Hybrid Video Compression Using Selective Keyframe Identification and Patch-Based Super-Resolution

Jeffrey Glaister, Calvin Chan, Michael Frankovich, Adrian Tang, Alexander Wong
Department of Systems Design Engineering
University of Waterloo
Waterloo, Canada
Email: {jlglaist, chcchan, mfrankov, a6tang, a28wong}@engmail.uwaterloo.ca

***Abstract*—This paper details a novel video compression pipeline using selective keyframe identification to encode video and patch-based super-resolution to decode for playback. Selective keyframe identification uses shot boundary detection and frame differencing methods to identify representative frames which are subsequently kept in high resolution within the compressed container. All other non-keyframes are downscaled for compression purposes. Patch-based super-resolution finds similar patches between an upscaled non-keyframe and the associated, high-resolution keyframe to regain lost detail via a super-resolution process. The algorithm was integrated into the H.264 video compression pipeline tested on webcam, cartoon and live-action video for both streaming and storage purposes. Experimental results show that the proposed hybrid video compression pipeline successfully achieved higher compression ratios than standard H.264, while achieving superior video quality than low resolution H.264 at similar compression ratios.**

***Keywords*-video compression; super-resolution; patch-based; keyframe**

I. Introduction

The impact of High Definition (HD) resolutions for media has become widespread in recent years. With respect to HD content, consumers are moving towards digital delivery solutions in lieu of physical media. HD has also infiltrated Internet services such as teleconferencing using new HD webcams. This creates problems in media delivery since increased quality implies that more information is being delivered, which ultimately results in increased bandwidth and storage. Therefore, new solutions must be developed to minimize the impact of the increased data and bandwidth requirements resulting from HD content delivery.

The problem of minimizing bandwidth consumption and storage of video content is referred to as video compression. The current standard for video compression is H.264/MPEG-4 AVC, which is used as one of the codec standards in the Blu-Ray format. Its compression gains are based on block-oriented motion-compensation [1]. In addition to H.264, there are many other solutions for video compression, but none use the application of super-resolution.

Super-resolution is traditionally used to enhance image details through exploiting spatial and temporal redundancy. In single frame super-resolution algorithms, details are enhanced by finding spatial redundancies [2] and in multi-frame super-resolution schemes, multiple images are used to enhance detail through temporal and spatial redundancies [3]. Many different approaches have been researched, including example-based algorithms requiring training data [4], interpolation-based algorithms [5], and regularization-based algorithms [6]. Recent research applies super-resolution to video compression. Algorithms include using a guided super-resolution technique to decode downsampled image sequences using motion and texture segmentation information [7] or using motion information from high-resolution keyframes to enhance downsampled normal frames [8].

However, one common problem that has been identified with using super-resolution in general is the amount of time required to process videos [9]. Current commercial software can take longer to enhance a video than the length of the video itself. The developed solution ideally approaches real-time encoding and decoding of the video.

This paper proposes a novel video compression pipeline using super-resolution processes that is able to achieve real-time quality recovery during decoding. This pipeline, split into an encoding phase and a decoding phase, is outlined in Fig. 1. The method builds on prior work done using super-resolution for video compression, performs encoding with selective keyframe identification and decoding using a patch-based approach to avoid explicit motion estimation and to increase processing time. The overall pipeline is referred to as Keyframe Super Resolution (KSR).

This paper is organized as follows. Section II outlines the selective keyframe encoding phase of the proposed compression pipeline. The patch-based super-resolution algorithm for decoding is described in Section III. Section IV contains the experimental results of the algorithm and comparisons to the current H.264 compression pipeline using three types of content: live action television, cartoons and web streaming. Finally, conclusions are drawn in Section V.

II. Selective Keyframe Encoding

The encoding phase is the component of the pipeline that improves the compression capability of the video compared to that achieved with H.264. Frames are downscaled within

978-0-7695-4589-9/11 $26.00 © 2011 IEEE
DOI 10.1109/ISM.2011.25

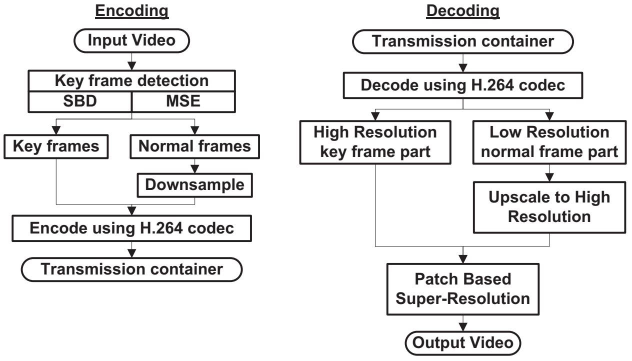


Figure 1. Encoding/Decoding Compression Pipeline

the video to decrease the amount of space required to store those specific frames. The encoding phase consists of two steps, shot boundary detection and frame differencing, to detect keyframes. Finally, the non-keyframes are processed and the video is re-encoded for transmission or storage.

A. Shot Boundary Detection

A shot boundary is defined as an abrupt or gradual transition between two contiguous sequences of frames in a continuous period of time. Abrupt transitions occur between two subsequent frames and gradual transitions span multiple frames. Shot boundaries are identified using the regional statistical differences method [10], which divides the frame into 16 sub-regions and calculates the mean of the grayscale pixel intensities within each sub-region. The difference between the mean of each sub-region over the two frames is determined and the average mean across the 16 sub-regions is calculated. This difference is then compared to a pre-defined threshold. If the mean is greater than the threshold, the frames are considered a transition and the second frame in the sequence is marked as a keyframe.

B. Frame Differencing

The second step of the process identifies drastic content changes that exist within a shot using a frame differencing algorithm. The difference in pixel intensities (scaled to be between 0 and 1) between the last detected keyframe and the current frame are analyzed and quantified as the Mean Squared Error (MSE). If the MSE is greater than 0.01, the frames are considered sufficiently different and the current frame is marked as a keyframe, replacing the last keyframe for future iterations of the frame differencing algorithm. Again, it should be noted that the threshold mentioned above was determined through experimentation, and is optimized to work well for cartoons and low motion video content.

C. Encoding Process

Once all keyframes have been identified, they are stored contiguously in an Audio Video Interleave (AVI) container and compressed with the H.264 codec. All non-keyframes, also called interpolated frames, are downscaled by a factor of four and subsequently stored in a separate AVI container to be compressed with the H.264 codec as well. Both AVI containers and a meta-data file containing the position of each keyframe in the original video stream are placed in a new container. By downscaling the interpolated frames before encoding with H.264, the resulting file size of the new container is inherently smaller than the original full resolution video compressed using H.264 only.

III. Patch-Based Super-Resolution Decoding

The decoding phase reduces the visible degradation caused by the compression in the encoding phase by applying a patch-based super-resolution algorithm on the downscaled frames to enhance the visual quality using the associated full resolution keyframes. The first step in this phase is to unpack the container with the keyframes and interpolated frames, and decode both encoded AVI files using H.264. Information contained in the meta-data file is used to place the keyframes in their proper position relative to the interpolated frames.

Since the interpolated frames were downscaled by a factor of four, the next step is to upscale those frames to the proper resolution using a simple upscaling technique. It was found through experimentation that a Lanczos kernel with $\alpha = 3$ was an appropriate method, but this results in visible compression artifacts and blurred edges.

The patch-based super-resolution algorithm is then applied by identifying the most representative keyframe to the current interpolated frame. The MSE between the current interpolated frame and the nearest keyframes are calculated. The keyframe with the lower MSE is considered most representative. The patch-based method is applied to the interpolated frame, to recover the detail lost due to compression. This patch-based super-resolution algorithm is illustrated in Fig. 2. Each pixel in the interpolated frame is analyzed and adjusted based on information taken from the spatio-temporal region surrounding the same pixel location in the keyframe. Due to how the encoding process selects keyframes, the high resolution frames contain similar visual information to the interpolated frame and redundant information is used to enhance the video without motion estimation.

For a pixel of interest in the interpolated frame, a search space was created in the keyframe centered at the same location as the pixel of interest. For each pixel in the search space, patches were created centered at that pixel. The MSE between each patch in the spatio-temporal search space and the patch corresponding to the pixel of interest in the interpolated frame were calculated to find the most representative patches. A 5x5 patch size and an 11x11 spatio-temporal search space size were determined experimentally to work well for finding highly representative patches from the keyframe. The MSE values were also used to calculate a

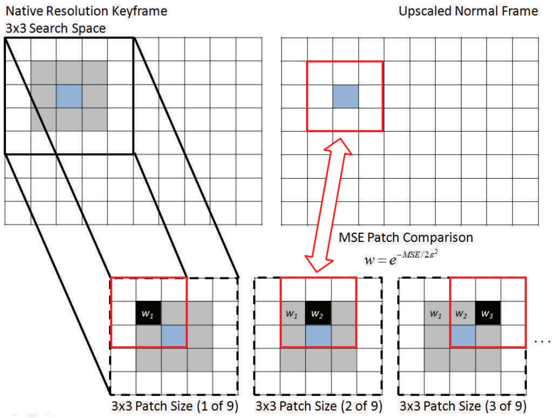


Figure 2. Patch-Based Super-Resolution

weight, given by Equation 1, where ϵ is the allowable error and is used to adjust the relative weighting.

$$w = e^{\frac{-MSE}{2\epsilon^2}} \tag{1}$$

The contribution from the patches become greater as the MSE decreased. The 11 best patches within a 0.01 MSE threshold were used to update the pixel of interest. The weight associated with the current pixel of interest was equal to the maximum weight associated with the other pixels in the search space. In the end, the pixel of interest was updated by taking a weighted average, considering the weights and center pixel intensities from each relevant patch.

IV. Results

To investigate the performance of the proposed method, nine test videos containing three types of content (live-action, cartoon and webcam video) were processed. The compression ratio improvement (versus standard H.264) and visual quality of the processed video were analyzed to compare KSR to current compression standards. The set of test videos consists of i) three live-action videos, ii) two cartoon videos, and iii) four webcam videos. The videos were compressed using the proposed compression pipeline.

The compression ratio improvement was calculated for each video, comparing file sizes of KSR H.264 and low resolution H.264 (LR H.264) videos. The average compression ratio for each video type is shown in Table I. Visual quality of the videos was qualitatively assessed and Figures 3-5 show examples of processed low motion videos.

Similar compression ratio improvements were achieved for all three types of videos. The videos produced using the proposed hybrid compression pipeline were noticeably more detailed and less noisy than the standard H.264 downsampling pipeline. Live-action videos had high compression ratio improvements but noticeable compression artifacts, while cartoons tended to have the best visual quality. The algorithm can be adjusted to have a higher encoding threshold and more keyframes, which would increase visual quality and decrease compression ratio improvements.

Table I
Compression ratio improvement of KSR H.264 over standard H.264

Video Type	Compression Ratio Improvement	Percentage Keyframes
Live-action	3.61:1	3.7%
Cartoon	3.06:1	11.4%
Webcam	2.95:1	13.3%

V. Conclusion

In this paper, we have proposed a novel compression algorithm using selective keyframe detection to encode video and patch-based super-resolution to decode. The encoding process identifies keyframes and compresses non-keyframes. The patch-based super-resolution algorithm finds similar areas in the keyframes and the non-keyframes to recover detail in the non-keyframes. Future work includes using techniques such as foveation to decrease processing time.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, *Overview of the H.264/AVC Video Codec Standard*, IEEE. Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560576, July 2003.

[2] D. Glasner, S. Bagon, and M. Irani, *Super-Resolution from a Single Image*, International Conference on Computer Vision (ICCV), October 2009.

[3] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, *Fast and Robust Multiframe Super Resolution*, IEEE Trans. on Image Proc., vol. 13, no. 10, pp. 1327-1344, September 2004.

[4] X. Li, K. Lam, G. Qiu, L. Shen and S. Wang, *Example-based image super-resolution with class-specific predictors*, J. Vis. Commun. Image R. Volume 20, pp 312-322, 2009.

[5] S. C. Park, M. K. Park, and M. G. Kang, *Super-Resolution Image Reconstruction: A Technical Overview*, IEEE Signal Processing Magazine, vol. 20, issue 3, pp. 21-36, May 2003.

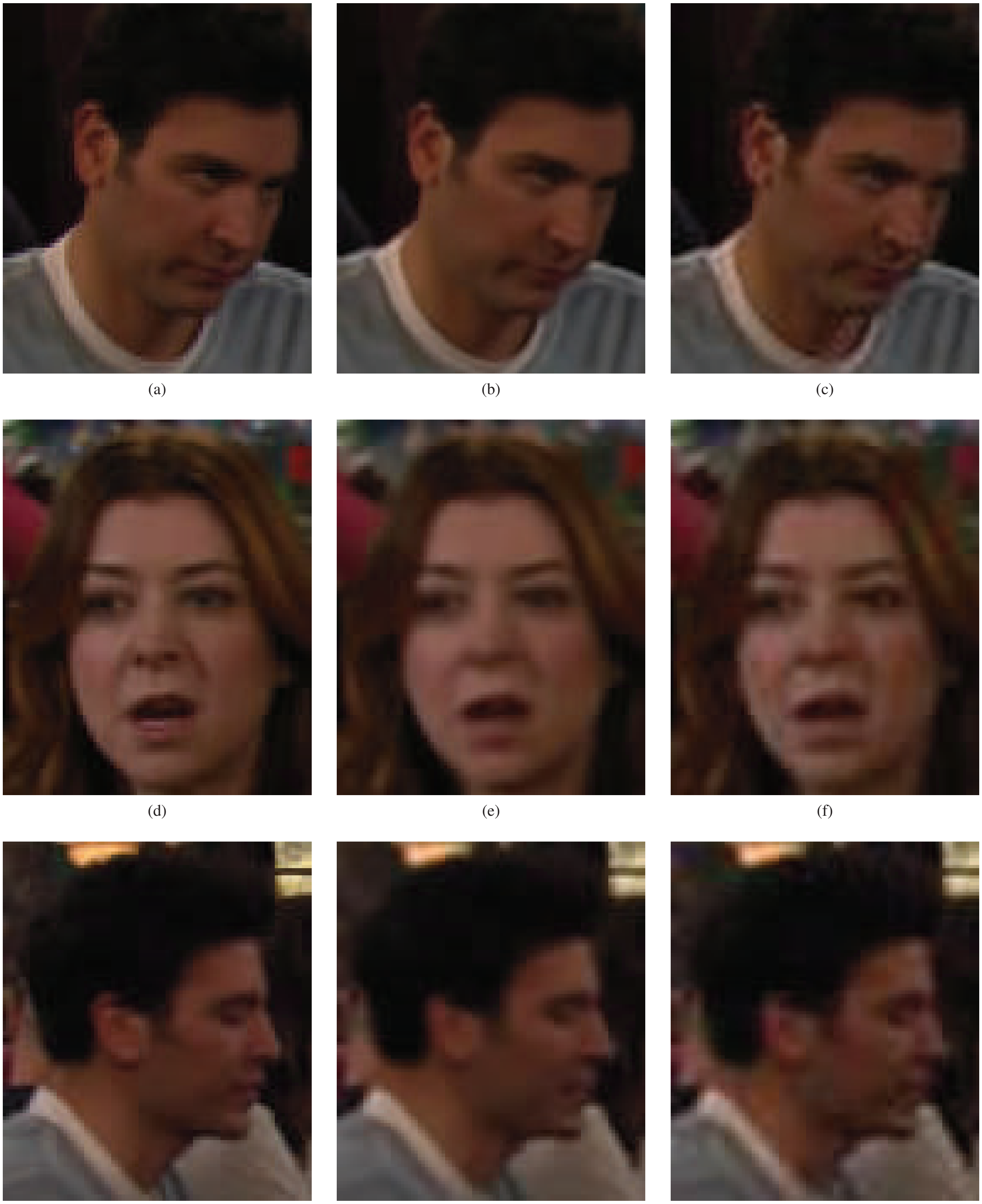
[6] T. F. Chan and C. K. Wong, *Multichannel image deconvolution by total variation regularization*, Proc. of SPIE, vol. 3162, pp.358366, 1997.

[7] D. Barreto, et al. *Region-based super-resolution for compression*, Multidimensional Syst. And Signal Process., Volume 18, No. 2-3, pp. 59-81, February 2007.

[8] F. Brandi, et al. *Super Resolution of Video Using Key Frames*, 15th IEEE Int. Conference on Image Process., pp. 321-324, October 2008.

[9] S. Farsiu, D. Robinson, M. Elad and P. Milanfar, *Advances and Challenges in Super-Resolution*, Int. J. of Imaging Syst. and Technology, vol. 14, no. 2, pp. 4757, August 2004.

[10] J. S. Boreczky and L. A. Rowe, *Comparison of video shot boundary detection techniques*, Journal of Electronic Imaging, Volume 5, Issue 2, pp 122-128, April 1996.



(a) (b) (c)

(d) (e) (f)

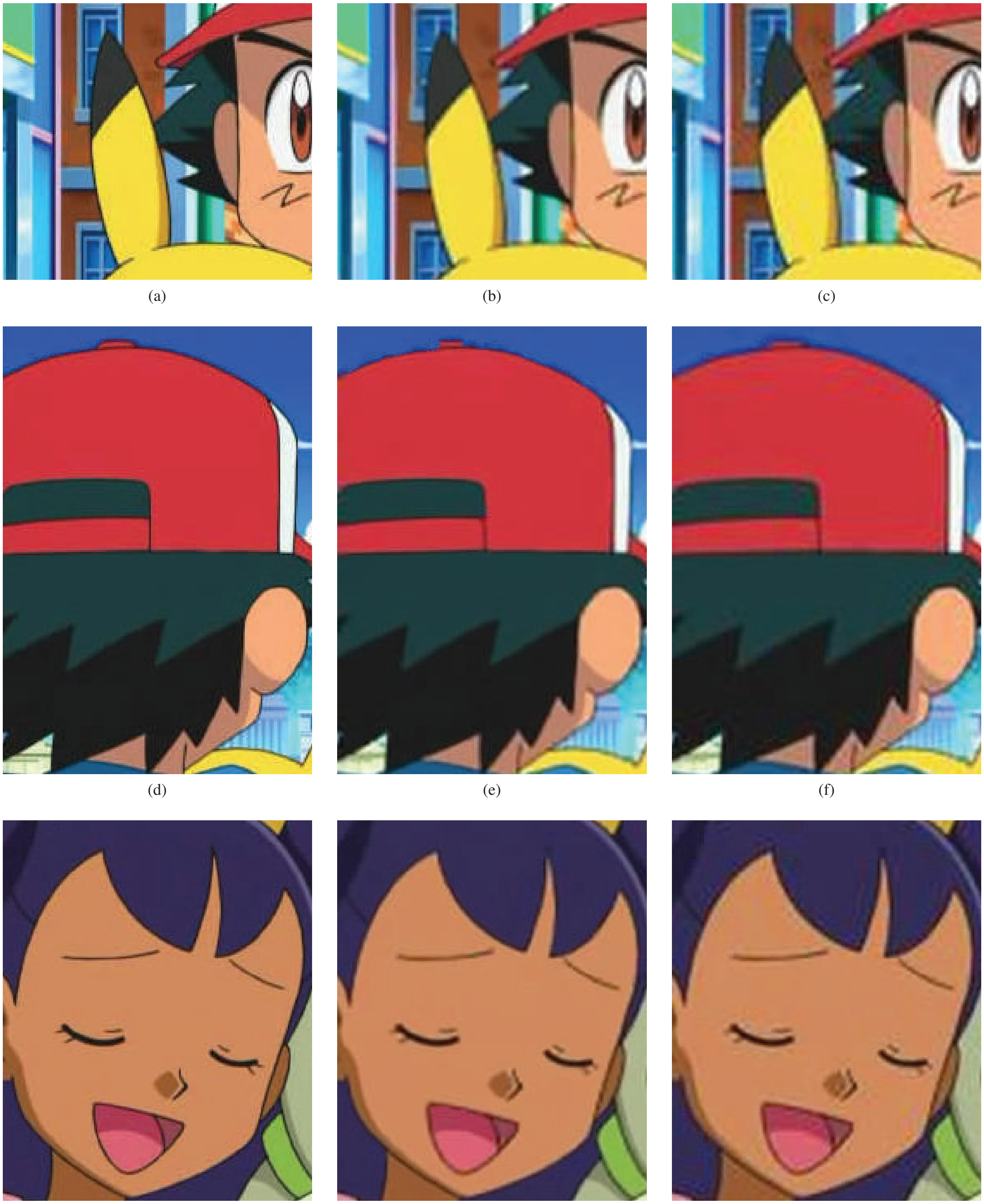
(g) (h) (i)

H.264

KSR H.264
Compression Ratio Improvement: 3.70:1

LR H.264
Compression Ratio Improvement: 4:1

Figure 3. An example of a processed live-action video. H.264 (a, d, g), KSR H.264 (b, e, h) and LR H.264 (c, f, i) are shown. ©Bays & Thomas Productions



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

H.264

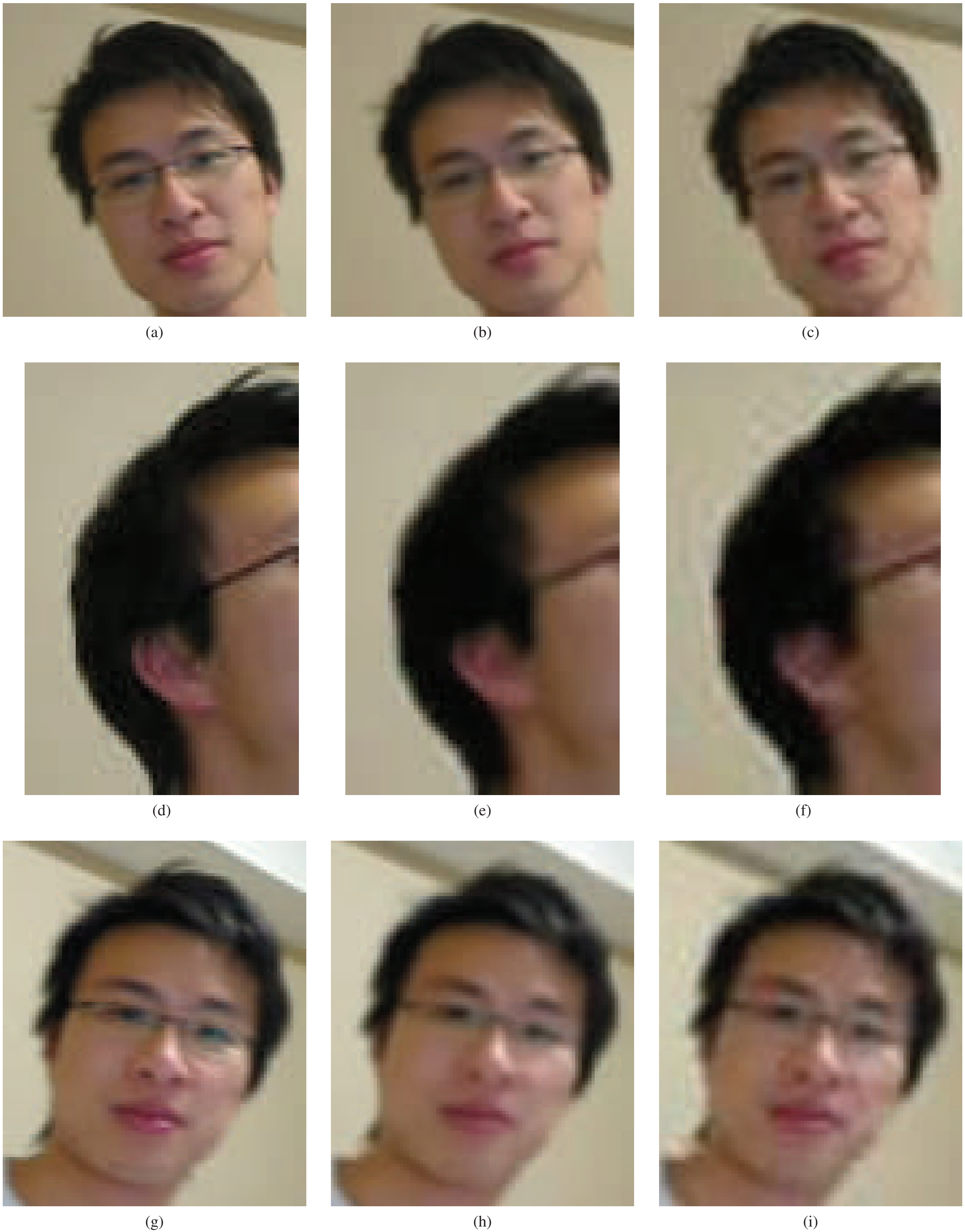
KSR H.264

Compression Ratio Improvement: 3.52:1

LR H.264

Compression Ratio Improvement: 4:1

Figure 4. An example of a processed webcam video. H.264 (a, d, g), KSR H.264 (b, e, h) and LR H.264 (c, f, i) are shown. ©OLM, Inc



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

H.264

KSR H.264
Compression Ratio Improvement: 3.27:1

LR H.264
Compression Ratio Improvement: 4:1

Figure 5. An example of a processed webcam video. H.264 (a, d, g), KSR H.264 (b, e, h) and LR H.264 (c, f, i) are shown.