QUALITY ISSUES FOR RESOURCE CONSTRAINED SCALABLE VIDEO ALGORITHMS

Christian Hentschel

Media Technology, Brandenburg University of Technology Cottbus
Konrad-Wachsmann-Allee 1, 03046 Cottbus, Germany
phone: +49 355 692128, fax: +49 355 692150, email: christian.hentschel@tu-cottbus.de
web: www.tu-cottbus.de/mt/

ABSTRACT

Video signal processing is shifting from dedicated hardware to software implementation due to its flexibility. Digital signal processors (DSPs) for media processing are limited in its resources to enable cost efficient implementations for consumer devices. One way to achieve cost-efficient implementations is to use resource-quality scalable video algorithms (SVAs). This implies that dynamic resource adaptations result in dynamic quality changes which might affect the overall image quality. Starting from properties of SVAs, typical issues on quality including proposals for high-quality image processing will be presented.

1. INTRODUCTION

Algorithms for media processing are usually designed for a specific quality, and for many years implemented on dedicated hardware for their specific environment. For instance, in traditional television receivers various, specific ICs are combined to perform e.g. color decoding for NTSC or PAL systems, noise reduction, or frame rate up-conversion.

It is the trend of technology to develop more and more programmable platforms that allow media applications in software. Expected advantages are reduced time-to-market, reuse of hardware and software modules, portability, and flexibility. These are the issues that gain interest with respect to dedicated hardware.

Restrictions are the always limited computation capabilities. Resource limitations become especially a problem for the lower-cost, mass market processors with lower performance. On the software module side, current algorithms are designed for highest quality at given resources. They are not scalable and have a fixed functionality. It is simple to predict that the number of algorithms running in parallel is platform dependent and very limited.

Some internet applications use a kind of scalable algorithms to control the data rate. This is done by subsampling the video data, by either deleting entire frames, lines, or pixels. Deleting information with change of resolution is not acceptable in many application areas such as high quality video processing in television systems.

An alternative is to use resource scalable video algorithms (SVAs) which could solve a number of problems with respect to real-time processing on programmable platforms [1]. SVAs are able to trade output quality with resource usage. For instance, a down-scaler for picture-in-picture applications may use simple pixel and line subsampling at low computational resources (very low quality) while using suitable pre-filters in case of sufficient resources (good to high quality). New quality issues occur since these scalable algorithms combined with QoS resource management may result in fluctuating quality with a low acceptance rate by consumers. Quality issues combined with resource usage are topics of the following sections.

2. PROPERTIES OF SCALABLE VIDEO ALGORITHMS

A scalable algorithm is an algorithm that:

- allows the adaptation of quality versus performance on a given architecture,
- supports different software and/or hardware platforms for media signal processing, and
- is easily controllable by a control device for several predefined settings.

A set of scalable algorithms in a modular form can perform the different applications needed in a set-top box, TV set, multimedia PC, or, more in general, media processing unit.

2.1 Basic scalable video algorithms

Fig. 1 shows the basic structure of a scalable algorithm. Having a common interface via a quality control block is essential to communicate the possible quality levels and resources required to the QoS environment. An algorithm can be split into a set of specific functions, some of which are scalable to provide different quality levels. The properties of the active algorithm depend on the appropriate combination of the quality levels of the functions. These combinations may vary, but only a few may provide acceptable quality levels for the SVA (Fig. 2). The quality control block contains this information and the appropriate settings for the functions. The optimal quality-resource combinations are connected by the curve with maximum quality at lowest resources.

Typical quality resource behavior of several investigated SVAs (sharpness enhancement, scaling, IDCT for MPEG decoding) is shown in Fig. 3. At minimal resources the quality can be very low, with a steep increase in quality for increasing resources. A realized example is a spatial downscaler with low resource usage by pixel subsampling with poor output quality (strong aliasing). An additional bilinear interpolation with few additional resources increases the quality significantly. Further quality improvements by using high-order polyphase filters become smaller compared to the resources required. As a result, a wide range of resource scalability is possible within a small range of quality. The area of high quality changes at small resource changes should be avoided for scalable media processing.

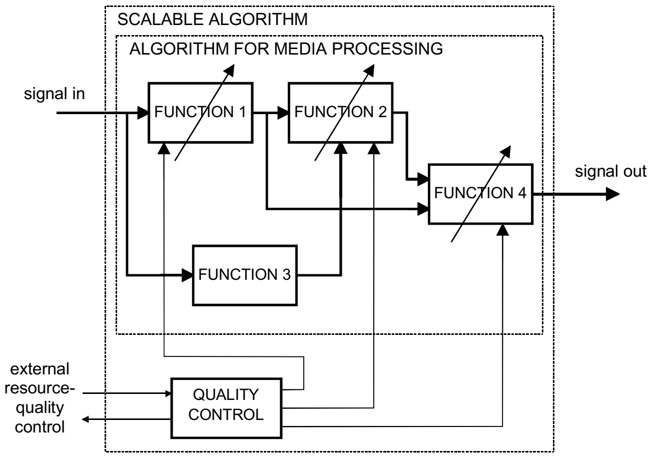


Fig. 1. Basic structure of a scalable algorithm.

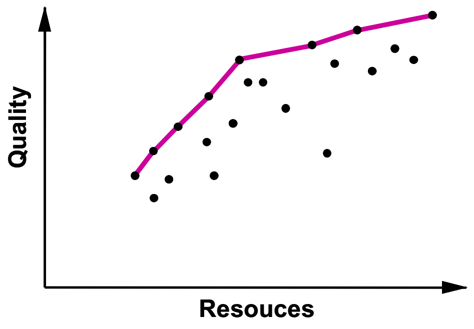


Fig. 2. Best choices of quality-resource combinations for functions of the entire scalable algorithm.

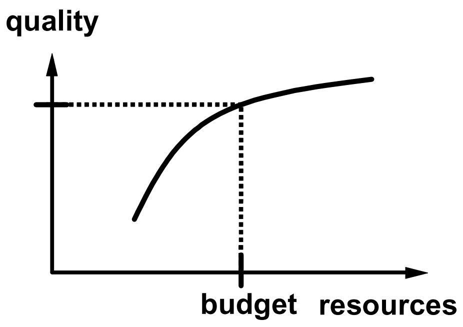


Fig. 3. Typical quality-resource behavior of scalable algorithms.

The resource-quality behavior in Fig. 3 is idealistic and cannot reflect more specific properties. Resources are multidimensional and can include CPU cycles, memory, bus bandwidth, coprocessors, etc. Quality can be measured in a multidimensional space as well. Properties such as sharpness, color reproduction, noise, quantization, algorithm specific compression artifacts can be expressed in quality, but also more specific properties such as spatial resolution in x- and y-direction, temporal resolution, judder, etc. A research topic is how these parameters influence each other, especially in a dynamic environment.

2.2 SVAs with data dependent resource requirements and quality

Programmable components are most suitable for irregular processing. Compression algorithms and non-linear processing algorithms like motion estimation have such irregular processing, while video processing algorithms such as scaling, image enhancement etc. are mostly executed by regular, pipelined pixel processing. Thus video processing on programmable components require different kinds of algorithms to ensure efficient and effective processing at available resources.

A different processing approach is illustrated in Fig. 4. The advantage of irregular processing is the option to do input data dependent image analysis (edges, textures, motion, etc.) and choose for a strategy how to process the data.

An example is priority processing of sharpness enhancement in video images. All relevant edges must be detected which could appear sharper by adding detail information. In flat, unstructured regions the addition of detail information would increase the noise level which lowers subjective image quality. With priority processing, first the most relevant edges should be enhanced, followed by lower priority regions. Flat regions should not be processed, or, in case of still available resources, could be processed by noise reduction algorithms.

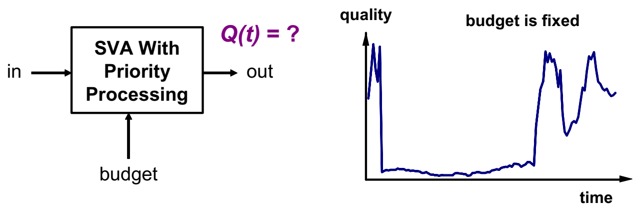


Fig. 4. Fixed processing budget and data dependent output quality.

An advantage of priority processing is to be able to interrupt image processing in case of low resources while still get the maximum on image quality. Depending on the image content (heavily structured or more flat areas) resources for a fixed output quality vary and vice versa (Fig. 4). Processing resources have no fixed relationship to output quality and cannot be used for quality estimation. Therefore, output quality measurement becomes an important subject for resource allocation and overall system and application optimization.

Fig. 5 shows an SVA with internal quality measurement to indicate its data and budget (resource) dependent performance. Typically, media processing functions can be used to estimate the output quality. In case of sharpness enhancement, the final priority level processed gives an indication of the achieved output quality. For motion estimation, the final, average accuracy and reliability can indicate the output quality.

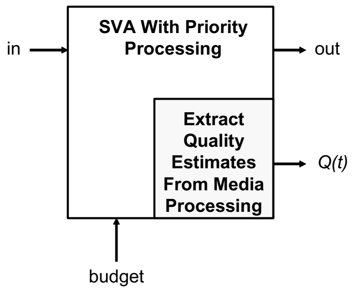

SVA With Priority Processing
in
out
Extract Quality Estimates From Media Processing
Q(t)
budget

Fig. 5. Quality measure within a scalable video algorithm.

3. QUALITY ISSUES IN AN SVA CHAIN

A modular system approach with simple QoS resource management is essential for flexible and manageable multimedia consumer terminals. Consumer terminals include mobile and stationary devices with stand-alone capabilities or within a network environment. Instead of measuring the quality at several positions of a processing chain, SVAs with integrated quality measurement could significantly simplify the system design.

In addition, properties of the input signal can help to optimize the functional structure as well as the quality of the overall system. Input signals come from different sources such as analog NTSC or PAL sources, Y/C, digital sources with different compression algorithms (MPEG, DV, H.264…), compression quality, etc. These input signals may have specific artifacts such as noise (SNR), blocking, ringing, etc. and could vary in resolution (HD, SD, CIF,…) and sampling formats (4:4:4, 4:2:2, 4:2:0, 4:1:1…). Since these input signal properties are important for the entire chain processing, these parameters should be collected in or close to the decoder as depicted in Fig. 6. Analysis information can then be used for optimizing quality in the resource-scalable processing chain.

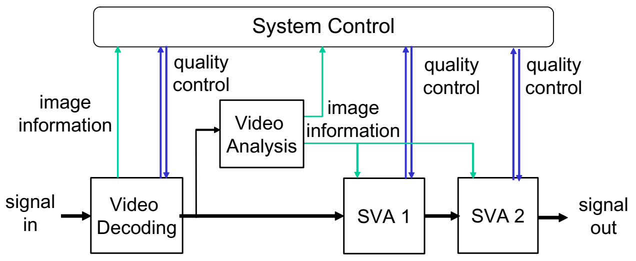


Fig. 6. Input signal quality analysis for quality control of the processing chain.

Another option for quality/resource usage optimization is the distribution of resources to algorithms depending on their priority. An example is an algorithm for noise reduction which heavily depends on accurate noise measurement. Noise measurement is typically done within the picture and consumes a part of the resources. In case of lack of resources to process the entire image, it would be a poor choice to skip the entire image or a part of it. The measured noise level of a particular video source including the particular transmission channel does not vary a lot from frame to frame, and therefore these resources could be shifted to the video processing algorithms as depicted in Fig. 7. Instead of measuring the noise level in every image, the noise level could be depicted from images with sufficient resources and repeated in images with lack of resources.

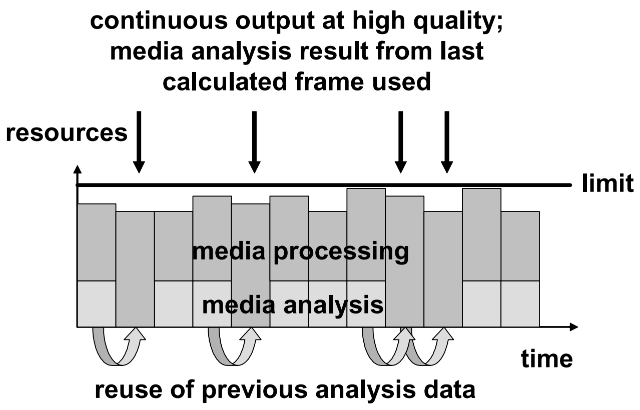

continuous output at high quality; media analysis result from last calculated frame used
resources
limit
media processing
media analysis
time
reuse of previous analysis data

Fig. 7. Freeing resources for high-quality processing by controlling media analysis algorithms.

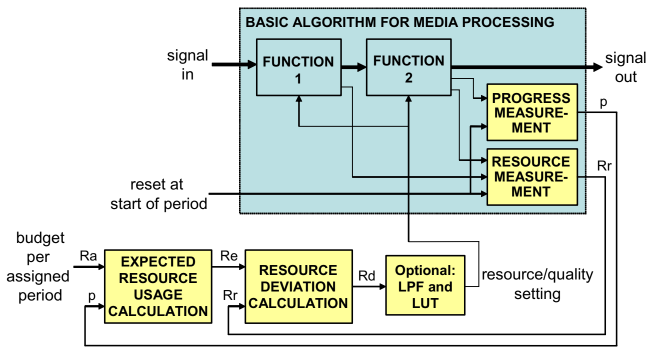
4. CONTENT DEPENDENT RESOURCE FLUCTUATIONS AND LOAD BALANCING

The general schematic diagram of a progress-based resource regulator is shown in Fig. 8 [2]. The basic algorithm for media processing contains two new functions for the measurement of progress P and used resources Rr. These measurements are derived from internal media specific processing data and are therefore independent from external system data. For example, in the 3D-RS motion estimator [3], P is the number of block lines that have been processed and Rr is the number of vector candidates that have been used (which is a good measure for the actual number of CPU cycles used [4]).

The external input indicates the available budget per assigned period, typically a frame. For the motion estimator example, this is the total number of vector candidates that is, on average, needed for processing all block lines within a frame.

Together with the measured progress P, the expected resource usage can be calculated at incremental periods of a frame. The expected resource usage can then be compared with the measured resource usage Rr to calculate the deviation from target Rd. The calculated target may be smoothed by a low pass filter and additionally corrected by a nonlinear function before adjusting the resource/quality setting of the scalable media processing algorithm.

This regulation scheme ensures that the regulation properties are independent of the amount of data already processed (actual progress, e.g. screen position). The regulation works on differences between expected and real resource usage during the assigned period and regulates close to a specific resource budget per frame, independent of the input data properties. Despite the very good regulation properties, also a quality gain has been recognized by observers.



LPF: Low-Pass Filter; LUT: Look-Up Table (non-linear function)

Fig. 8. Fast load regulation within an SVA for optimized quality on given resources.

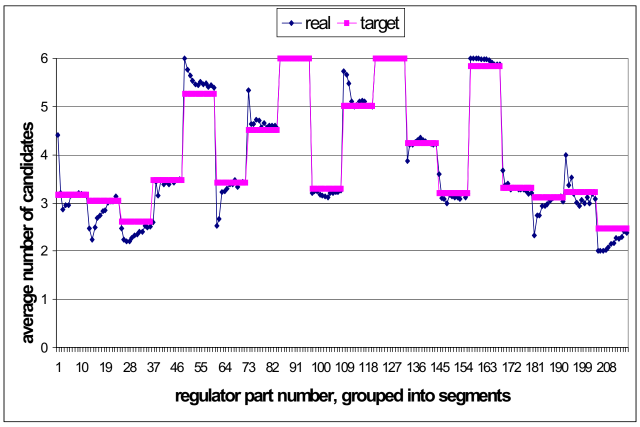


Fig. 9. Motion estimator regulation results within segments of a frame. Shown are the target (squares/bars) and real (diamonds) number of vector candidates.

Media processing may require different resources for different parts of the image such as stationary image parts, moving areas, textured areas, flat areas, motion vector fields with similar direction and velocity, etc., to achieve approximately constant perceptual quality. The regulator investigated so far does not take this into account.

The schematic diagram in Fig. 8 is, however, suitable to regulate resource usage to individually pre-determined budgets in image areas or segments while still providing regulation to an overall budget for the entire frame. The image may be divided into irregular segments in shape and size based on the content. Alternatively, the image may be divided into segments in a regular grid.

In the example 3D-RS motion estimator we have divided the image into 18 rectangular segments on a regular grid of 3 (horizontally) by 6 (vertically). Each segment gets an assigned budget (number of vector candidates) and a (normalized) segment progress measure (number of processed block lines) and regulation is done within each segment individually. Fig. 9 shows the impressive behavior of the regulation to the average number of vector candidates. At the start, even a severe mismatch between the used resources and the target amount of resources will be compensated for when finishing the segment processing.

5. CONCLUSIONS

In the past, image quality has often been optimized on single processing algorithms which later were combined for the entire application. The result of the final quality often depended on the experience and knowledge of the engineers. The signal source, encoding, and transmission already affects image quality, which should be preserved as much as possible. In the receiver, incoming image quality as well as processing steps including the order of certain algorithms have to be taken into account for optimal image quality. A new challenge are scalable applications which continuously adapt image quality to the available resources. Since image quality in scalable applications will dynamically fluctuate, new, real-time methods were investigated to achieve good output quality. Several ideas to optimize image quality in dynamic systems have been discussed in the paper. Investigations showed the effectiveness of these methods.

ACKNOWLEDGMENT

The author thanks all Philips members supporting the work on scalable video processing for their valuable contributions with special thanks to Maria Gabrani, Rob Wubben, and Ralph Braspenning.

REFERENCES

[1] C. Hentschel, R.J. Bril, M. Gabrani, L. Steffens, K. van Zon, S. van Loo, “Scalable Video Algorithms and Dynamic Resource Management for Consumer Terminals,” Int. Conf. on Media Futures (ICMF 2001), Proceedings, Florence (Italy), May 2001, pp. 193-196.

[2] C. Hentschel, R. Wubben, “Novel Resource Regulator for Media-Processing Algorithms on Programmable Components,” Digest of the ICCE’2005, Las Vegas (USA), Jan. 2005, pp. 443-444.

[3] G. de Haan, J. Kettenis, B. Deloore, “IC for Motion Compensated 100 Hz TV, with a Smooth Motion Movie-Mode,” Digest of the ICCE’95, Chicago (USA), June 1995, pp. 40-41.

[4] R. Braspenning, G. de Haan, C. Hentschel, “Complexity Scalable Motion Estimation,” International Conference on Visual Communications and Image Processing (VCIP), Proceedings, San Jose (USA), January 2002, pp. 442-453.