

Practical Content-Adaptive Subsampling for Image and Video Compression

Alexander Wong
Department of Electrical and Computer Eng.
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
a28wong@gmail.uwaterloo.ca

William Bishop
Department of Electrical and Computer Eng.
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
wdbishop@uwaterloo.ca

Abstract

Subsampling is a commonly used technique for modern image and video compression. Existing image and video standards such as JPEG, MPEG, and H.264 provide support for uniform chroma subsampling. This paper presents an algorithm that uses adaptive luma subsampling based on a combination of three perceptually significant image characteristics (texture, edges, and brightness) to complement uniform chroma subsampling. The algorithm is computationally efficient, simple to implement, and easy to integrate into existing standards. Experimental results show that the introduction of adaptive luma subsampling improves image quality both quantitatively and qualitatively when compared with the sole use of uniform chroma subsampling.

1. Introduction

Image and video compression have become integral in consumer electronics given the increasing need for the efficient storage of visual data. The most widely used compression schemes in consumer devices are based on block-transform coding. These include still image compression schemes such as JPEG [10], video compression schemes such as MPEG [4, 5] and recent video compression schemes such as H.264/AVC [6].

A concept that is often used in image and video compression is the concept of subsampling where information is sampled at a lower sampling frequency to reduce the amount of data to be stored. Images consist of color information (chroma) and brightness information (luma). Subsampling techniques generally fall into two categories:

1. Spatial Subsampling:
Visual content within an image or video frame is sampled at a lower resolution than the original
2. Temporal Subsampling:
The number of frames in a video is reduced by sam-

pling at a lower frame rate than the original content [7]. Temporal subsampling should not be confused with motion estimation and compensation which attempts to exploit the similarities between neighbouring frames in a video.

One of the spatial subsampling techniques widely used in modern image and video compression standards is chroma subsampling where resolution used for color information is lower than the resolution used for brightness information. The human vision system is less sensitive to color than brightness so color information can be sampled at a lower spatial frequency than brightness information without a noticeable degradation of visual quality. The compressibility of the total content is still limited by the fact that brightness information is not subsampled. However, a uniform subsampling of luma information often does not result in good visual quality, especially given the non-stationary nature of still images and videos. However, in certain situations, luma subsampling can be applied to a region of an image or video frame to improve compressibility without significantly degrading visual quality. This technique can be used to reduce the amount of data required by an image or video frame. If additional compression is not required, the savings in terms of data can be used to increase the visual quality of the image. An adaptive subsampling scheme that utilizes adaptive luma subsampling and uniform chroma subsampling is therefore highly desirable.

A number of adaptive spatial subsampling algorithms have been proposed [1, 8, 9, 3, 2]. However, there are some major drawbacks to these algorithms. The algorithms proposed by Belfor et al [1], and Liu et al [8] are designed to use subsampling as the only means of data compression. Both algorithms use complex techniques to achieve a high compression rate using only subsampling. The algorithm proposed by Belfor et al [1] makes use of multiple subsampling modes by evaluating all supported subsampling modes for each block and choosing a suitable mode based on a quality criterion. The computational cost of evaluating

all supported subsampling modes is often unreasonable in the context of video compression.

The algorithm proposed by Liu et al [8] utilizes a complex tree structure of variable-sized subimages to perform adaptive subsampling. The algorithm proposed by Panuspong et al [9], uses adaptive subsampling as well as other techniques to achieve image compression. A complex region growing of neighboring subimages is utilized to group subimages for subsampling in a tree structure. Both of these techniques are particularly difficult to integrate into existing video compression standards since blocks may need to be reconstructed based on motion estimation. Substantial structural changes are necessary to the base algorithms of these techniques to support motion estimation. Also, achieving an acceptable level of decoding performance may be a challenge in this context.

The algorithm proposed by Cheung [3] requires splitting the frequency content of images into two different layers (a low-pass layer and a high-pass layer). This is very difficult to integrate into existing image and video standards, given that blocks represent all frequencies of a region. The complexity of splitting video data into two different layers can be quite significant. The algorithm proposed by Chang et al [2] uses activity levels to determine when to adaptively subsample image data. This approach works well for some images and the computational cost is relatively small.

It should be noted that none of the approaches mentioned consider the influence of different perceptually important characteristics such as brightness upon the human perception of visual quality. Many of the techniques use rate distortion as a measure for determining whether a region is to be subsampled. However, such metrics do not reflect the characteristics of the human vision system and may not result in good perceptual quality. Furthermore, all of the approaches mentioned use a single metric to determine when adaptive subsampling should be used.

The main contribution of this paper is that it presents a practical algorithm for performing adaptive luma spatial subsampling in image regions based on a combination of three content characteristics (texture, edges, and brightness). The proposed algorithm has a low computational complexity making it suitable for use in video applications. The algorithm has been designed to be integrated into existing image and video standards without significant modifications to the standards. In this paper, the background behind the use of subsampling in modern image and video compression standards is discussed in Section 2. The proposed algorithm is described and explained in detail in Section 3. The testing method used to evaluate the proposed algorithm is presented in Section 4. Experimental results comparing the proposed algorithm to the sole use of uniform chroma subsampling are presented in Section 5. Conclusions and topics for future research are discussed in Section 6.

2. Subsampling

In modern block-transform compression standards, the source image or video frame undergoes an RGB-to-YCbCr color-space transformation, where Y represents the brightness (luminance) of a pixel, while Cb and Cr represent the color (chrominance). The chroma components (Cb and Cr) are then sampled at a lower spatial resolution than the brightness component Y, thereby requiring less information to be stored. This technique is commonly known as chroma subsampling. It is based on the fact that the human vision system is less sensitive to color than brightness. Therefore, color information can be sampled at a lower resolution without significant visual degradation. A number of different chroma subsampling schemes have been used in commercial image and video compression standards. Subsampling schemes are typically expressed using ratios to represent the relationship between the luma sampling frequency and the chroma sampling frequencies. For example, if the 4:2:2 chroma subsampling scheme is used, the color components are sampled at half the horizontal resolution of the brightness component compared to a 4:4:4 chroma scheme. Fig. 1 illustrates the 4:4:4 and 4:2:2 chroma subsampling schemes. A summary of the supported chroma subsampling schemes for popular block-transform image and video standards is shown in Table 1.

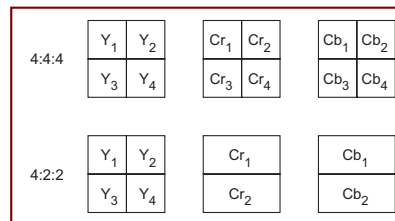


Figure 1. 4:4:4 and 4:2:2 Chroma Subsampling Schemes

3 Adaptive Subsampling Algorithm

The chroma subsampling schemes used in modern standards typically use ratios that are powers of 2. For the purpose of integrating adaptive luma subsampling into existing standards, it is therefore desirable to introduce a subsampling scheme that uses similar ratios. The proposed adaptive luma subsampling algorithm uses a 2:2:0 subsampling scheme where the brightness and color components are sampled at half the horizontal and vertical resolutions of the original content. The 4:2:0 and 2:2:0 subsampling schemes are shown in Fig. 2.

The proposed algorithm extends upon the concept of uniform chroma subsampling used in existing standards by in-

Table 1. Image and Video Compression Standards

Coding Standard	Support Chroma Formats
JPEG	4:4:4, 4:2:2, 4:2:0
MPEG-1	4:2:0
MPEG-2	4:4:4, 4:2:2, 4:2:0
MPEG-4 ASP	4:2:0
MPEG-4 AVC† (H.264)	4:4:4, 4:2:2, 4:2:0
VC-1	4:2:0

† depends on the profile chosen

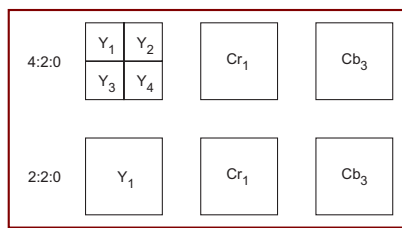


Figure 2. 4:2:0 and 2:2:0 Subsampling Schemes

Introducing an adaptive luma subsampling scheme that uses content characteristics from the original image or video frame. It is possible to exploit the perceptual sensitivity of the human vision system to key content characteristics such as texture activity, edge density, and brightness. As such, visual quality may be improved while reducing the amount of information that needs to be stored. The proposed algorithm is practical, relatively efficient, and capable of producing high visual quality. Furthermore, the proposed algorithm can be integrated into existing formats with ease. The proposed algorithm is illustrated in Fig. 3.

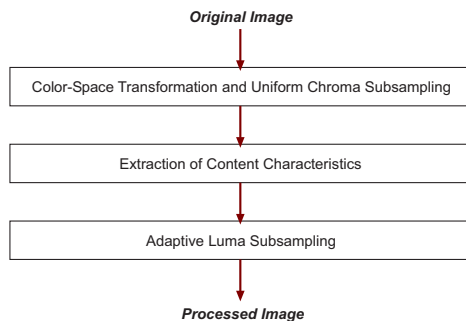


Figure 3. Content-Adaptive Subsampling Algorithm

3.1 Color-Space Transformation and Chroma Subsampling

During this stage, the color-space transformation and chroma subsampling procedures used in modern block-transform image and video compression are performed. The original image is converted from the RGB colorspace into the YCbCr colorspace. The chrominance components Cb and Cr are then uniformly subsampled based on the chosen chroma format. The subsampled Cb and Cr components are processed separately from the brightness component Y. For example, in MPEG-1 video, the supported chroma format is 4:2:0 so the Cb and Cr components are subsampled at half the horizontal and vertical resolutions of the Y component.

The Cb and Cr components are processed in the exact same manner as specified by the original compression standard. The human vision system cannot perceive as much detail from the chrominance components as it can from the luminance component. Therefore, the uniform chroma subsampling implemented in existing standards works well in the majority of cases. The increased complexity of adaptive chroma subsampling is not justifiable.

3.2 Extraction of Content Characteristics

During this stage, the original Y components are divided into N blocks of size $m \times n$ based on the compression standard used. For example, in MPEG-1 video, a macroblock consists of a 16×16 luminance block and two 8×8 chrominance blocks and so the Y component would be divided into N blocks of size 16×16 . The following perceptually significant content characteristics are used to determine which blocks are subsampled:

1. Edge density of the block
2. Texture activity of the block
3. Brightness of the block

3.2.1 Edge Density

To determine the edge densities for the blocks, the entire image is processed using an edge detection algorithm to create a binary edge map with edge and non-edge pixels being represented by 1 and 0 respectively. An edge density rating $ER(x)$ for a block x is calculated for each of the N blocks as follows:

$$ER(x) = E_x / N_x \quad (1)$$

where E_x is the number of edge pixels in block x , and N_x is the number of pixels in block x . Edges are crucial to the way the human vision system perceives and interprets the environment. As a result, the human vision system is very

sensitive to edge degradation in an image. Therefore, edge characteristics are important when deciding whether a block is to be subsampled. For example, a block with a high edge density rating should not be subsampled as it would lead to a very noticeable image degradation (e.g., edge aliasing artifacts). This is particularly important for non-photorealistic content (e.g., animation) where edges clearly define objects.

Blocks with an edge rating lower than a specified threshold T_{edge} are considered potential candidates for subsampling and undergo further evaluation. For the purpose of testing, a threshold of $T_{\text{edge}} = 0.1$ is used. This threshold was determined based on several subjective perceptual quality tests, where perceptual degradation was noticeable if the threshold was set at a higher value. If a block exceeds the threshold, there is no need to evaluate texture activity and brightness. To reduce the computational complexity of edge extraction, a simple edge detection algorithm such as Sobel edge detector is used.

3.2.2 Texture Activity

To determine the texture activity of a candidate block, a texture rating $TR(x)$ is calculated for block x as follows:

$$TR(x) = s_{\text{pixel}}^2(x) \quad (2)$$

where $s_{\text{pixel}}^2(x)$ is the spatial variance of pixel intensities in block x . Blocks with low texture activity such as smooth areas can be subsampled and reconstructed in the decoding stage with little perceived degradation compared to blocks with high texture activity. For animation and drawings, this is particularly important since large uniform regions commonly exist. Blocks with low texture activity ratings should be considered for subsampling. The spatial variance is used as a texture activity metric due to its low computational complexity.

3.2.3 Brightness

Finally, the overall brightness rating $BR(x)$ is calculated for block x as follows:

$$BR(x) = \mu_{\text{pixel}}(x) \quad (3)$$

where $\mu_{\text{pixel}}(x)$ is the sample mean of the pixel intensities in block x normalized to the range between 0 and 1. The human vision system is less sensitive to details in dark areas. Therefore, these blocks have a higher ability to conceal noise and image degradation that may result from subsampling. Blocks with a low brightness rating should be considered for subsampling. The sample mean is used as an overall brightness metric due to its low computational complexity.

3.3 Adaptive Luma Subsampling

During this stage, blocks with edge densities below the specified threshold (T_{edge}) are evaluated based on their texture rating and brightness rating. Blocks are subsampled if the following criteria are satisfied:

$$TR(x) < T_{\text{texture}} \text{ OR } BR(x) < T_{\text{brightness}} \quad (4)$$

For the purpose of testing, a brightness threshold of $T_{\text{brightness}} = 0.2$ is used. This threshold was determined based on several subjective perceptual quality tests, where perceptual degradation was noticeable if the threshold was set at a higher value. The texture threshold T_{texture} is set based on the compression standard used. All blocks that satisfy the criteria are sampled at half the horizontal and vertical resolutions of the original Y component, resulting in the 2:2:0 subsampling scheme. The subsampled blocks and non-subsampled blocks can be quantized differently for improved visual quality without the need for additional storage space since bits saved through subsampling can be redistributed to blocks that benefit from storing additional information.

3.4 Implementation Issues for Video Compression

Unlike still image compression standards such as JPEG, not all blocks in a video frame can be subsampled for the purpose of video compression. Motion estimation, where content from past and / or future frames is used to predict blocks in the current frame, complicates the subsampling of blocks in a video frame. A simple solution to the problem of subsampling in the presence of motion estimation in current video compression standards such as MPEG-1/2/4 and H.264/AVC involves the subsampling of only intra-blocks. Using this approach, all motion estimation is based on non-subsampled blocks combined with the reconstructed image content from subsampled blocks.

3.5 Implementation Complexity Analysis

It is important to consider the additional performance and storage overhead associated with the algorithm. In the actual implementation, a single bit flag is stored for each block to indicate whether the block is subsampled at 2:2:0 or not. These flags can be compressed together using run-length encoding and stored in the header of the current image or the header of the video frame. The additional data that needs to be stored has a negligible effect on the storage requirements. In terms of performance, the main additional processing task involves determining whether a block

is subsampled based on its bit flag and then performing interpolation, if necessary, to rescale the block to normal size. The same interpolation methods used for uniform chroma subsampling can be used for luma subsampling. Based on testing, the performance overhead of the proposed adaptive subsampling algorithm is less than 3% of the total decompression time.

4 Testing Methods

A comparison of the proposed algorithm to the standard method of solely using uniform chroma subsampling was conducted using quantitative and qualitative measures on a set of JPEG images and MPEG videos. The algorithms were tested using 8 different test cases ranging from stock images (e.g., Lena) to video frames captured from animations and live action videos. All test images and video frames use 24-bit RGB color representations. A brief description of each test case is provided below.

Lena: A 512×512 image of Lena.

Photo: A 450×338 photograph from a car's front windshield.

Animation 1: A 720×480 video frame from a science fiction animated cartoon.

Animation 2: A 720×480 video frame from a modern animated cartoon.

Live Action 1: A 720×480 video frame from a crime drama movie.

Live Action 2: A 720×480 video frame from the "Susi" test video clip.

Live Action 3: A 720×480 video frame from the "Flowers" test video clip.

Live Action 4: A 720×480 video frame from the "Tennis" test video clip.

To judge the performance of the proposed algorithm in a quantitative manner, a measure of the peak-signal-to-noise ratio (PSNR) was calculated for the test cases for both algorithms.

5 Experimental Results

The PSNR improvement results are shown in Table 2. The quantitative measurements show PSNR gains in the test cases when the proposed algorithm is used.

Fig. 4, Fig. 5, and Fig. 6 show some of the test cases processed using the two algorithms. The images and video frames demonstrate a qualitative improvement when the proposed algorithm is used. The edges are noticeably

cleaner and fine details are preserved in all regions. Furthermore, the output produced by the proposed algorithm has fewer visible blocking artifacts and better brightness consistency. The improved perception of quality results from data bits being used to preserve visual information in areas that the human vision system deems important.

6 Conclusions and Future Research

This paper has introduced a practical algorithm for content-adaptive subsampling of compressed images and video frames. The algorithm is computationally efficient, simple to implement, and easy to integrate into existing image and video compression formats. Experimental results show that overall visual quality is noticeably improved when compared to the sole use of uniform chroma subsampling. It is our belief that this algorithm can be successfully implemented in existing and future image and video compression standards. Future research will investigate the refinement of the characteristic thresholds in an attempt to improve image quality and compression performance further.

Acknowledgements

This research has been sponsored in part by Epson Canada and the Natural Sciences and Engineering Research Council of Canada.

References

- [1] R. Belfor, M. Hesp, R. Lagendijk, and J. Biemond. Spatially Adaptive Subsampling of Image Sequences. *IEEE Transactions on Image Processing*, 3(3):492–500, 1994.
- [2] I. Chang, P. Chang, and T. Liu. Adaptive Subsampling JPEG Image Coding. In *Proceedings of the International Conference on Consumer Electronics*, pages 264–265, 1995.
- [3] W. Cheung. Adaptive Subsampling for Image Data Compression. In *Proceedings of the IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering*, volume 2, pages 938–941, 1993.
- [4] ISO/IEC. Coding of Moving Pictures and Associated Audio for Digital Storage Media up to About 1.5 Mbits/s - Part 2: Video, November 1993.
- [5] ISO/IEC. Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video, November 1994.
- [6] Joint Video Team of ITU-T and I. JTC 1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, May 2003.
- [7] D. Kim, J. Kim, I. Ra, and Y. Choi. A New Video Interpolation Technique Based on Motion-Adaptive Subsampling. *Transactions on Consumer Electronics*, 45(3):782–787, 1999.

Table 2. PSNR Gain for Test Cases

Image	Compression Rate (bpp)	PSNR		PSNR Gain
		Uniform Chroma Subsampling (dB)	Proposed Algorithm (dB)	
Lena	0.39	28.69	29.19	+0.50
Photo	0.33	29.77	30.81	+1.04
Animation 1	0.58	26.36	26.83	+0.47
Animation 2	0.44	26.96	27.51	+0.55
Live Action 1	0.22	29.31	30.86	+1.55
Live Action 2	0.28	31.49	33.27	+1.78
Live Action 3	0.37	22.26	22.81	+0.55
Live Action 4	0.42	20.92	21.23	+0.31



(a) Uniform Chroma Subsampling



(b) Adaptive Subsampling

Figure 4. Subsampling results for a “Live Action 2” video frame at 0.22 bpp



(a) Uniform Chroma Subsampling



(b) Adaptive Subsampling

Figure 5. Subsampling results for a “Photo” image at 0.33 bpp



(a) Uniform Chroma Subsampling



(b) Adaptive Subsampling

Figure 6. Subsampling results for a “Tennis” video frame at 0.42 bpp

- [8] S. Liu and M. Hayes. Variable Resolution-Adaptive Sampling Approach to Image Coding. In *IEEE Proceedings of Southeastcon*, volume 2, pages 1096–1100, April 1991.
- [9] K. Panusopone, F. Cheevasuvi, and K. Rao. Adaptive Subsampling for Image Compression. In *Proceedings of the Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 239–243, 1995.
- [10] G. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–34, 1991.