
TV and Video Processing

Scott Janus

Contents

Introduction	472
Scan Conversion	473
Film Mode Detection	476
Frame Rate Conversion	477
Aspect Ratio Conversion	478
Basic Adjustments	481
Brightness and Contrast	481
Saturation and Hue	482
Advanced Adjustments	482
Adaptive Brightness and Contrast	483
Video Artifact Removal	484
Advanced Color Processing	485
Conclusion	486
Directions of Future Research	486
Further Reading	486

Abstract

This chapter describes video processing algorithms commonly in use today. In addition to describing how they work at a generic level, this chapter will also explain why such processing is necessary, even on today's high definition content. Sample pictures are included that visually demonstrate the key principles of the various algorithms. The processing categories that are covered include: deinterlacing, film mode detection, scaling, anamorphic scaling, nonlinear anamorphic scaling, hue, saturation, brightness, contrast, de-noise, detail enhancement, and frame rate conversion.

S. Janus (✉)

Visual and Parallel Computing Group, Intel Corporation, Folsom, CA, USA

e-mail: scott.janus@intel.com

List of Abbreviations

CCD	Charge-coupled device
FPS	Frames per second
i	Interlaced, as in 1080i, referring to 1920×1080 interlaced content
p	Progressive, as in 720p, referring to 1280×720 progressive content

Introduction

The quality of video in the past few decades has improved remarkably, to the point where watching a nostalgic show from your childhood can be a surprisingly low-fidelity experience. Some of the quality improvements have come from an increase in resolution; standard definition broadcasts are slated to cease in 2009 in the USA, and DVD-Video is slowly but surely being supplanted by Blu-ray. Part of the increase in quality comes from new video compression codecs: H.264 and VC-1 offer substantial improvements in efficiency over MPEG-2 (as discussed in chapter “► [Video Compression](#)”).

Yet despite these advances in the baseline video itself, many of the quality enhancements have come from the application of algorithms applied to the decoded, uncompressed video. These techniques are sometimes referred to as post-processing algorithms, in that they are applied after decoding.

Few enhancements were needed in the early days of video, because the content almost always precisely matched the available displays. The format of the live broadcasts was guaranteed to match the characteristics of all the televisions capable of receiving the transmissions. Today, however, the world is a much different place. Even constrained to the traditional consumer electronics (CE) ecosystem in the early years of the twenty-first century, there was still a wide range of different video formats with different resolutions, scan types, and aspect ratios. PCs make the situation more complex, in that the content can be practically any size and displayed via an arbitrary-sized window. PC resolutions and refresh rates are much more varied than that supported by traditional CE interfaces.

The much discussed convergence of PCs and dedicated CE video systems has steadily – yet asymptotically – progressed. Many engineers spent the previous decade working to ensure that CE video could be displayed well on PCs. Today, many engineers are working to ensure that PC/Internet video can be displayed well on CE devices. At the same time, video on mobile devices such as cell phones and mobile Internet devices (MIDs), as well as console game stations, further adds to the combinatorial complexity of the video ecosystem.

This chapter will review many of the enhancement algorithms in place today. For many of the algorithms, there is no universal implementation, and, indeed, there is a great deal of intellectual property and company differentiation tied up in the nuances of a particular implementation. However, the basic concepts are discussed in a generalized form herein.

These algorithms can be classified into three main categories. The first category contains those algorithms that are needed to address a fundamental difference

between the video content and the display device. These differing parameters include the scanning type, aspect ratio, resolution, and frame rate. Without some form of conversion between these mismatched parameters, the video will basically be unwatchable.

The second group consists of those algorithms needed to ensure the presentation of the video on a particular display in a particular environment accurately reproduces a visual experience consistent with a reference implementation. These adjustments ensure that all the dark and bright areas are correctly visible and the colors are properly represented.

The final category consists of those algorithms that strive to somehow improve upon the baseline implementation or otherwise differentiate a given product from its competitors.

Scan Conversion

There are two fundamentally different scan types: interlaced and progressive. Progressive scanning records every line of scene for a particular instance of time. 1080p Blu-ray content and 720p HDTV broadcasts are examples of progressive content. Film content is also inherently frame-based. Being an analog physical medium, there is no real concept of rows or pixels. Still, the entire scene is captured at full spatial resolution with every opening and closing of the shutter. Progressive scanning is illustrated in Fig. 1.

Interlaced scanning records or displays every other line of the scene for a given time stamp. The half frame is known as a field. The fields corresponding to the even-numbered lines of the scene are unsurprisingly known as even fields, and the odd-numbered lines compose the odd fields. Legacy analog television and HDTV's 1080i format are examples of interlaced scanning. Interlaced scanning has historically been used to address technological bandwidth constraints of the time. It is illustrated in Fig. 2.

At first glance, it may seem as though having every other line of the video missing would result in noticeably visible results. After all, half the content is missing! However, when interlaced content is displayed on an interlaced monitor (which includes almost every standard definition television made in the twentieth century), the human psychovisual system's perception of the TV's phosphor decay results in an acceptable presentation of the content. The missing lines are not obvious.

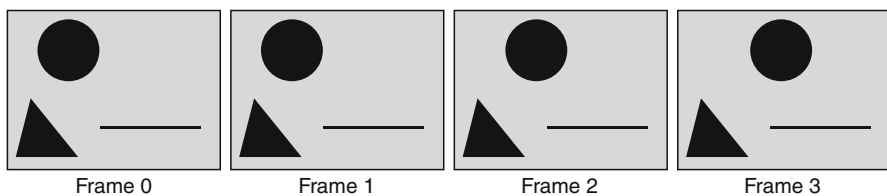


Fig. 1 Progressive scanning

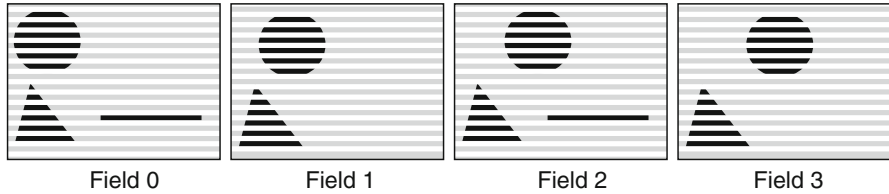


Fig. 2 Interlaced scanning

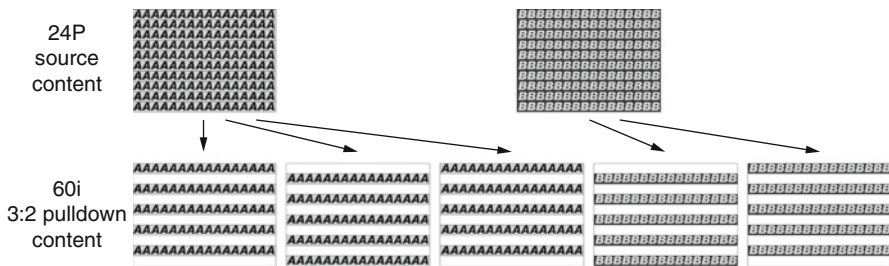


Fig. 3 3:2 Pulldown

The fact that there are two fundamentally different scanning techniques means that some type of conversion must take place if one type of content is displayed on a different type of display. This problem cropped up in the early days of television as engineers considered how to transmit progressive film content over the interlaced-only television broadcast system.

In geographies using a 60 Hz interlaced scan rate, the 24 FPS film content is converted using a technique known as 3:2 pulldown. The 3:2 comes from the fact that the first frame is sampled three times and the second frame is sampled two times to create fields at the necessary cadence. This procedure is shown in Fig. 3. A similar technique known as 2:2 pulldown is used to convert 24 FPS content to 50 fields per second for other geographies. *Telecine* (Matchell 1982) is the generic term for converting film content to interlaced video.

With the advent of progressive displays – most notably computer monitors – in the closing years of the twentieth century, the problem of deinterlacing content became widespread. Merely natively displaying the interlaced content is inadequate, as the human eye can easily see that half of the data is missing. The missing rows of data must be filled in with something. It is impossible to exactly recreate the missing fields, as that data was never captured. Instead, we must approximate the missing information.

One simple approach is to estimate the missing field by interpolating from the pixel directly above and below the absent pixel (De Haan and Bellers 1998). This approach is shown as bobbing and is illustrated in Fig. 4. It is easy to implement but has noticeable artifacts around high contrast, static images such as subtitles. Such

Fig. 4 Bobbing

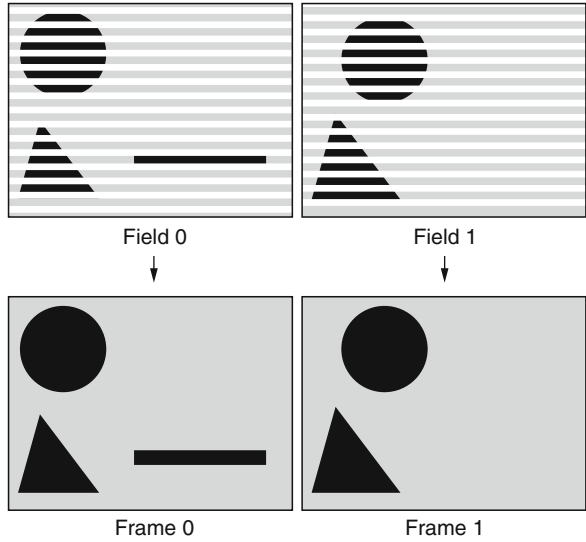
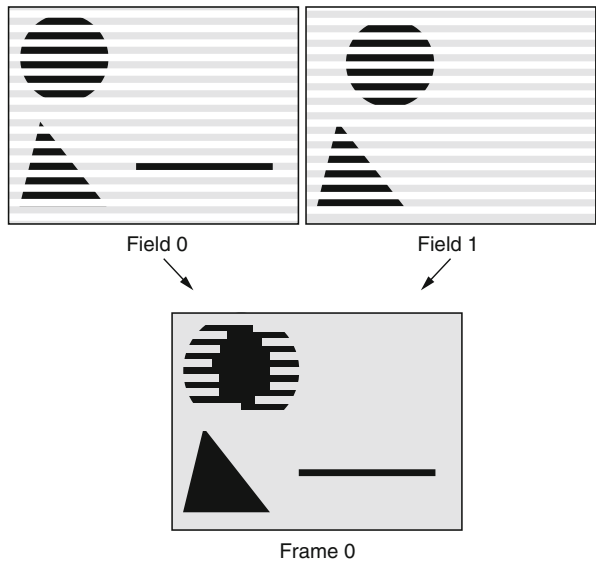


Fig. 5 Weaving



regions tend to flicker, due in part to the fact that the odd and even fields are sampled at different spatial locations.

Another simple deinterlacing algorithm is to simply combine two adjacent odd and even fields into a single frame, as shown in Fig. 5. Weaving, as it is known, works great with static images. However, regions of large movement result in an

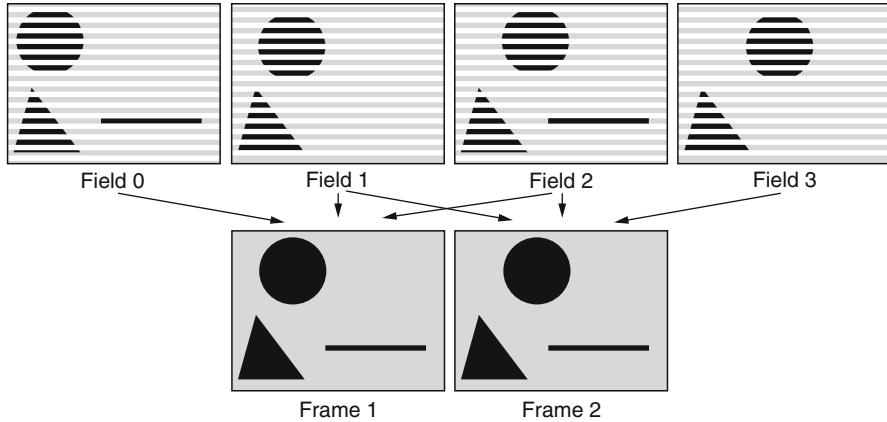


Fig. 6 Advanced deinterlacing

artifact known as combing or feathering. In these regions distinctive horizontal artifacts can be seen due to the two fields—which are samples of two different instances in time—being presented at the same time. Note that whereas bobbing converted 60 fields per second into 60 frames per second, weaving creates 30 frames per second.

A more sophisticated approach is to apply bobbing to regions of the screen with high motion and weaving to regions of the screen with little or no movement. This adaptive deinterlacing can be quite effective, as it discriminately applies bob or weave to minimize the artifacts of each algorithm, as demonstrated in Fig. 6. The size of the regions varies based on the implementation; they can be as small as a pixel, in which case the algorithm is known as pixel-adaptive deinterlacing. Generally speaking, the smaller the region, the more expensive it is to implement the algorithm. Similarly, expense also increases with the range of motion the algorithm can detect.

An even more advanced approach is to use the motion data of adjacent fields to reconstruct a completely new frame. The position of moving objects in the missing field is calculated from the references and placed appropriately. This motion-compensated approach is quite complex to implement.

Film Mode Detection

It is possible that the interlaced content harbors progressive content, such as a telecined movie. Applying the aforementioned deinterlacing algorithms to such content will result in suboptimal results, an example of which is shown in Fig. 7. The sample illustrates the problem when applying weaving, but the problem applies to all the algorithms.

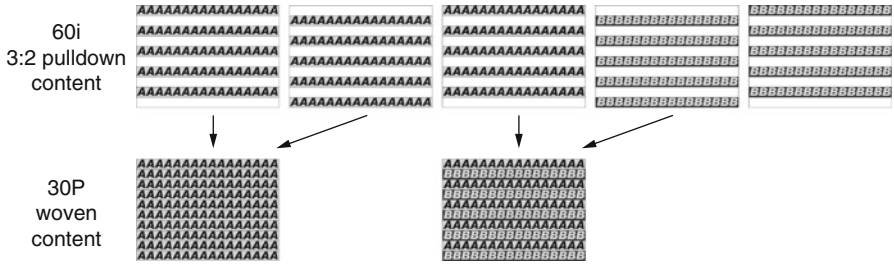


Fig. 7 Incorrect deinterlacing

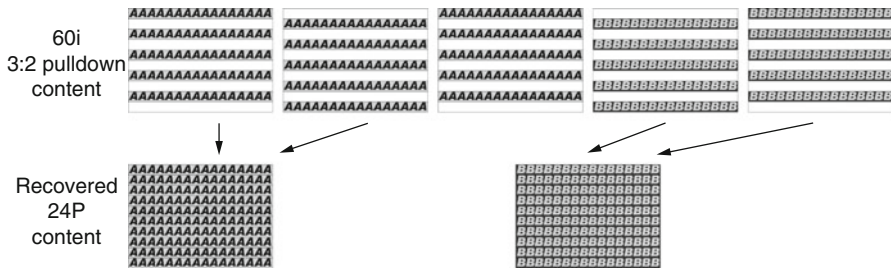


Fig. 8 Correctly recovering progressive content

The best possible result is to monitor the content and compare adjacent fields to see if they originate from the same progressive frame. If done properly, the original progressive frames can be completely recovered from the interlaced content and optimally displayed, as shown in Fig. 8. This process is known as film mode detection (and correction) or inverse telecine.

Frame Rate Conversion

It is possible for the frame rate of the video content to be different than the frame rate of the display device that is showing the content. For instance, a Blu-ray movie may be recorded at 24 FPS and viewed on a PC that has an 85 Hz refresh rate active. Obviously, some sort of conversion between the frame rates must occur in order for the video to be viewed in an acceptable fashion. The simplest approach is simply to replicate the frames, as shown in Fig. 9. This approach provides reasonable quality, although an alert eye can detect the fact that each unique film frame is being presented for a different duration of time.

The variable duration is due to the fact that the display refresh rate may not be an even multiple of the content's frame rate. In the case of 24 FPS content and a 60 Hz display, the video frames are alternately presented for three and two display refreshes. This cadence results in each video frame being presented for an average

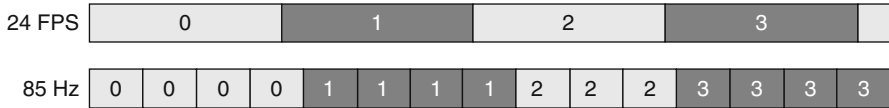


Fig. 9 Frame replication

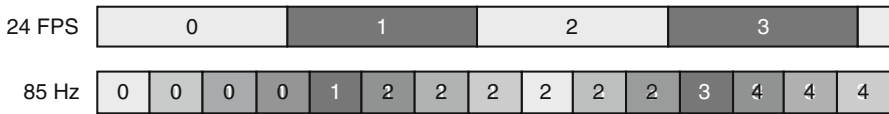


Fig. 10 Frame interpolation

of $60/24 = 2.5$ video frames per display refresh. In the cast of 24 FPS content and an 85 Hz display, the cadence is quite complex, as $85/24$ is an irrational number.

A more sophisticated approach is to not merely replicate frames but to create new intermediate frames. This is a complex process; merely performing a linear interpolation of the intermediate frames will only result in blurred content, as conceptualized in Fig. 10. Instead, some motion-compensated approach must be used. This can be done on a global (picture-wide) basis or at a finer grain.

Typically, these more advanced frame rate conversion algorithms are used in systems where the content and display ratios are a multiple of one another. For instance, many of the newer HDTVs take in a 24 FPS signal from Blu-ray movies and internally convert it to the display’s native 120 Hz refresh rate. The increased number of frames can decrease judder artifacts inherent in the cinematic content’s low frame rate and help mitigate motion blur issues experienced by some display technologies.

Aspect Ratio Conversion

Often, the aspect ratio of the content differs from the aspect ratio of the display. Most retail video has an aspect ratio of 4:3 or 16:9. Older TVs have a 4:3 aspect ratio; most newer ones have a 16:9 ratio. Playing 4:3 content on a 16:9 display or 16:9 content on a 4:3 display requires some form of conversion. The aspect ratio topic is further complicated by the wide availability of Internet video that can have practically any aspect ratio and the fact that video is often played on PCs in a window of arbitrary size. Nevertheless, the techniques used below can be used to span any mismatch between content and display aspect ratios. Although I will use the term display aspect ratio in the following entries to refer to the properties of a monitor’s entire screen, the concepts can be generalized to smaller target regions of video, such as might be seen in picture-in-picture scenarios or when watching windowed video on a PC.

One simple approach is to simply stretch the video to completely fill the screen, as demonstrated in Fig. 11. This approach is popular with many consumers, as the



Fig. 11 Full screen stretching



Fig. 12 Cropping



Fig. 13 Letterboxing

entire screen is active. Video purists decry the technique, however, because the aspect ratio of the content is distorted. For instance, circles are deformed into ovals.

Another approach is to cut out a section of the content that matches the video’s aspect ratio. This cropping technique (Fig. 12) makes use of the display’s entire real estate while preserving the content’s aspect ratio. However, portions of the video are lost. This artifact is most notable in scenes containing two people facing at each other at the extreme edges of the screen. Cropping can remove both of the actors from the viewable picture.

The cropping of key elements can be mitigated to some extent by moving the cropping window back and forth across the content to the areas of interest. This technique is known as pan and scan. Note that pan and scan is only practical with human control. It is only suitable for a prior processing, such as preparing wide-screen content to be broadcast in a 4:3 format.

Letterboxing is a technique that displays the entire scene on the screen while preserving the original aspect ratio. This feat is accomplished by uniformly scaling the content to match the most constraining dimension and filling the remainder of the screen with black, as illustrated in Fig. 13. This approach is the preferred one for videophiles. Some people do not like it because it leaves regions of the screen



Fig. 14 Wide-screen cropping

unused. Letterboxing refers to insertion of horizontal black bars above and below the content when adapting content to a display with a wider aspect ratio. When adapting content to a narrower aspect ratio, black columns are inserted on either side; this technique is known as pillarboxing.

People are sometimes surprised to find letterboxes appearing on their wide-screen TVs when watching movies. This scenario occurs because cinematic aspect ratios are completely disconnected from video aspect ratios. The most common film aspect ratios (such as 1.85:1 and 2.35:1) are wider than 16:9, so letterboxing is still necessary.

A special case of cropping can be used when dealing with letterboxed content. If wide-screen content is letterboxed into a narrow-screen transmission or storage format and subsequently displayed on a wide-screen TV, the cropping window can be set to the display aspect ratio. The resulting image completely fills the screen while preserving the aspect ratio (Fig. 14).

In all the aspect ratio conversion schemes discussed so far, some scaling or resampling of the content is typically required. All of the scaling for the above techniques uses linear scaling, by which, I mean that the scale factor is consistent across the picture. Stretching content to fill the screen uses anamorphic scaling: the scale factor in the horizontal direction is different than the scale factor in the vertical domain. Still, it is linear in that the scale factor remains constant in each direction.

Anamorphic scaling is common in the cinema. It is also used in DVD-Video playback, because the video is sampled using non-square pixels. Typically, DVD-Video content is stored as 720×480 samples for both 4:3 and 16:9 contents. For each aspect ratio, a different anamorphic scaling must be used to properly scale the 720×480 content to the square pixels of today's displays.

Scaling always results in a degradation of the image quality. The topic is worthy of a book itself. For this brief overview, let us just note that unnecessary scaling should be avoided. Competitive solutions today use sophisticated algorithms involving many source samples (or taps) per single destination sample. Simple bilinear scaling is usually inadequate. The most sophisticated algorithms use temporally adjacent frames to increase the effective resolution of the current frame.

This brief segue on scaling leads us to the final aspect ratio conversion technique. It has many different brand names but no common generic name. I refer to it as nonlinear anamorphic scaling. In this technique, the horizontal scale factor is constant, but the vertical scale factor varies across the picture ([Intel Corporation Intel Clear Video Technology](#)). This allows the content aspect ratio to be correctly



Fig. 15 Nonlinear anamorphic scaling

reproduced in the center of the picture at the expense of exaggerated distortions at the edges of the screen, as seen in Fig. 15. This approach utilizes the entire real estate of the screen. As such, it is seen by some as an optimized compromise between cropping and stretching. However, it has very noticeable artifacts. For instance, a car driving across the frame will start with wildly distorted tires that morph into proper circles as it reaches the center region and then stretch back out as it reaches the far edge of the screen.

Basic Adjustments

Even in the early days of a primarily homogenous video environment, some basic video processing controls were necessary to account for variations in products from different manufacturers and to adjust for ambient lighting conditions. These adjustments were necessary to get a perceptually uniform behavior across the different products and installations. In other words, by properly calibrating the TV using these basic adjustments, a person would ideally perceive the same image regardless of the living room or showroom he happened to be in. Of course, in practice, this precise level of calibration rarely happens, as the average consumer does not understand how to properly adjust all the settings to match the reference behavior.

Brightness and Contrast

Brightness and contrast are two commonly used but commonly misunderstood controls (Keith 2007). Both of these operate solely on luma components. Brightness is more technically known as black level adjustment. Contrast refers to the gain of a picture.

Adjusting the brightness of a picture adjusts the entire range of output values. In other words, decreasing the brightness will lower both the black level and the white level. Increasing the brightness will increase both the black and white levels. An example of brightness adjustment is shown in Fig. 16.

Contrast, however, actually alters the distance between the black and white levels. Just manipulating the contrast itself will not alter the black level but will raise and lower the white level. All intermediate values will be stretched accordingly. An example is shown in Fig. 17.



Fig. 16 Adjusting brightness



Fig. 17 Adjusting contrast

Loosely speaking, the function of brightness and contrast can be calculated by the following function:

$$\text{Output_value} = (\text{contrast} * \text{input_value}) + \text{brightness}$$

Saturation and Hue

Hue and saturation manipulate the color. They operate solely on the chroma components and have no impact on luma.

Saturation refers to the intensity of the chroma components. Saturation is accomplished by multiplying the chroma values by a constant. Saturation adjustment examples are shown in Fig. 18.

Hue adjustment basically rotates and shifts all the colors. It can be thought of as a rotation of the hue ring, as demonstrated by examples in Fig. 19.

Advanced Adjustments

For quite some time now, television and display manufacturers have been adding features to their products that are designed to make the video appear better than a reference display. In this context, better is subjective. Basically, the intent is to differentiate from competing products. If all displays exactly matched the reference



Fig. 18 Adjusting saturation

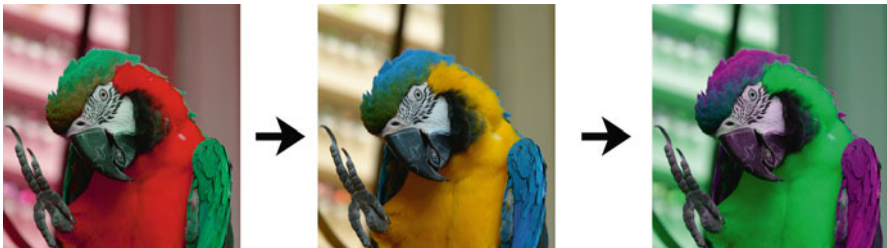


Fig. 19 Adjusting hue

display, then consumers presumably would just buy the cheapest unit. However, if one display presents the video in a fashion that somehow looks more pleasing to the eye than the others, then the consumer might be willing to pay more money for it.

Even when dealing with the latest high definition video formats, there is still a lot of room for improvement. As discussed in chapter “► [Video Compression](#),” consumer video is compressed in a lossy fashion. Most of the content is stored as 8 bit per sample 4:2:0, which has limited dynamic range and color fidelity. There is a distinguishable gap between this content and the limits of human perception. Work is under way to narrow this gap by increasing the bit depth and color gamut of the compressed video, but in the meantime, advanced video enhancement algorithms strive to improve today’s content

Adaptive Brightness and Contrast

One such class of algorithm is adaptive contrast enhancement. This feature generates a histogram of the image and automatically adjusts the contrast to provide an ideal presentation for the current scene. Some hysteresis is needed to ensure the contrast does not change widely with every single frame, and care must be taken to make sure that intentionally dark scenes remain dark. Advanced versions of the algorithm can adaptively adjust the contrast in various regions of the screen.

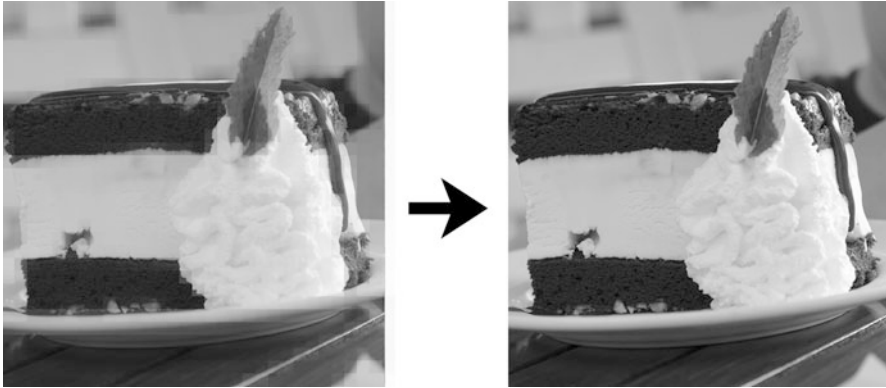


Fig. 20 Deblocking

Another technique is to monitor the ambient lighting conditions and adjust the brightness (and possibly contrast) of the image to match the characteristics of the human perceptual system. For instance, in a dark room, the brightness should be lower than when the same content is being viewed in a bright room. This approach does not necessarily monitor the video content but instead modulates parameters based on external conditions.

Video Artifact Removal

As previously noted, video compression is lossy and introduces artifacts. These artifacts can be mitigated by applying clean-up algorithms to the decoded images.

Deblocking attempts to smooth the boundaries between block-shaped regions with different average values. The human eye is surprisingly good at detecting these regions. Newer video codecs ([ISO IEC 14496 10](#); [SMPTE 421M VC 1](#)) include deblocking filters as part of the decoding algorithm, but not all video streams encoded in these formats take full advantage of the deblocking capabilities. Also, older codecs such as MPEG-2 do not inherently support deblocking, so applying it as a post-processing stage can noticeably improve the picture quality. An example of deblocking is shown in [Fig. 20](#).

High-frequency information is often discarded as part of the compression process. Detail or sharpness algorithms can reproduce some of these characteristics, as demonstrated in [Fig. 21](#). Poorly implemented sharpness filters can create new artifacts, such as creating ringing around high-frequency details. Good filters recreate original detail without introducing new problems.

High-frequency noise is always a problem with analog content and even with most digital content. The CCDs that actually digitize the real world introduce noise, as do the compression algorithms. Noise removal algorithms can remove these artifacts, as seen in [Fig. 22](#), and create a cleaner picture. Care must be taken to



Fig. 21 Sharpening



Fig. 22 Noise reduction

only remove the unwanted noise; just applying a general blurring filter will mitigate the noise but will also lose fine details that are part of the desired picture.

Advanced Color Processing

There are many different techniques in contemporary products that strive to improve the color response of the system. These basically strive to create more intense or vibrant colors than the reference implementation. Some of these algorithms try to boost colors that have limited bandwidth in a particular domain. Some take advantage of the broader range of colors that the new wide gamut displays are capable of displaying. Often, these color enhancements are not technically correct, in that they produce colors that are different than a defined spec, such as BT.709. However, many viewers prefer the enhanced colors to the baseline implementation.

One interesting class of algorithms is skin tone detection. These algorithms attempt to detect regions of the screen that contain samples of human skin and adjust the color of the skin independent of the rest of the picture. These can arguably make the depicted people look better. Again, this is straying from an accurate reproduction of the original scene. Presumably, a properly calibrated system would accurately depict any flesh tones just as accurately as the rest of the scene. There are a wide range of different flesh tones out there in the real world, so it is not always clear if there is a universal advantage to this type of processing.

Conclusion

We are in the middle of an amazing period of video quality improvement. Even as the transition from standard definition to high definition continues, a widespread proliferation of high-quality HDTVs and high-resolution PC monitors has sparked an arms race of competitive video enhancement features. As long as video storage techniques remain restricted to reproducing a significant subset of what humans can perceive – which seems likely to be the case for decades to come – there will always be a market to enhance that video to create a perceptually improved experience. It is also worth noting the recent influx of relatively low bit-rate content, such as Internet-streamed video content à la YouTube, is a particularly fertile ground for new video processing techniques to emerge.

Directions of Future Research

The field of video processing research remains active, with new capabilities being introduced with the annual release of new televisions by major manufacturers. These features include more sophisticated implementations of the basic techniques described in this chapter, as well as adaptive mechanisms that attempt to automatically improve subjective video quality.

The increasing availability of displays with wider gamuts and higher refresh rates has also prompted research into mapping existing video standards into these new domains. Also of interest are techniques for improving the display of low bit-rate and low-resolution user-generated content, which has become widely available due to the proliferation of video-capable phones and Internet sites such as youtube.com.

Further Reading

De Haan G, Bellers EB (1998) Deinterlacing-an overview. Proc IEEE 86:1839–1857

Intel Corporation. Intel® Clear Video Technology controls questions and answers. intel.com. <http://www.intel.com/support/graphics/sb/CS-029863.htm>

ISO/IEC 14496–10 coding of audio-visual objects – part 10: advanced Video Coding

Janus S (2002) Video in the 21st century. Intel Press, Hillsboro

- Keith J (2007) *Video demystified: a handbook for the digital engineer*, 2007th edn. Newnes, New York
- List P et al (2003) Adaptive deblocking filter. *IEEE Trans Circuits Syst Video Technol* 13(7):614–619
- Matchell R (1982) Digital techniques in film scanning. *IEE Proc Sci Meas Technol* 129:445–453
- Poynton CA (1996) *Technical introduction to digital video*. Wiley, New York
- Robin M, Poulin M (2000) *Digital television fundamentals*. McGraw-Hill, New York
- SMPTE 421M VC-1 compressed video bitstream format and decoding process