Video Quality for Face Detection, Recognition, and Tracking

PAVEL KORSHUNOV and WEI TSANG OOI
National University of Singapore

Many distributed multimedia applications rely on video analysis algorithms for automated video and image processing. Little is known, however, about the minimum video quality required to ensure an accurate performance of these algorithms. In an attempt to understand these requirements, we focus on a set of commonly used face analysis algorithms. Using standard datasets and live videos, we conducted experiments demonstrating that the algorithms show almost no decrease in accuracy until the input video is reduced to a certain critical quality, which amounts to significantly lower bitrate compared to the quality commonly acceptable for human vision. Since computer vision percepts video differently than human vision, existing video quality metrics, designed for human perception, cannot be used to reason about the effects of video quality reduction on accuracy of video analysis algorithms. We therefore investigate two alternate video quality metrics, blockiness and mutual information, and show how they can be used to estimate the critical video qualities for face analysis algorithms.

Categories and Subject Descriptors: I.2.10 [**Vision and Scene Understanding**]: Video Analysis; C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms: Experimentation; Measurement; Performance;

Additional Key Words and Phrases: Video Analysis Algorithm, Video Quality, Blockiness, Mutual Information, Video Surveillance

1. INTRODUCTION

Humans are the typical end recipients of images and video footage. Recently, however, various systems and applications started to rely on video analysis algorithms to automate their tasks. Such systems include video surveillance, autonomous vehicles, and applications running on small wireless mobile devices. For example, a video surveillance system can automatically analyze video, without alerting the human guard, until a suspicious event occurs; an autonomous aircraft or vehicle can detect, track, and follow a target based on results of the video analysis; a mobile phone with camera used for automated tagging of friends on a photo or identification of the current location based on analysis of a landmark captured in the picture.

Typically, these automated systems have more than one remote video sensor unit (surveillance camera, mobile camera-phone) and either a control server or a set of remote proxies that process information received from the units. There can be two options regarding where the video is analyzed. The first option is to analyze the video at the sensor and transmit the results to the server. The second option is to transmit the video streams to the control server and analyze the video at the server. Each option has its advantages and disadvantages. Running video analysis on the sensors consumes little bandwidth but requires complex and computationally-capable sensor units, increasing cost and complicating system maintenance. Analyzing the video at the server leads to cheaper and more energy efficient sensor units, but streaming video poses a higher demand on network bandwidth. Therefore, a tradeoff exists between cost and energy efficiency of video sensors and network bandwidth.

Since the end recipients of video in these systems are non-human, the video quality, and hence the bitrate, can possibly be reduced. Typical digital surveillance cameras produce video that has spatial resolution of no less than 320×240, high SNR, and 10-30 fps. Such quality is normally the minimal desirable quality for the human visual system. Although it is shown that humans can perform recognition tasks for lower video quality [Rouse and Hemami 2008], typical surveillance video is recorded at the highest affordable quality to ease visual monitoring. We hypothesize that some video analysis algorithms should not require such high video quality to perform accurately. For such algorithms, we can reduce the quality of the video, thus reducing its bitrate, allowing us to use cheap video sensors and save on network bandwidth.

To verify this hypothesis, we conducted experiments with several commonly used and freely available face analysis algorithms to find their requirements on the input video quality. We determined how much of the input video quality can be reduced (through frame dropping, compression, spatial scaling, or their combination) without reducing the resulted accuracy of these algorithms. Since we used only several algorithms, we do not claim the generality of our results. We believe, however, that this paper is the first attempt to conduct an extensive study of video quality requirements for video analysis algorithms.

To determine video quality requirements, we use the following face analysis algorithms: (i) Viola-Jones [Viola and Jones 2001] and Rowley [Rowley et al. 1998] face detection algorithms, (ii) QDA-based face recognition algorithm [Lu et al. 2003], and (iii) CAMSHIFT [Bradski 1998] face tracking algorithm. We choose these algorithms, because they are non-trivial and freely available for experiments. Also, for face detection and face recognition, the standard test data with ground truth is available, making the evaluation of these algorithms easier. We measure the changes in accuracy for each of the studied algorithms when fed with input video at different qualities. To change the input video quality, we use various *video adaptations*, such as JPEG compression[1], frame dropping, as well as bicubic, nearest neighbor, and pixel area relation spatial scaling. We find that the algorithms show almost no degradation in accuracy until a certain quality threshold, which we term *critical video quality*, is reached. We demonstrate that encoding video with critical video quality can amount to significant video bitrate reductions (e.g., 23 times for Viola-Jones face detection algorithm).

Since different quality-degrading video adaptations can be used, it is important to have a common definition of video quality, specifically SNR quality, so that we can

[1] We use library by Independent JPEG Group (IJG)

reason about the effects of these adaptations on different video analysis algorithms. The existing SNR quality metrics, such as PSNR, SIMM, and PQV, are not suitable, because they are developed to measure quality from the human perspective. Different analysis algorithms have different meaning of the term “video quality”, which also differs from its conventional meaning in terms of human perception.

A video quality metric that matches the “perception” of video analysis algorithms needs to have the following properties. For a given algorithm, the quality metric should help in identifying critical video qualities corresponding to different video adaptations. The metric should be easy to compute, and it needs to be general enough to suit different types of video analysis algorithms and video adaptations. Optionally, the metric’s value can reflect the reduction in video bitrate, allowing us to compare different video analysis algorithms in terms of their efficiency and tolerance to low video quality.

In this paper, we consider two video quality metrics, blockiness and mutual information, and examine them for the above properties. Blockiness is one of the common video artifacts introduced by compression algorithms. Other video artifacts include blurriness, color bleeding, and sharpness. Mutual information measures a general loss of information in the degraded image and is more independent from the type of compression used. We show that with the help of these metrics, we can estimate critical SNR video quality for our video analysis algorithms, without resorting to exhaustive experimental search, which is not attractive to use in practical systems. In our experiments, blockiness demonstrates higher precision in estimation of the critical quality. Mutual information is less precise compared to the blockiness but is more independent from the choice of video adaptation.

We also show that compression algorithms can be simplified if the resulting image or video is to be used as input to video analysis algorithms instead of human vision. We demonstrate it by simplifying JPEG quantization table and showing that critical SNR quality corresponding to the modified version of JPEG leads to the same reduction in bitrate as the original JPEG compression.

The following summarizes our contributions:

- We have demonstrated that each algorithm used in our experiments has a critical video quality. Video encoded with such quality has very low bitrate compared to conventional video for human visual system. It does not, however, significantly affect the accuracy of the algorithms.
- We argue for the need of alternative video quality metrics that are suitable for video analysis algorithms. We present experimental results illustrating the advantages and disadvantages of using blockiness and mutual information as video quality metrics.
- We use simpler JPEG tables to compress images for face detection, demonstrating that it is possible to develop compression algorithms better suiting the requirements of computer vision.

In the next section, we discuss the related work in bitrate reduction of transmitted video and approaches to video streaming in practical systems. In Section 3, we give a detailed overview of the experiments conducted in the paper. In Section 4, we present the results of finding critical video quality for the studied face analysis algorithms. In Section 5, we propose estimating the critical SNR quality using blockiness and mutual information. We conclude the paper with Section 6.

2. RELATED WORK

Many techniques were proposed for adapting video transmission rate to meet bandwidth constraints. One of the first suggested methods, presented by Eleftheriadis and Anastassiou [Eleftheriadis and Anastassiou 1995], uses a rate-distortion function to find minimal distortion. Based on the bandwidth capacity predicted via monitoring the current state of the network, the video is dynamically reshaped with different quantization values. Extending this idea, Kim and Altunbasak [Kim and Altunbasak 2001] suggested a technique to reshape video by scaling its spatial, temporal, and SNR properties. This technique was later generalized into a utility-based framework by Kim *et al.* [Kim et al. 2003]. These approaches aimed at reducing the time and complexity of re-encoding the video for the network with limited bandwidth. In this paper, we adapted some of these ideas, though we focus on the case where the end recipients are video analysis algorithms rather than human.

Many latest video and image coders support region of interest (ROI) coding, which allows encoding different regions with different quality [Schumeyer et al. 1997; Sanchez et al. 2004]. For instance, a static background can be encoded with lower quality than a moving foreground object, drastically reducing overall video transmission. The major challenge with this approach, however, is to identify such region of interest. Model-based and object-based coding research aims to solve this problem for various video streaming applications. A model-based coder, commonly used in video telephony, encodes a parameterized 3-D head model that is represented through facial animation parameters (FAP) derived from the video. Smolic *et al.* [Smolic et al. 1999] proposed to estimate these parameters based on a set of feature points tracked for every video frame; while Eisert and Girod [Eisert and Girod 1998] estimated FAPs based on hierarchical optical flow. Eisert *et al.* [Eisert et al. 2000] extended their work to combine 3-D model-based coding and wavelet-based coding. Hakeem *et al.* [Hakeem et al. 2005] proposed object-based coding method that does not require a 3-D or 2-D model of an object. The authors suggested using a generic contour-based tracker together with background modeling for extracting a moving foreground object. All these ROI-based coding methods require cameras to have high computational power in order to detect, describe, or track an object of interest. In this paper, however, we assume cheap video sensors with little computational ability, while video analysis and heavy computations are performed at remote processing servers, which, in turn, can employ ROI-based encoders.

Research in video surveillance also proposed several solutions for reducing the amount of data transmitted over network. Yuan *et al.* [Yuan et al. 2003] and Nair *et al.* [Nair and Clark 2002] presented systems that avoid using excessive network bandwidth by periodically sending still images from a video source to the end user. VSAM [Collins et al. 2000] deals with bandwidth constraint by sending only one low quality video at a time and relies on workstations attached directly to video sources for the detection, tracking, and classification of events. Such solutions limit the amount of visual information that is available to the viewer and are not scalable. The authors of many recent surveillance systems, for example SfinX [Rangaswami et al. 2004] and KNIGHT [Javed et al. 2003], did not address the problem of video streaming and, instead, focus on developing more accurate video analysis algorithms. The authors of DOTS surveillance system [Girgensohn et al. 2007], acknowledge the scalability problem in their indoor surveillance system, allowing only 15 video cameras streaming simultaneously. This paper addresses the problem of video streaming in surveillance systems by taking advantage of video analysis algorithms' tolerance to low video quality. Since in typical surveillance scenario, suspicious events are rare [Wu et al. 2003], it is not necessary for human to constantly observe all video streams but only those that require his/her attention. Therefore, most of the time, the video is transmitted for video analysis algorithms only, allowing us to significantly reduce its quality and, hence, increase the scalability of the surveillance system.

3. EXPERIMENTS OVERVIEW

Before proceeding to describe our results, we first explain in more details how our experiments were conducted. To determine the video quality requirements for a particular video analysis algorithm, we degrade the original video to a point when the accuracy of the algorithm drops significantly. We call such point a *critical video quality*, indicating the quality threshold above which the algorithm performs with its original accuracy. The video is degraded in small steps with a video adaptation, such as JPEG compression or frame dropping. It should be noted that, in this paper, we understand the accuracy of an algorithm as a *relative* measurement. It refers to how much the accuracy changes when the video is degraded from its original quality.

We use the following video analysis algorithms in our experiments: OpenCV[2] implementation of Viola-Jones [Viola and Jones 2001] face detection, Rowley face detection [Rowley et al. 1998] algorithms; QDA-based face recognition [Lu et al. 2003] algorithm; and CAMSHIFT (OpenCV) face tracking [Bradski 1998] algorithm. We picked these algorithms because they are freely available, fairly complex, and commonly used in various applications. Also, for face detection and face recognition algorithms, there are standard datasets with ground truth available.

Test Data. Datasets used in our experiments are summarized in Table I. We use standard MIT/CMU and Yale datasets with provided ground truth for testing the accuracy of face detection and face recognition algorithms respectively. For face recognition, typically, the set of images is divided into gallery and probe subsets. Images in gallery have faces that are assumed to be known at the moment of recognition and images in probe set contain faces that are being recognized by the algorithm. To avoid bias in our recognition results, we divide the Yale dataset into four randomly generated pairs of gallery (36% images) and probe (64% images) subsets; our experimental results are obtained as average values corresponding to four subset pairs. For face tracking algorithm, due to the lack of standard test videos, we use our own videos of a face captured with a web-cam (see a snapshot

[2] http://www.intel.com/research/mrl/research/opencv/

Dataset	Characteristics	Short description	Algorithm
MIT/CMU (subset A)	images with 168 faces of different sizes	various background/lighting conditions	face detection
Yale	images with 165 faces of 15 people, 320× 240	various lighting conditions, different facial expressions	face recognition
Videos with moving faces	video, 600 frames, 352× 288, 30 fps	office settings, web-cam, face moves close-far from the camera	face tracking
Video of the lab door	video, 22000 frames, 320× 240, 5 fps	surveillance of the door in a research lab	face detection, face recognition

Table I. Summary of datasets used in the experiments with different video analysis algorithms.

example in Figure 6). We also test face tracking on movies and news clips.

To test face detection and face recognition algorithms in practical scenario, we recorded a one hour video of the door in our research lab (see the last row of Table I), simulating an indoor video surveillance system. We used Cannon VCC4 camera with the default video quality settings (320×240 resolution and JPEG compression 90). An example of a frame from this video sequence is shown in Figure 13(a). Among the recorded 22,000 frames, we manually marked 237 faces as ground truth, including 138 frontal and 99 profile faces. This set of frames is used to verify critical video qualities estimated for face detection and face recognition algorithms using mutual information metric. We evaluate the recognition algorithm by using the standard verification performance metric [Grother et al. 2003]. Frames with detected faces, including false positives, are used as the input probe faces. For each person in the test videos, one representative face is pre-selected and is used in verification matching of the probe faces in the video.

Video Adaptation and Algorithm Accuracy. Table II summarizes the video adaptations used to change the video quality for our video analysis algorithms. SNR video quality is degraded with IJG[3] implementation of JPEG compression algorithm. In this implementation, compression quality 1 corresponds to images with the highest compression ratio (the most distorted image) and 99 to images with the lowest compression ratio (the least distorted image). To evaluate accuracy of face detection, we compute the detection index as follows. For each JPEG quality, the number of detected faces is recorded. Using available ground truth, we obtain the number of correctly detected faces and divide it by the recorded total number of faces to get the detection index. We also divide the number of faces that are wrongly detected by the algorithm by the total number of faces to obtain the false positive index of face detection.

In experiments with Yale dataset, the identification task of face recognition algo-

[3] http://www.ijg.org/

Video adaptation	**Degradation pattern**	**Algorithm**	**Dataset**
JPEG compression	quantizer from 1 to 99 with step 2	face detection, face recognition	MIT/CMU, Yale
	quantizer from 10 to 100 with step 10	face tracking	videos with moving faces
Scaling (nearest neighbor, bicubic, pixel area relation)	2 to 100 percent of original size with step 2	face detection, face recognition	MIT/CMU, Yale
Combination of JPEG compression and nearest neighbor scaling	scaling from 10 to 100 percent of original size with step 10, compressing at each step from 1 to 99 with step 2	face detection, face recognition	Video of the lab door
Frame dropping	drop i out of $i + j$ frames	face tracking	Videos with moving faces

Table II. Summary of video adaptations used in the experiments with different video analysis algorithms.

rithm is evaluated using the standard performance metric, rank one of cumulative match characteristic (CMC) [Grother et al. 2003]. CMC rank one value is computed for images from the probe set only. In experiments with practical surveillance video, we evaluate the recognition algorithm by using the standard verification performance metric instead. We also degrade SNR quality of videos used to test face tracking algorithm. We use Microsoft Video 1 codec to compress videos, changing its quantizer value from 10 (higher distortion, low quality) to 100 (best quality) with step 10.

OpenCV implementation of nearest neighbor, bicubic, and pixel area relation scaling algorithms are used for reducing spatial video quality. The change in spatial quality is tested for face recognition algorithm and Viola-Jones face detection algorithm (only in some experiments). With a given scaling algorithm, we reduce spatial sizes of images from 100 to 2 percent of the originals with step size 2. Then, we scale them back to the original sizes. Such downscaling-upscaling transformation can be used in a practical scenario of distributed video surveillance system, as demonstrated in Figure 1. The video with reduced spatial size is sent by a camera to a proxy through network. Upon receiving a video frame, the proxy upscales it to its original size and runs a video analysis algorithm. The downscaling-upscaling of the video stream allows us to reduce amount of data transmitted across the network link between camera and proxy/server.

The scaling can also be combined with JPEG compression as we demonstrate for face detection and face recognition algorithms in Section 5.2. We combine nearest neighbor scaling with JPEG compression in the following way. Images are first prescaled to several spatial sizes (20%, 30%, 40%, etc.) after which they are compressed with JPEG quantizer varying between 1 and 99 with step 2. Then, images

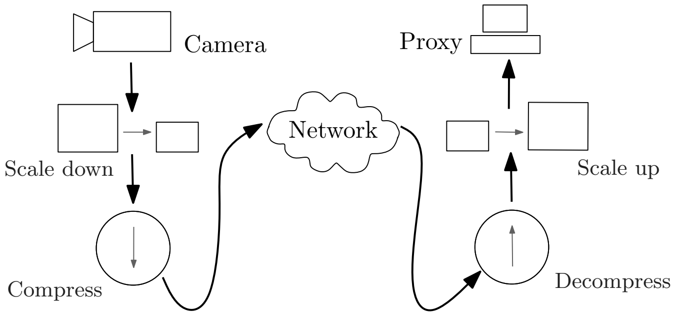


Fig. 1. Video surveillance scenario of combining scaling and compression adaptations to further reduce bitrate.

are decompressed and scaled back to their original spatial sizes. Compressing down-scaled video frame allows achieving even higher reduction in bitrate at the expense of receiving frame with higher distortion at the proxy.

For face tracking algorithm, we change the video frame rate of a test video by dropping frames from the original video using drop pattern: "drop i out of $i + j$ frames" (see Figure 2 for illustration). We vary i and j from 1 to 14. The value i represents the gap between frames, and j represents how many consecutive frames remain. For example, if we drop every third frame, i equals to 1 and j to 2; when three consecutive frames out of nine frames are dropped, i is 3 and j is 6. Note that while these two patterns give the same average frame rate, the accuracy of the tracking algorithm can be different.

We compute the accuracy of face tracking algorithm as follows. The mean distance between the center of the tracked face in degraded video (with applied drop pattern) and the center of the face in the original video is recorded. We use the ratio of this mean distance and half the average diagonal of the tracking rectangle as a metric of accuracy for the tracking algorithm. Essentially, it measures the error of tracking across all frames in the video. Therefore, we call the metric *average error*. For a given dropping pattern, smaller average error means better relative accuracy of the face tracking algorithm.

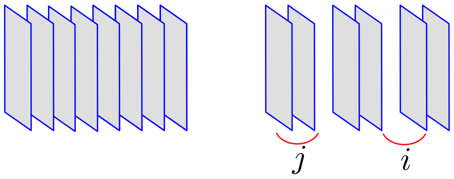


Fig. 2. Dropping i out of $i + j$ frames. i is the drop gap.

4. CRITICAL VIDEO QUALITY

In this section, we present experimental results of finding critical video quality for our examples of video analysis algorithms. The main focus of the experiments is on

Viola-Jones face detection, CAMSHIFT face tracking, and QDA-based face recognition algorithms. Only selected experimental results are presented for Rowley face detection algorithm, due to the lack of space and their similarity to Viola-Jones's results. For the same reason, we also emphasize more on pairs of SNR quality and face detection, spatial quality and face recognition, and temporal quality and face tracking.

4.1 Face Detection

First, we investigate how the accuracy of Viola-Jones and Rowley face detection algorithms change when SNR quality of the video is reduced. Viola-Jones algorithm is an object detection algorithm that uses a cascade of classifiers based on Haar-like features [Viola and Jones 2001]. Intuitively, it should perform accurately as long as images contain such features. Rowley algorithm is based on the statistical changes in intensity values across a given image. Those regions that reflect the patterns collected through algorithm's training are marked as a face. We present experimental findings showing changes in accuracy of these two algorithms for degraded SNR quality (see Section 3 for more detailed description of experiments).

The experimental results for Viola-Jones algorithm and MIT/CMU dataset are presented in Figure 3(a). The figure shows both detection and false positive indexes of the face detection algorithm against different compression qualities. It can be noted that the average accuracy of the face detection algorithm does not change significantly when JPEG compression quality is decreased from 99 to 9 (indicated with the dashed vertical line on the figure). For quality less than 9, the detection index demonstrates a sharp decrease. Since 90-95 is the default JPEG compression quality used in typical video surveillance cameras (e.g., Axis 207, Canon VCC4), compressing images to quality 9 can lead to significant reduction in size. Also note that the false positive index does not increase in response to reduced compression quality, which means that only the detection index is affected. Therefore, we can transmit video frames compressed with quality 9 and achieve similar detection results as with uncompressed video. If we conservatively choose 20 as the critical compression quality, we find that the average file size of JPEG images in the MIT/CMU data set is 15.8 KB compared to 135.6 KB for original images (a nine times reduction in size). This reduction, however, does not directly apply to a normal video, since video encoders typically use motion estimation between frames to achieve higher compression. More details about the experimental results with streaming video can be found in our previous work [Korshunov and Ooi 2005].

The effect of JPEG compression on accuracy was also tested for Rowley face detection algorithm. Results, presented in Figure 3(b), demonstrate that this algorithm is generally less accurate compared to Viola-Jones algorithm (see Figure 3(a)). Nevertheless, the detection index of Rowley algorithm shows the same pattern of being at its original level until JPEG compression quality is reduced to value 13 (indicated with the dashed vertical line in the figure). Conservatively, the critical compression quality can also be chosen as 20. False positive of Rowley algorithm is lower than Viola-Jones algorithm and it is also not affected by the decrease in compression quality.

Figure 3(a) and Figure 3(b) demonstrate that both face detection algorithms have noticeable fluctuations in the detection index. The main reason for such

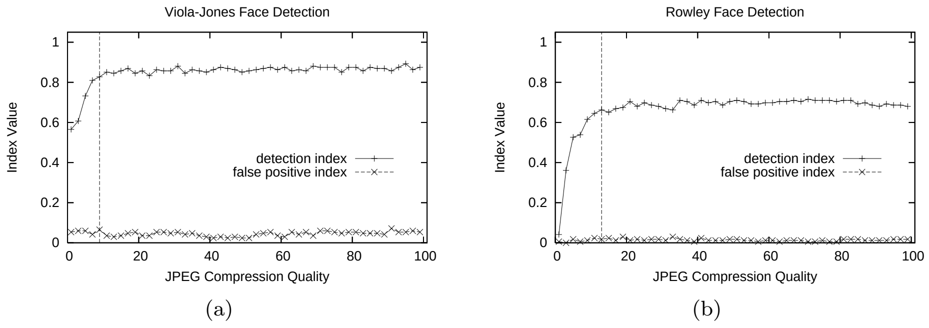


Fig. 3. Accuracy of face detection algorithms vs. JPEG compression quality.

fluctuations in the detection lies in the reliance of the Viola-Jones and Rowley algorithms on different threshold values, which are empirically obtained through offline training of their classifiers. These values affect the detection sensitivity of algorithms to the faces in the input images. Slight changes in the pixel values of an image due to compression can unpredictably affect the decision of the algorithm on faces that are near the threshold. Also, these algorithms are sensitive to factors such as face size, lighting, background conditions, etc. For more details refer to our earlier study [Korshunov and Ooi 2005], where additional experimental results supporting this reasoning are presented.

It is hard to find a definitive and quantitative answer to why these face detection algorithms remain accurate for highly compressed images. Intuitively, algorithm's accuracy depends on what type of features it searches for in an image and how it performs the search. The type of distortions, caused to video/image by reduction in quality, affects algorithms' accuracy as well. For instance, Viola-Jones algorithm is based on Haar-like features, which are not affected much by the blockiness artifact (the strongest artifact of JPEG compression) compared to, say, an edge detection algorithm. The design of the algorithm, however, plays very important role as well. Many modern algorithms (including Viola-Jones and Rowley algorithms) are based on empirical training using a large pool of real-life images with faces of various qualities, shapes, and scales. Therefore, thresholds and pruning values obtained in the training stage have a strong affect on algorithms' accuracy as well as their robustness to reduction in video quality. We have previously discussed in more details the effect of Viola-Jones algorithm's thresholds on its response to JPEG compression [Korshunov and Ooi 2005].

4.2 Face Recognition

The accuracy of QDA-based face recognition algorithm [Lu et al. 2003] is evaluated for the following spatial video adaptations: nearest neighbor and pixel area relation scaling algorithms (see Table II). The results are presented in Figure 4(a) and Figure 4(b) respectively. Similarly to the accuracy pattern of face detection algorithms, the accuracy of face recognition does not change until video quality is reduced to a critical spatial quality. As indicated with dashed vertical lines in the figures, for nearest neighbor scaling algorithm the critical quality is 20% of the

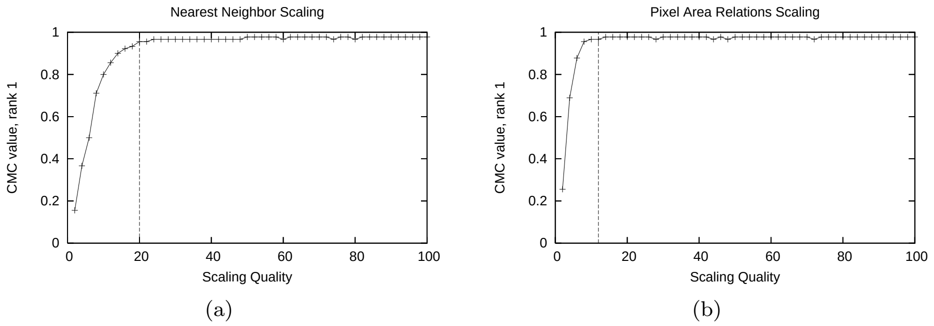


Fig. 4. Identification CMC value of face recognition vs. scaling quality of two scaling algorithms.

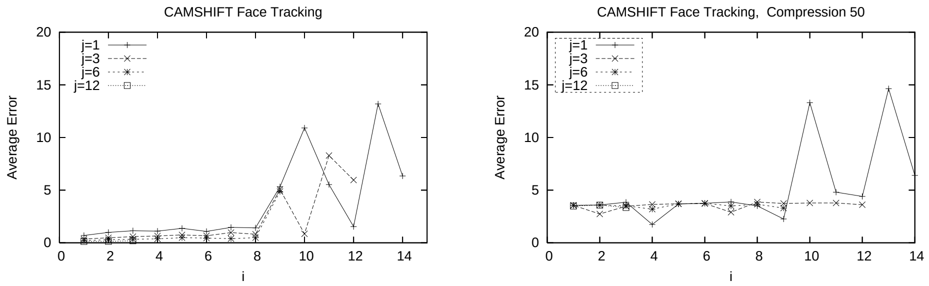
original images sizes and for pixel area relation it is 11%. On average, for images from Yale set, these qualities reduce file sizes to 9.9% of their original sizes (10 times reduction) for the nearest neighbor scaling algorithm and to 4.2% (24 times reduction) for the pixel area relation scaling.

Similarly to face detection algorithms, face recognition remains accurate for images with significantly reduced quality. We use QDA-based recognition, which relies on a quadric offline-trained classifier to determine whether two input faces belong to the same person or not (a categorization task). Detection and recognition algorithms are designed to perform their task accurately on data (images or video) with large amount of noise. Video artifacts of compression and scaling can be regarded as noise added to an image. Therefore, since detection and recognition algorithms are designed to be prone to noise, they are robust on video that is highly compressed or scaled.

4.3 Face Tracking

In this section, we study the trade-off between accuracy of implemented in OpenCV library CAMSHIFT [Bradski 1998] face tracking algorithm and two qualities of the video, temporal and SNR. We run the tracking algorithm on video with different frame dropping patterns and compute tracking average error as described in Section 3. We also test face tracking for different compression qualities.

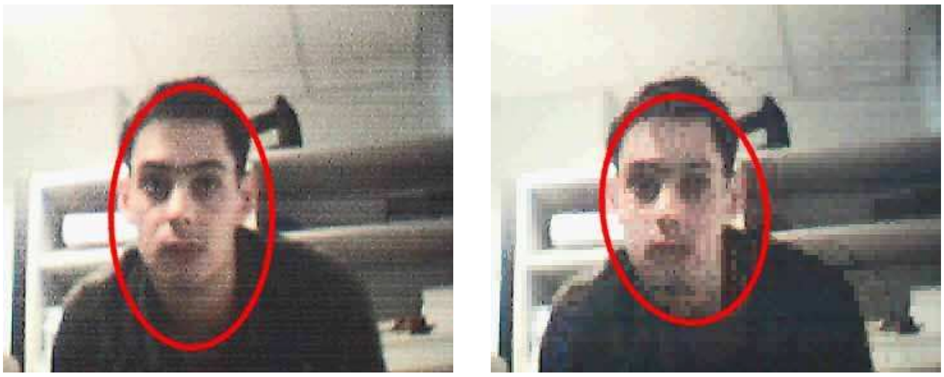
Figure 5(a) shows the average error for one of the test videos for patterns with i varying from 1 to 14 and j equal to 1, 3, 6 and 12. The figure shows that drop gap i plays a more important role in the accuracy of the tracking algorithm compared to j. We can see from the figure that accuracy is consistent with increase of i and decrease of value j. Only when gap i is more than 8, the algorithm shows unpredictable behavior; we call this drop gap a *critical drop gap.* The reason for unpredictable behavior is that CAMSHIFT algorithm searches for a given object's histogram inside a subwindow of the current frame of the video, which is computed as 150% of the object size detected in the previous frame. Therefore, if the object, moves between two frames from its original location for a distance larger than half of its size, the algorithm will lose the track of the object. With another drop gap, the face may be able to move out and move back into the search subwindow. Hence,



(a)

(b)

Fig. 5. Average error vs. drop gap for CAMSHIFT algorithm. Compression quality 100 and 50.



(a)

(b)

Fig. 6. A snapshot frame from a test video for CAMSHIFT face tracking. In (a) it is compressed with quality 100 and in (b) with quality 50.

the oscillations in the algorithm's accuracy occur.

Such observations demonstrate the significance of gap i for the accuracy of the face tracking algorithm. In the video used for Figure 5(a), i should be bounded by 8 for the tracking to be consistently accurate. Therefore, the algorithm can achieve reasonable accuracy (within two pixels) using the pattern: "Drop 8 frames out of 9 frames." In other words, the video source only needs to send at 1/9 of the original frame rate.

Next, we study the effect of SNR quality on the accuracy of face tracking. We compress the video with different compression qualities and repeat the experiments with frame dropping pattern. The results for video with compression quality 50 are shown in Figure 5(b). We can see that accuracy is lower on average for video of higher compression ratio. An increase in compression ratio leads to an increase in average face distance ratio since highly compressed video has fewer details, making the border of a tracked face less distinct. Figure 6(b) shows the effect of the compression with quality 50 using a frame sample from the test video.

The results reported above come from experiments on a single video, captured using a web-cam in a normal office environment. We repeat the experiments for different videos with different content and notice that for a movie clip with talking person, moving his hands occasionally, the critical drop gap is 14, even when the video was compressed with quality 10. On the other hand, for a movie clip showing a character moving his head constantly in a fast and jumpy motion, the critical drop gap is found to be 4 ("drop 4 out of 5 frames" pattern). We also run experiment with web-cam video captured in different lighting conditions. The critical drop gap found for various videos and different compression qualities is plotted in Figure 7. The figure shows that compression quality does not significantly affect accuracy of the face tracking algorithm, hence, the type of face motions is a major video constraint for the accurate tracking. Face tracking algorithm is resistant to video compression because it is based on histogram matching. Since DCT-based compression removes high frequencies from a video, it does not have a significant effect on the histogram of a face.

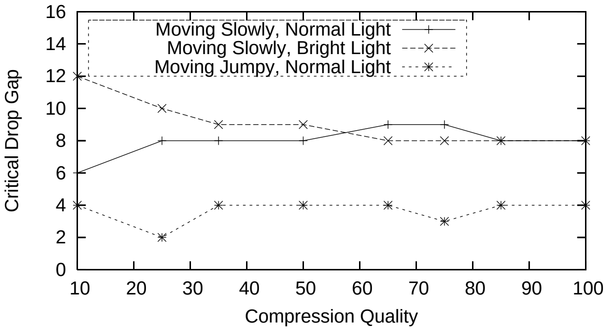


Fig. 7. Critical drop gap vs. compression quality.

5. CRITICAL SNR QUALITY ESTIMATION

We demonstrated that the tested face analysis algorithms can sustain high reductions in video quality. However, in practical scenario, determining critical video quality for a given video analysis algorithm is difficult. In the previous section, to find critical quality, we used an exhaustive search by running an algorithm on videos/images with gradually reduced quality. Such search is inefficient and therefore undesirable.

To overcome the above issues, we propose using metrics specific to computer vision to compare videos degraded by video adaptations with different types of distortion. Such metric can also be used for finding critical video quality for an analysis algorithm, provided the metric is a "perceptual" metric for the algorithm, i.e., it fits the way the algorithm analyses the video. Although several quality metrics exist, such as objective PSNR metric or perceptive VQM and SIMM, they were designed for human visual system and, therefore, cannot be applied directly to video analysis. Video analysis algorithms, unlike humans, have different requirements on the video quality, and hence, the challenge is to design a metric that is suitable for different algorithms.

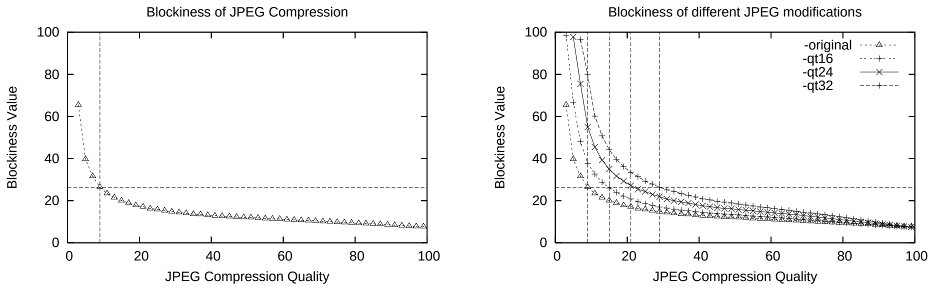
A video quality metric suitable for algorithms needs to satisfy several properties. It should correspond to the tradeoff between video quality and algorithms' accuracy. In fact, since the sweet spot in such tradeoff is the most important point, the metric should exhibit a significant drop in quality around the region where the sweet spot occurs. Different tradeoffs, however, result from using different video adaptations with the algorithm. For instance, Figure 4(a) and Figure 4(b) show two different tradeoffs for two different scaling adaptations and a recognition algorithm. Therefore, good video quality metric should match such different sweet spots accurately. Since different video adaptations degrade video differently, finding a suitable metric is a challenging task. Other properties of a suitable video quality metric include ease of computation and generality to suit different video analysis algorithms. Optionally, if the metric correlates with changing video bitrate, its value can be used to compare different video analysis algorithms on how they perform on low video quality.

In this section, we consider two different metrics that can be used to measure SNR quality of the video: blockiness and mutual information. Blockiness is one of the common distortion types, often called *video artifacts*. We choose blockiness for two reasons: (i) it is the most prominent artifact of JPEG compression, and (ii) Viola-Jones algorithm relies on Haar-like features, which, intuitively, should be affected by this artifact. Blockiness, however, is not suitable for non-blocky image adaptations such as bicubic scaling or JPEG 2000 compression. Therefore, we also consider mutual information, as an alternative metric, which measures general information loss in the degraded image. To demonstrate that the proposed metrics satisfy the first property above, we show that, for a given video analysis algorithm, the same metric's value matches the critical video quality obtained with different video adaptations. Therefore, using this value, we can practically estimate critical quality for different types of adaptations without running multiple empirical experiments demonstrated in the previous section.

In Section 5.1 and Section 5.2, we show how blockiness and mutual information metrics can be used to estimate critical SNR quality for Viola-Jones, Rowely, and QDA-based face recognition algorithms. We use JPEG compression and various scaling algorithms as examples of different adaptations. We demonstrate that blockiness metric can be used with blocky video adaptations, while mutual information does not depend on adaptations type. Adaptation-independence makes mutual information more convenient to use in practice, but it is also shown to be less accurate than blockiness in estimation of SNR video quality.

5.1 Blockiness metric

We demonstrate, in this section, that blockiness can be used as a video quality metric for various blocky video adaptations. To avoid inconsistencies with definition of critical SNR quality, we call the corresponding value of blockiness metric as *threshold* on blockiness. We first find such threshold for a face detection or recognition algorithm and a single video adaptation with blockiness artifact, e.g., JPEG compression. To demonstrate that it can be used as metric, we show that



(a) (b)

Fig. 8. Value of blockiness metric vs. JPEG compression quality for different modifications of JPEG algorithm.

the same threshold value can be used to determine critical SNR quality for other blocky video adaptation as well.

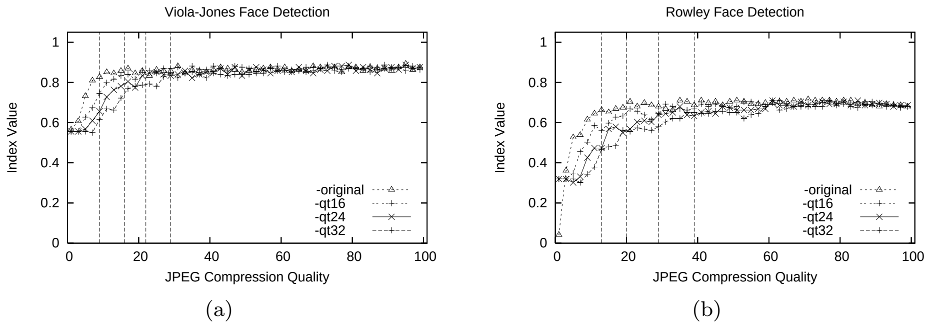
A non-reference blockiness metric by Muijs and Kirenko [Muijs and Kirenko 2005] is adopted in our experiments. We chose this metric because it is easy to implement and easy to adjust for blocks of different size. In a given blocky image, the metric measures the contrast between local gradient of the block's edge and the average gradient of the adjacent pixels. Essentially, the metric's value is the ratio of these gradients. It considers horizontal and vertical block edges separately and takes the average of these values across all the blocks in the image.

We use images from MIT/CMY dataset for face detection algorithms (see Section 3 for more details) with JPEG compression as video adaptation. For recognition algorithm, we use Yale dataset and different scaling algorithms.

Face Detection. Since blockiness is the most prominent video artifact of JPEG compression, it is reasonable to suggest that this artifact would affect accuracy of face detection. We compute blockiness for each compressed image assuming that block artifacts of JPEG have a size of 8×8 pixels. Since we later use blockiness for other video adaptations that have blocks of different sizes, we normalize its original value by multiplying it with the block's size. Using the MIT/CMU dataset, we measure the blockiness for different JPEG compression qualities and plot the results in Figure 8(a).

For Viola-Jones face detection algorithm, taking JPEG compression quality 9 (the sweet spot in Figure 3(a)), we can suggest 26.4 to be the threshold on blockiness (indicated by the dashed line in Figure 8(a)). In order to verify that blockiness is a suitable quality metric for Viola-Jones algorithm, we need other video adaptations with blockiness artifact, and, for these adaptations, the same threshold value should fit the corresponding sweet spots.

We artificially created other blocky video adaptations by modifying JPEG compression. We created three simple quantization tables of JPEG that lead to blockier images than the original JPEG compression. Tables are constructed without any specific reason in mind, except they should be simple and emulate the pattern of the original quantization table. We used formula $a_{ij} = (4+i)(4+j)$, to obtain seven



(a)

(b)

Fig. 9. Accuracy of Viola-Jones and Rowley face detection algorithms vs. JPEG compression quality for different modifications of JPEG algorithm.

rows of the first table with last row and column repeated twice. Multiplying values of this table by 3/2, we obtain the second table and multiplying them by 2, we obtain the third table. We term the corresponding JPEG compressions according to their tables' most top-left values: "qt16", "qt24", and "qt32". JPEG with the original quantization table is marked as "original".

Blockiness values for our JPEG modifications are compared in Figure 8(b). Taking 26.4 as threshold on blockiness determined above, we can estimate that the critical SNR quality for "qt16" should be 15, for "qt24" should be 21, and for "qt32" should be 29 (all values are indicated in the figure with dashed vertical lines). Plotting accuracy of Viola-Jones algorithm against compression qualities of these JPEG modifications in Figure 9(a) demonstrates that the estimated critical SNR qualities match the sweet spots of the corresponding curves very well. Therefore, the same threshold on blockiness determines the critical SNR quality value for Viola-Jones face detection algorithm and several different versions of JPEG compression.

To verify that blockiness as the quality metric is not specific to Viola-Jones algorithm only, we conducted the above experiments for Rowley algorithm. The threshold on blockiness is determined as 21.5 based on the sweet spot value 13 from Figure 3(b) and the blockiness measurements of JPEG in Figure 8(a). Therefore, critical SNR qualities for different modifications of JPEG can be estimated as 20 for "qt16", 29 "qt24", and 39 for "qt32" (from Figure 8(b)). Plotting accuracy of Rowley algorithm against our versions of JPEG compression in Figure 9(b) confirms the estimated values as they fit the corresponding sweet spots.

Rowley face detection algorithm is based on variations in pixel intensities, which are not blocky type of features as Haar-features of Viola-Jones algorithm. Nevertheless, blockiness metric estimates the critical SNR quality for Rowley algorithm well, because we use JPEG compression, for which blockiness is a main video artifact. This observation indicates that the accuracy of face detection is mostly affected by the type of video adaptation's distortion rather than features that algorithm relies on in its detection.

Note that the proposed simple modifications of JPEG are more preferable compared to original JPEG compression. First, the original quantization table is em-

(a)

	Critical quality	Image size (bytes)
original	9	6955
qt16	15	6861
qt24	21	6604
qt32	29	6739

(b)

	Critical quality	Image size (bytes)
original	13	8547
qt16	20	8171
qt24	29	7980
qt32	39	8035

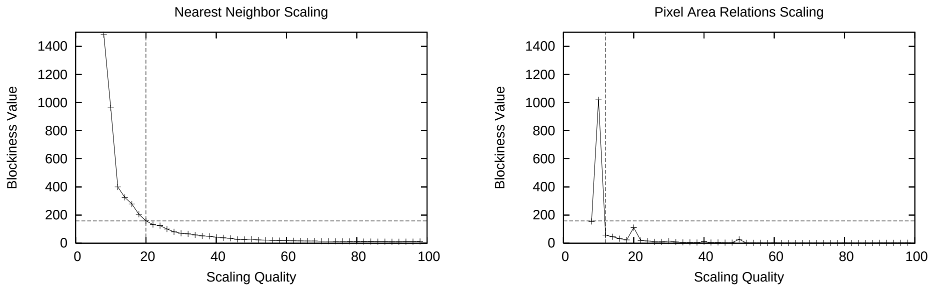
Table III. Critical video qualities and corresponding average images sizes estimated with blockiness metric for Viola-Jones (a) and Rowley (b) algorithms with original and modified JPEG compressions.

pirically determined to fit human visual system, which is not well suited for video analysis algorithms. Second, modified quantization tables can be expressed using formula and hence easier to use in practice compared to storing tables in memory of every device that uses JPEG compression (the current situation). The only concern with simpler modifications of JPEG would be that their critical SNR qualities amount to bigger file size compared to original JPEG. To address this concern, we measured the average size of tested images compressed the critical qualities for Viola-Jones algorithm in Table III(a). From the table, we notice that each critical quality corresponds to images with average size 8% of the images compressed with conventional JPEG quality 90. Hence, our simplified versions of JPEG lead to similar or arguably better bitrate reductions than the original JPEG. Similarly, Table III(b) shows that for Rowley algorithm, critical SNR qualities of original and modified JPEG compressions result in images with comparable average sizes. These findings suggest that simpler and more efficient encoders can be developed for these face detection algorithms.

Face Recognition. For QDA-based face recognition algorithm, we estimate the critical video quality using blockiness metric for nearest neighbor and pixel area relation scaling algorithms. These scaling algorithms exhibit strong blockiness video artifacts. Unlike JPEG compression, however, sizes of resulted blocks depends on the value of scaling quality (the percentage to which images are pre-scaled to). For example, consider downscaling an original image to 50% using nearest neighbor. After scaling back, each pixel in the resulted image is repeated, resulting in the blocks of 2×2 pixels. Therefore, we adopted the blockiness metric used in Section 5.1 to blocks of different size. The blockiness value for nearest neighbor and pixel area relation scaling algorithms are presented in Figure 10(a) and Figure 10(b) respectively. Combining these measurements with results on accuracy of the face recognition algorithm given in Figure 4(a) and Figure 4(b), we can find value 158.5 to be a threshold on blockiness. Note that the same threshold value is obtained for different scaling adaptations. This fact indicates that blockiness can be used as SNR quality metric for QDA face recognition as well.

5.2 Mutual information metric

Video artifact metrics can be used only with video adaptations that produce the measured artifacts. Such restriction causes inconvenience in using artifact metrics



(a)

(b)

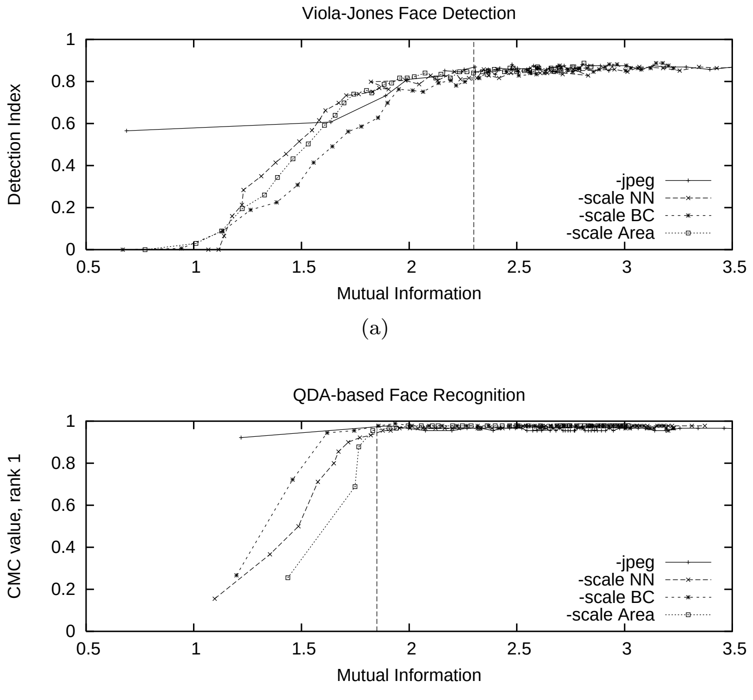
Fig. 10. Blockiness metric vs. scaling quality of two scaling algorithms.

in practice. Therefore, it is desirable to have a video quality metric that is more independent of the way the video is degraded. In this section, we propose mutual information as such a metric and show that it suits face detection and face recognition algorithms.

Mutual information was first introduced in information theory [Shannon 1948] and has proven itself as a good similarity metric in image registration. It measures the amount of statistical information two different images share about each other. It is easy to compute and it is a more general measure of distortion compared to a video artifact metric (such as blockiness), which focuses on a specific type of distortion. Also, mutual information is a better measure of video quality for video analysis algorithms than PSNR. This is because, for instance, mirroring an image to itself, while not affecting the performance of face detection or face recognition, changes its PSNR. Mutual information value, on the other hand, is not affected by such operations.

We demonstrate the advantages of mutual information by measuring the quality of video degraded with different types of video adaptations. In addition to previously used blocky adaptations (JPEG, nearest neighbor, and pixel area relation scaling), we also consider bicubic scaling algorithm, which adds a strong blurriness artifact to the degraded image. We conduct experiments for Viola-Jones face detection and QDA-based face recognition algorithms. Similar to experiments with blockiness, we show that mutual information can be used as a metric of video quality for the selected algorithms. It means that a single threshold value of mutual information can be used to estimate the critical quality for a particular algorithm across various video adaptations.

To compare experimental results on mutual information for different adaptations, we plot the value of mutual information vs. the accuracy of a given video analysis algorithm. The results are presented in Figure 11(a), for face detection and in Figure 11(b), for face recognition. We explain how a single curve on the graph is obtained, using example of JPEG compression (marked as "jpeg") and face detection algorithm. Images from the MIT/CMU test dataset are compressed with JPEG compression qualities varying from 1 to 99. For each JPEG quality, we compute detection index of the face detection algorithm and the average mutual information, using original uncompressed images as references. Note that mutual



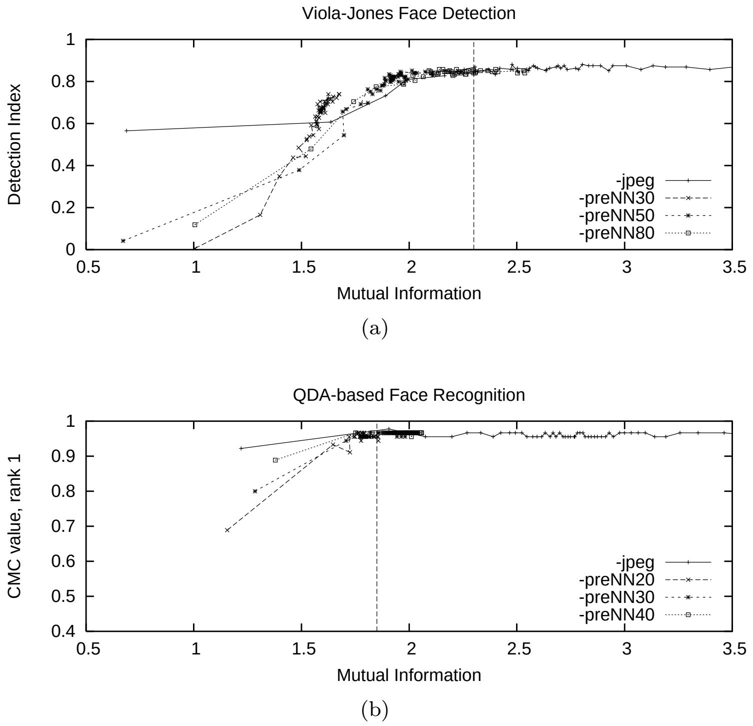
(a)

(b)

Fig. 11. Mutual information vs. accuracy of face detection and face recognition algorithms. Different curves correspond to different types of video adaptations.

information has lower value for more distorted images and higher value for less distorted. The resulted pair of detection index and mutual information represent one point on "jpeg" curve. Curves for scaling algorithms are obtained similarly. Curves marked as "scale NN", "scale BC", and "scale Area" correspond to nearest neighbor, bicubic, and area-based scaling respectively. For face recognition algorithm, Yale dataset is used (partitioned to probe and gallery subsets as described in Section 3), and cumulative match characteristic (CMC) rank one value [Grother et al. 2003] is computed.

Figure 11(a) demonstrates that a mutual information value between 2 to 2.3 can be considered as a threshold corresponding to the critical video quality for the face detection algorithm for the given set of images. The threshold is actually an interval, because the face detection algorithm is not very robust to high noises in images showing frequent fluctuations in accuracy. In practice, we can conservatively use 2.3 to be the threshold for mutual information metric, as indicated with the dashed vertical line in the figure. This value reflects the quality 17 for JPEG compression (which is between sweet spot value 9 and our conservatively selected critical quality 20), 54 for nearest neighbor, 48 for bicubic, and 52 for area scaling algorithms. Degrading images in MIT/CMU dataset to these qualities corresponds to approximately 12, 4, 5, and 6 times reductions in average image sizes.



(a)

(b)

Fig. 12. Mutual information vs. accuracy of face detection and face recognition algorithms. Different curves correspond to different combinations of nearest neighbor scaling and JPEG compression.

For face recognition, the CMC rank one value is plotted against the value of mutual information in Figure 11(b). From the figure, the face recognition threshold value on mutual information can be conservatively set to 1.8. This value gives approximately 10, 11, 21, and 29 times reduction in Yale image sizes for JPEG compression, nearest neighbor, bicubic, and area scaling algorithms respectively.

Since blocky and blurry types of video adaptations were used in these experiments, it demonstrates that, compared to artifact metrics, mutual information is adaptation independent. Therefore, we can use mutual information to measure SNR quality for a combination of different video adaptations. For example, video frames can be scaled down first and then compressed with JPEG to achieve a higher bitrate reduction. We only need to make sure that for the resulted frames, the value of mutual information is above the threshold.

Combining Several Video Adaptations. Figure 1 shows a practical video surveillance scenario, where the surveillance video is reduced by scaling followed by compression. Combination of two adaptations allows even higher reductions in video size compared to using single adaptation (compression or scaling). We use nearest neighbor scaling for its speed. It also shows the worst reduction results compared with other scaling algorithms. As described in Section 3, images from MIT/CMU

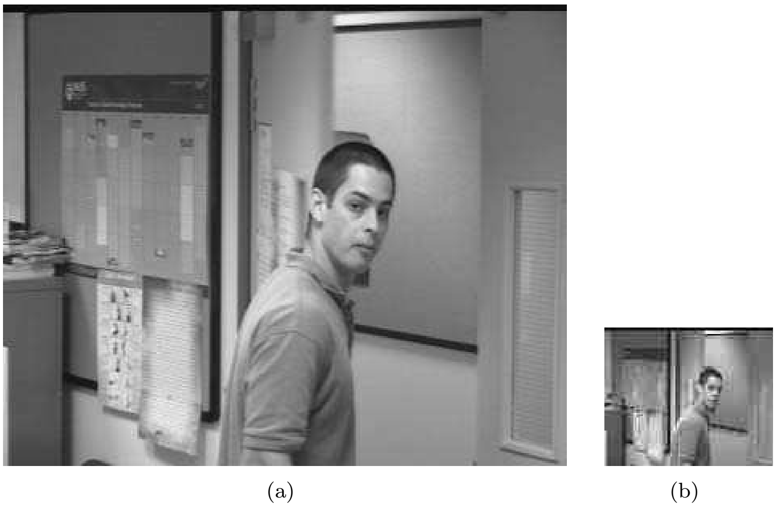


Fig. 13. An example of original video frame (JPEG compression value 90) used in practical tests (a) and an example of test frame scaled with nearest neighbor to 30% followed by JPEG compression with quality 20 (b).

and Yale datasets are degraded following this sequence: prescaled, compressed with JPEG, decompressed, and scaled back to their original resolution. The accuracy of Viola-Jones face detection and QDA-based face recognition algorithms are compared on the degraded and original images. Accuracy vs. mutual information are plotted in Figure 12(a) for face detection and in Figure 12(b) for face recognition. For face detection algorithm, the images were prescaled to 30%, 50%, and 80% of their original resolution, which correspond to curves “preNN30”, “preNN50”, and “preNN80”. Images for face recognition algorithm were prescaled to 20%, 30%, and 40%. The threshold values on mutual information that were found in the previous section are indicated with dashed vertical lines in the corresponding figures.

By looking at Figure 12(a) and measuring the reduction in files sizes for the corresponding transformations, we notice that there is no apparent benefit in combining neighbor scaling and JPEG compression video adaptations for face detection algorithm. Only images prescaled to 80% and compressed with JPEG compression quality higher than 75 have mutual information larger than the threshold. Evidently, the best choice for face detection, because of simplicity and amount of bitrate reduction, is to apply a single JPEG compression with quality 17.

With face recognition, the situation is different (Figure 12(b)). By measuring resulted files sizes, we found that the best reduction in size is achieved by prescaling images to 30 with nearest neighbor and then compressing them with JPEG quality 20.

Lab Experiments. To verify the critical video qualities determined in the previous section for face detection and face recognition algorithm in a practical scenario, we installed a video camera in our research lab and pointed it at the door (see the description of the test video given in Section 3). We degrade the original video frames to JPEG quality 20, as the critical SNR quality for face detection. For face recognition, we prescale the video with nearest neighbor algorithm to 30 percent first, then compress it with 20 JPEG quality. An example of the original camera frame shown in Figure 13(a) can be visually compared with the degraded frame in Figure 13(b). The resulted reductions in bandwidth are presented in Table IV. The reduction in bandwidth amounts to 3.9 times for face detection and 12.5 times for face recognition. If we also reduce original video frame rate from conventional 30 fps to 5 fps, which is a reasonable frame rate for detection and recognition, the reduction amounts to 23 times for face detection algorithm and 75 times for face recognition.

Video	Mutual Information	Bitrate (kbps)	Reduction
Original	-	4403.2	-
Video for FD	2.7158	1138.8	3.9
Video for FR	1.798	352.2	12.5

Table IV. The reduction of video bitrate: original video, degraded video for face detection (FD), and for face recognition (FR) algorithms.

We evaluated both video analysis algorithms with video degraded in the above manner, considering each frame as a separate image. Coordinates of faces detected by face detection algorithm were given as an input to the recognition algorithm. We evaluate the recognition algorithm by using the verification, instead of identification, performance metric [Grother et al. 2003]. The choice of evaluation metric is not essential to us, since we only concern with the consistency in algorithm's performance when the video is changed from the original high quality to the degraded low quality.

Evaluations of two algorithms showed that face detection algorithm correctly detected 144 out of 237 faces in images compressed with both JPEG quality 20 and 90. The algorithm, however, had falsely detected four faces for quality 20 and one face for 90. To avoid occasional false positives occurring due to algorithm's fluctuations, only faces that are present in three consecutive frames were counted as a real face. The detected faces from the degraded video, including false positives, were used as the inputs to the recognition algorithm. Recognition showed two false positives for degraded video (expectedly, false positives from face detection were not recognized) and surprisingly five false positives for the original video. We used only one face per person in the gallery for verification. Adding more faces per person with different expressions might improve the recognition performance.

From our experiments, we can notice that the same type of degradation results in different mutual information values depending on the image types. This is because computation of mutual information requires the reference image. Therefore, in practice, two situations need to be considered: (i) finding the threshold on mutual information for the given video and (ii) checking if mutual information for current live frames exceeds the threshold. Since the original and degraded video frames are required for computing mutual information, during the normal operation of the system, its value should be computed at the video source for each frame.

The threshold value on mutual information can be found interactively during the calibration stage of the system, by incrementally decreasing the video quality and evaluating the performance of video analysis algorithms. Another way is to build a table of typical thresholds values for different categories of images offline and use corresponding values in particular live scenarios.

Experiments with artifact and mutual information metrics demonstrate that once the corresponding threshold is found for a face detection or recognition algorithm, it can be used to determine critical SNR qualities for different video adaptations, e.g., JPEG compression or nearest neighbor scaling. To understand which metric to use and what the metric's threshold is, we reason about a video analysis algorithm (understand what video features it relies upon) and a video adaptation (determine how it degrades the video). Limited empirical experiments, however, are still required for finding metric's threshold for SNR quality.

6. CONCLUSION

In this paper, we evaluated the effect of video quality degradation on several typical examples of face analysis algorithms. The surprising finding of the paper is that the tested algorithms show very high tolerance towards large reductions in video quality. Our experiments demonstrated that the accuracy of the algorithms show no significant decrease until the video is degrade to a certain quality threshold, which amounts to at least 10 times lesser video bitrate than video conventionally encoded for human vision. We also argued that an algorithm-oriented video quality metrics need to be developed. Metrics based on video artifacts, such as blockiness, and mutual information were considered.

Due to heterogeneous and empirical nature of common video analysis algorithms, our results cannot be generalized for all different algorithms. However, we believe that non-trivial and useful video analysis algorithms can be grouped into a limited number of categories that show similar responses in terms of accuracy to various reductions in video quality. Often, algorithms either rely on empirical data or are training based; hence, it is difficult to fully formalize their behavior. Therefore, the idea that video analysis algorithms require lesser video quality than humans needs to be supported with experiments using more examples of algorithms. Changes in algorithms' accuracy need to be studied for major video adaptations used in practical systems, such as in MPEG-4 and H.264.

Overall, the results of the paper strongly suggest that it is impractical and inefficient to treat video analysis algorithms in the same manner as a human video observer. Resource-efficient video analysis algorithms can, and should, be designed. Video encoding algorithms designed for computer vision need to be developed, since, in terms of video quality required, computer vision is very different from human vision.

REFERENCES

Bradski, G. R. 1998. Computer vision face tracking as a component of a perceptual user interface. In *Proceedings of the Forth IEEE Workshop on Applications of Computer Vision, WACV'98*. Princeton, NJ, 214–219.

Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., and Hasegawa, O. 2000. A system for video surveillance and monitoring. Tech. Rep. CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University. May.

Eisert, P. and Girod, B. 1998. Model-based coding of facial image sequences at varying illumination conditions. In *IMDSP Workshop*. Alpbach, Austria, 199–122.

Eisert, P., Wiegand, T., and Girod, B. 2000. Model-aided coding: a new approach to incorporate facial animationinto motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology 10*, 344–358.

Eleftheriadis, A. and Anastassiou, D. 1995. Constrained and general dynamic rate shaping of compressed digital video. In *Proceedings of the IEEE International Conference on Image Processing, ICIP'95*. Washington, DC, USA, 396–399.

Girgensohn, A., Kimber, D., Vaughan, J., Yang, T., Shipman, F., Turner, T., Rieffel, E., Wilcox, L., Chen, F., and Dunnigan, T. 2007. DOTS: support for effective video surveillance. In *Proceedings of the 15th ACM International Conference on Multimedia, ACMMM'07*. Augsburg, Germany, 423–432.

Grother, P. J., Micheals, R. J., and Phillips, P. 2003. Face recognition vendor test 2002 performance metrics. In *Proceedings of the 4th International Conference on Audio Visual Based Person Authentication, AVBPA'03*. Guildford, UK, 937–945.

Hakeem, A., Shafique, K., and Shah, M. 2005. An object-based video coding framework for video sequences obtained from static cameras. In *Proceedings of the 13th ACM International Conference on Multimedia, ACMMM'05*. Singapore, 608–617.

Javed, O., Rasheed, Z., Alatas, O., and Shah, M. 2003. KNIGHT^M: A real-time surveillance system for multiple overlapping and non-overlapping cameras. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'03*. Baltimore, Maryland, 649–652.

Kim, J., Wang, Y., and Chang, S. 2003. Content-adaptive utility-based video adaptation. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'03*. Vol. 3. Baltimore, Maryland, 281–284.

Kim, M. and Altunbasak, Y. 2001. Optimal dynamic rate shaping for compressed video streaming. In *Proceedings of the International Conference on Networking, ICN'01*. Colmar, France, 786–794.

Korshunov, P. and Ooi, W. T. 2005. Critical video quality for distributed automated video surveillance. In *Proceedings of the 13th ACM International Conference on Multimedia, ACMMM'05*. Singapore, 151–160.

Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. 2003. Regularized discriminant analysis for the small sample size problem in face recognition. *Pattern Recognition Letters 24*, 3079–3087.

Muijs, R. and Kirenko, I. 2005. A no-reference blocking artifact. measure for adaptive video processing. In *Proceedings of the 13th European Singal Processing Conference, EUSIPCO'05*. Antalya, Turkey.

Nair, V. and Clark, J. J. 2002. Automated visual surveillance using hidden markov models. In *Proceedings of the 15th International Conference on Vision Interface, VI'02*. Calgary, 88–92.

Rangaswami, R., Dimitrijevi, Z., Kakligian, K., Chang, E., and Wang, Y. 2004. The SfinX video surveillance system. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'04*. Taipei, Taiwan.

Rouse, P. and Hemami, S. 2008. Analyzing the role of visual structure in the recognition of natural image content with multi-scale ssim. In *Proceedings of SPIE Human Vision and Electronic Imaging XIII Conference*. Vol. 6806. San Jose, CA, USA, 680615.1–680615.14.

Rowley, H., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 23–38.

Sanchez, V., Basu, A., and Mandal, M. 2004. Prioritized region of interest coding in JPEG2000. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR'04*. Vol. 2. Melbourne, Australia, 799–802.

Schumeyer, R., Heredia, E. A., and Barner, K. E. 1997. Region of interest priority coding for sign language videoconferencing. In *Proceedings of the First IEEE Workshop on Multimedia Signal Processing, MMSP'05*. Princeton, NJ, 531–536.

Shannon, C. 1948. A mathematical theory of communication. *Bell System Technical Journal 27*, 379–423.

Smolic, A., Makai, B., and Sikora, T. 1999. Real-time estimation of long-term 3-d motion parameters for snhcface animation and model-based coding applications. *IEEE Transactions on Circuits and Systems for Video Technology 2*, 255–263.

Viola, P. and Jones, M. 2001. Robust real-time face detection. In *Proceedings of the ICCV 2001 Workshop on Statistical and Computation Theories of Vision, ICCV'01*. Vol. 2. Vancouver, Canada, 747.

Wu, Y., Jiao, L., Wu, G., Chang, E., and Wang, Y. 2003. Invariant feature extraction and biased statistical inference for video surveillance. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS'03*. Miami, FL, 284–289.

Yuan, X., Sun, Z., Varol, Y., and Bebis, G. 2003. A distributed visual surveillance system. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS'03*. Miami, FL, 199–204.

Received Month Year; revised Month Year; accepted Month Year