our case, we assume that the decoder displays the (temporally) nearest neighboring frame.

5.1.2. Temporal quality measure

In order to measure temporal distortions we find out how much the sequence dynamics differ between the original and the approximating sequence. For the dynamic measure the difference between two succeeding images is produced: one for the sequence of original images (D_j), and another for the approximations (d_j). This process is depicted in Figure 5.

In the calculation of the spatial quality measure for the approximating sequence, the images $O(j)$ were of lower quality. However, in the calculation of the dynamic quality we use original images for both the approximating and the original sequence. So the only difference between the two arrays of images within the dynamic calculation is that the same image may be replicated several times in the approximating sequence, especially if neighboring images differ only by a small amount.

The dynamic component calculates difference or delta images D_j for both the original and the approximating image sequence. This is done by subtracting gray–values of neighboring images and storing the unsigned value as a gray–value of the difference picture D_j (Figure 5). As a result, a difference image between each two pictures of both the

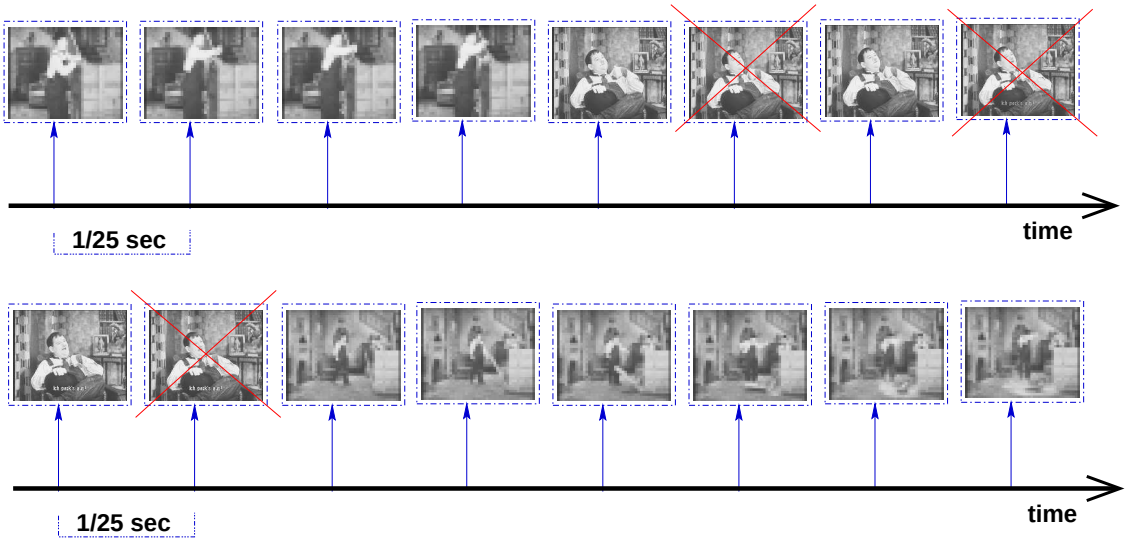


Figure 6. Perceptual aspect of the sequence "Laurel and Hardy". An initial scene with high motion shows Laurel opening a box. His arms and upper body move strongly, so our algorithms has decided on selecting all frames at the expense of spatial resolution. Another scene of the video shows Hardy sitting and talking. As motion is low, spatial resolution is high, and frame dropping saves bandwidth. In a third scene, Hardy walks across the room and water pours out of the box. Again, our algorithm favored spatial scaling.

original sequence D_j and the approximating sequence d_j is produced. The sum of the Euclidean distances $\mid D_j - d_j \mid$ for all j forms the measure of the difference in dynamics.

In fact, different motion energy can only occur if pictures in the approximated sequence are dropped that contained substantial changes with regard to that sequence's neighbors. In contrast to a sequence of pictures changing only slightly, dropping frames will not cause significant changes.

5.1.3. Combination of the two components

In order to receive a single value measure, the spatial and the temporal quality measure have to be combined in some way. A first intuitive approach would be to combine the two values linearly,

$$Q_{\texttt{total}} = \alpha Q_{\texttt{temporal}} + (1-\alpha) Q_{\texttt{spatial}}, \quad 0 \leq \alpha \leq 1.$$

However there are some aspects that give reason to combine the two measure multiplicatively, i.e.

$$Q_{\texttt{total}} = Q_{\texttt{temporal}} \cdot Q_{\texttt{spatial}}.$$

The multiplicative combination is motivated by the observation that both dimensions have a critical minimum quality level. For example, a certain minimum frame rate is essential to realize the sequence as a video. If the frame rate drops beneath a vital minimum frequency, even a very high spatial quality cannot compensate for the suffered distortion. The same holds true for the spatial quality. Besides this fact, the multiplicative combination disburdens us from the task to normalize both measures into the same domain.

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed pre–processing algorithm, we processed several video sequences. In the following the experimental setup is outlined and results for two video sequences are presented.

We based the algorithm on the metric described in Section 5. The implementation of the algorithm processes eight successive frames at a time to define the best trade–off between spatial and temporal resolution for those eight frames.

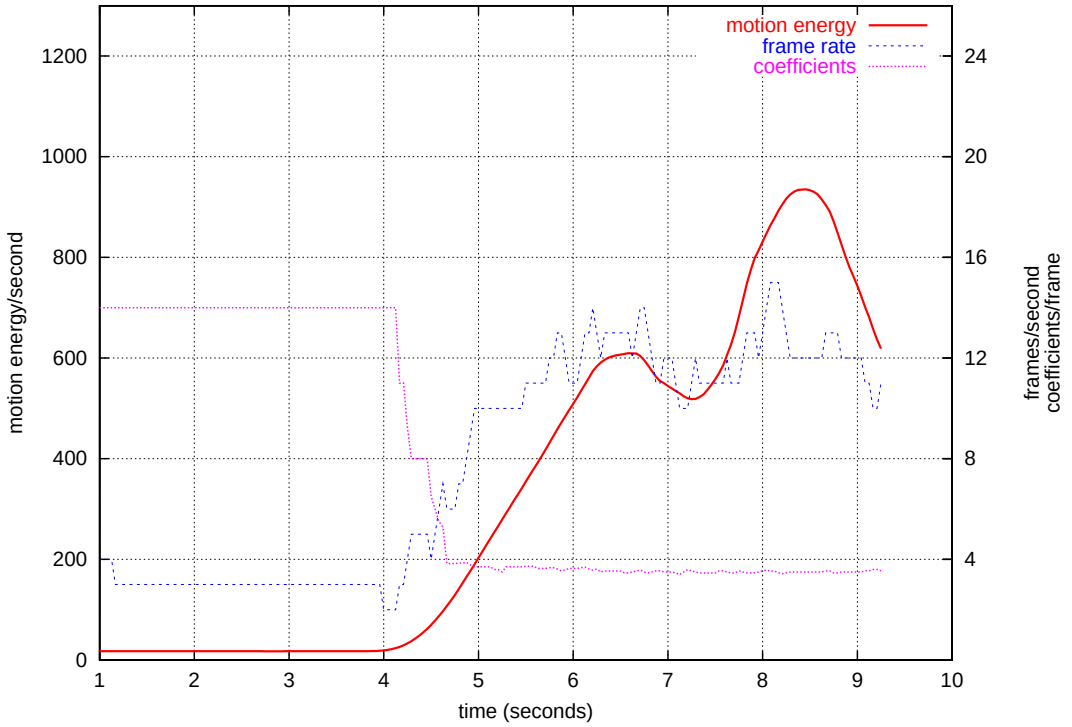


Figure 7. Results of test sequence "traffic". The three graphs show the moving averages of each 24 frames (one second). Note that we chose a minimum number of three coefficients per frame since otherwise the quality is not acceptable. The maximum spatial quality is reached with 30 coefficients.

For spatial scaling we used a layered DCT approach described in Section 3. Thus an atomic information unit in our test implementation consists of a single DCT coefficient for each block.

Our algorithm decides how many coefficients C_0 are coded for each block within a frame. The first naive practice would be to allocate the same number of coefficients for every single block of a frame. As most frames contain 8×8 blocks with different details, we have chosen a more advanced implementation: Blocks with more details are coded with more coefficients than those with less detailed information. Consequently, the blocks of a frame are coded with C_0 coefficients in average.

The algorithm has then been tested on several different video sequences. In this paper we present the results for two different videos. The "Laurel and Hardy" sequence is gray–scaled and contains both slow and high motion scenes. The "traffic" sequence is a color sequence and shows a crossing with traffic lights. In the first four seconds of the sequence the traffic lights are red and therefore very little motion occurs. In the following the traffic lights turn to green and the cars slowly start moving. The amount of motion increases as the cars are accelerating.

In the two test sequences, our algorithm favored spatial scaling in scenes with high motion and decided on frame dropping in scenes with little motion. The perceptual aspect of such a hybridly scaled sequence is demonstrated in Figure 6. Scenes with high motion show a great number of frames per second at the expense of spatial resolution. The individual frames in such sequences are thus blocky (DCT encoding) or blurry (pyramid and wavelet encoding), but render the motion at their best. In scenes with little motion, the individual frames mirror good spatial resolution and the bandwidth is saved by dropping frames.

In Figure 7 the motion energy, the number of frames per second and the number of coefficients per frame for the "traffic" sequence are displayed. Again, the algorithm places many frames in sections with high motion and drops frames in parts with little motion. Note that the number of frames per second increases in proportion to the motion energy. The number of coefficients used per frame decreases in turn.

Our algorithm uses three frames per second with 14 coefficients per frame in the section with little motion (seconds 1–4), six frames with six coefficients while the cars accelerate (seconds 4–5), and twelve frames with about four coefficients when the cars reach their maximum speed (seconds 8–9).

A closer look at the graph of Figure 7 provides the information that the frame rate rapidly changes within the test sequence "traffic". When reviewing the preprocessed "traffic" video we found these rapid changes of the frame rate quite annoying. We assume that the human observer gets used to a certain degree of flickering after a while. When there is a strong jitter in the degree of flicker this adaptation will not happen. The results indicate that our metric currently does not cover this phenomenon. In order to overcome this effect we applied a low pass filter to the frame rate and the spatial resolution. This post processing improved the perceptual quality significantly*.

7. CONCLUSIONS AND OUTLOOK

In this paper we have described an algorithms for intelligent spatio–temporal video–scaling based on human perception. When the available bandwidth does not allow full resolution in space and time, a perceptual quality metric decides whether scaling the spatial or the temporal dimension is less annoying for the visual aspect of the video at the given moment.

The experimental results presented in this work show our approach to video–scaling based on human perception to be promising. There are various aspects that require further research though.

Future work will focus on the metric simulating the human perception of video. Questions in this area might be: What weighting factors of both spatial and temporal sensitivity best model the human perception? How might multi-resolution be introduced into the model? How can color perception be introduced into the measure?

The quality testing, i.e., the judgment whether a temporally–spatially scaled video V_1 at a given bandwidth is superior to a differently scaled video V_2 at the same bandwidth — which entails the definition of the metric — has been performed within our institute to date with eight test persons. We will set up a test environment according to the ITU recommendations.[20]

An interesting question is how oscillating quality influence the quality judgment of human observers? Intuitively, spatial resolution varying from one frame to the other might be evaluated negatively. But are there thresholds in spatial resolution difference and in time resolution difference below which the perceived quality increases?

Finally, our algorithm will have to be integrated into an MPEG–2 encoder to obtain comparable objective answers.

REFERENCES

1. ISO/IEC 13818-2, "Information technology – generic coding of moving pictures and associated audio – part 2: Video," 1995.
2. J. P. Frisby, *Seeing — Illusion, Brain and Mind*, ISBN 0-19-217672-2, Oxford University Press, Walton Street, Oxford, 1979.
3. B. A. Wandell, *Foundations of Vision*, Sinauer Associates Inc, Sunderland, Massachusetts, USA, 1995.
4. C. J. van den Branden Lambrecht, *Perceptual Models and Architectures for Video Coing Applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1996.
5. K. Mullen, "The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings," *J. Physiol.* , 1985.
6. F. V. Ness and M. Bouman, "Spatial modulation transfer in the human eye," *J. Opt. Soc. Am.* **57**(3), pp. 401–406, 1967.
7. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Chapman & Hall, New York, 1997.
8. F. Bock, *Analyse und Qualitätsbeurteilung digitaler Bilder unter Verwendung von Wavelet–Methoden*, ISBN 3-8265-4278-9, Shaker Verlag, Aachen, 1998.
9. A. Steudel, *Das unscharfe Paradigma in der modernen Bildcodierung*. PhD thesis, Technische Universität Darmstadt, ISBN 3-89722-102-0, 1998.

*The video sequences can be found under http://www.informatik.uni-mannheim.de/ cjk/mmcn01/. We highly recommend to download and watch them.

10. A. A. Webster, C. T. Jones, M. H. Pinson, S. D. Vorna, and S. Wolf, "An objective video quality assessment system based on human perception," in *SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 15–26, Society of Photo–Optical Instrumentation Engineers (SPIE), (San Jose, CA), February 1993.
11. W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Compression Standard*, Van Nostrand Reinhold, New York, 1993.
12. E. Amir, S. McCanne, and M. Vetterli, "A layered dct coder for internet video," in *Proc. of IEEE International Conference on Image Processing ICIP '96, Lausanne Switzerland*, pp. 13 – 16, IEEE, September 1996.
13. S. McCanne, *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, CA, USA, 1996.
14. P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications* , 1983.
15. M. Merz, K. Froitzheim, P. Schulthess, and H. Wolf, "Iterative transmission of media streams," in *Proceedings of the Conference on Multimedia '97*, pp. 283–290, ACM, 1997.
16. C. A. Poynton, *A Technical Introduction to Digital Video*, John Wiley & Sons, 1996.
17. C. Kuhmünch and G. Kühne, "Efficient video transport over lossy networks," Tech. Rep. 7-98, University of Mannheim, http://www.informatik.uni-mannheim.de/~cjk/publications/, April 1998.
18. A. Robertson and J. Fisher, *Color Vision, Representation and Reproduction*, ch. 2. In Benson [Ben85]. ISBN 0-07-004779-0, 1985.
19. V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards*, Kluwer Academic Publishers, Norwell, MA, 1997.
20. I. R. I.-R. BT.500-9, *Methodology for the subjective Assessment of the quality of television Pictures*. International Telecommunication Union, 1998.