Multimedia Systems 8: 37–46 (2000)

Multimedia Systems

Metrics for shot boundary detection in digital video sequences

Ralph M. Ford, Craig Robson*, Daniel Temple, Michael Gerlach*****

The Pennsylvania State University, The Behrend College, Erie, PA 16563-1701, USA

Abstract. The detection of shot boundaries in video sequences is an important task for generating indexed video databases. This paper provides a comprehensive quantitative comparison of the metrics that have been applied to shot boundary detection. In addition, several standardized statistical tests that have not been applied to this problem, as well as three new metrics, are considered. A mathematical framework for quantitatively comparing metrics is supplied. Experimental results based on a video database containing 39,000 frames are included.

Key words: Scene change detection – Shot boundary detection – Video indexing

1 Introduction

To generate indexed databases, video sequences are first segmented into shots, a process that is referred to as *shot boundary detection*, *scene change detection*, or *digital video segmentation*. This is a necessary step for the identification of key frames in the video, and the subsequent retrieval of scene content. A shot is defined as one or more frames generated and recorded contiguously that represents a continuous action in time or space [4]. Video-editing procedures produce two general types of shot boundaries: abrupt and gradual. An abrupt cut is the result of splicing two dissimilar shots together, and this transition occurs over a single frame. Gradual transitions occur over multiple frames and are the product of effects such as fade- ins, fade-outs, wipes, and dissolves. The inter-frame changes are often subtle, and difficult to detect, during gradual transitions.

Many researchers have reported metrics for shot boundary detection, but there is little quantitative evidence to indicate which are the best. The work by Boreczky and Rowe [3] provided a quantitative comparison of five detection techniques: histograms (sum of absolute binwise differences), region histograms, running histograms, a pixel difference algorithm, and discrete cosine transform (DCT) differences. They concluded that region histogram techniques were the best. However, they did not compare the efficacy of different metrics to answer questions such as whether the histogram-based sum of absolute binwise differences is superior to the chi- square metric. In addition, many metrics, such as statistic-based difference metrics, were not considered.

The objectives of this paper are: (1) to provide a mathematical framework for the comparison of metrics, (2) to survey the metrics applied to this problem, and to present several metrics that have not, and (3) to provide a quantitative comparison of the metrics using grayscale and color-coded images. Experimental results indicate that the new metrics are effective for shot boundary detection.

The paper is organized as follows. Section 2 provides a mathematical framework for comparing the metrics. Metrics reported in the literature are surveyed, and the new metrics are introduced in Sect. 3. Section 4 provides a quantitative comparison of the metrics using both grayscale and color-coded images. Section 5 provides a summary and conclusions.

2 Test statistics and hypothesis testing

The method of comparing two consecutive images, or data sets, in a video sequence is known as the *statistical method of indirect proof*. First, a test hypothesis is formed. In this case, the hypothesis is that the data sets are from the same source, and likewise belong to the same shot. A metric, also known as a test statistic, is proposed for comparing the data sets and a probability law for the metric determined. If it can be shown that the metric value is improbable given the test hypothesis, the hypothesis is rejected, and it is concluded that the data sets are from different sources (a shot boundary has occurred). Deriving a probability law for a metric requires the assumption of a certain probability law on the inputs. Determining the probability law is not important for this application. It is important that a threshold can be selected that adequately discriminates between the two hypotheses, such that a correct decision is made a high percentage of the time. This is summarized for a given metric ζ as follows:

H_0 = hypothesis images are from same shot,

H = hypothesis images are from different shots,

* *Currently with*: Intel Corporation, Sacramento, Calif.
** *Currently with*: IBM Corporation, Austin, Tex.
*** *Currently with*: Raytheon Corporation, Marlborough, Mass.
Correspondence to: R.M. Ford; e-mail: rmf7@psu.edu

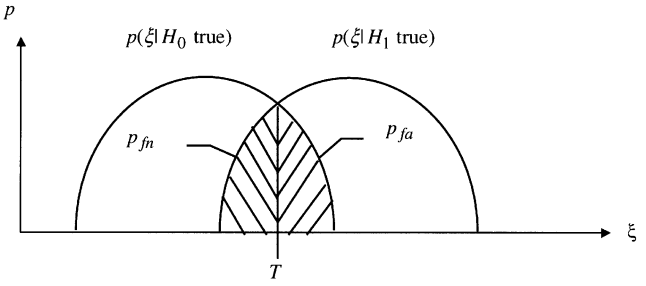

Fig. 1. Probability distributions for the two hypotheses

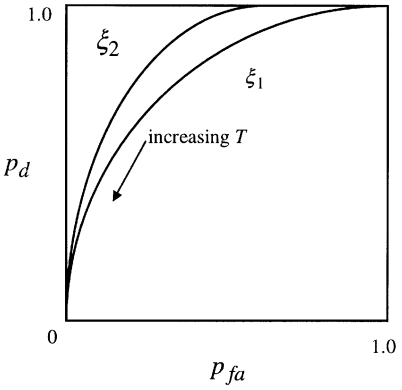

Fig. 2. Example ROCs for two test metrics

if $\zeta \geq T$ then decide H_1 true,

if $\zeta < T$ then decide H_0 true.

A decision based on the threshold, T, leads to three possible outcomes:

1. a correct decision – decide H_0 when H_0 is true, or decide H_1 when H_1 is true.
2. a false alarm (or false positive) – decide H_1 when H_0 is true.
3. a false negative – decide H_0 when H_1 is true.

Example probability distributions for the two hypotheses are shown in Fig. 1.

The probability of a false alarm is p_{fa} = p(decide $H_1|H_0$ true) and the probability of a false negative is p_{fn} = p(decide $H_0|H_1$ true). These probabilities are shown graphically by the shaded regions in Fig. 1. The probability of detecting a shot boundary is $p_d = 1 - p_{fn}$. From the law of total probability, the total probability of error for a given threshold T is

$$p_e = p_{fa} \times p(H_0 \text{ true}) + p_{fn} \times p(H_1 \text{ true}). \quad (1)$$

Intuition suggests that T be selected to minimize p_e. However, the threshold value that minimizes p_e may produce unacceptably large false-alarm or false-negative probabilities. In a typical database, $p(H_0 \text{ true}) \gg p(H_1 \text{ true})$, and therefore p_e is minimized by making p_{fa} small at the expense of p_{fn} and p_d. In our experiments, it is assumed that $p(H_0 \text{ true}) = p(H_1 \text{ true})$.

Another way to measure the performance of a metric is to consider a plot known as the Receiver Operating Characteristic (ROC) [17]. The ROC is a plot of the false-alarm and detection probabilities on the x and y axes, respectively, as a function of T. Example ROCs are shown in Fig. 2 for two test metrics. This example shows that, as T increases p_d and p_{fa} decrease, as expected from Fig. 1. Figure 2 also indicates that test metric ζ_2 is superior to ζ_1. For any given false-alarm rate, ζ_2 has a higher detection probability. The area above each curve, A_ζ, provides a way to compare metrics. A_ζ can take values in the range from 0 to 1, 1 being the worst case, and 0 the best. A_ζ takes into account all threshold values, rather than the performance for a single threshold.

3 Test metrics

Four classes of shot boundary detection metrics have been reported. The first is based on comparisons of image intensity histograms. The histograms are compared on a binwise basis, and the differences integrated into a single metric. The second is to compare images based on first- and second-order intensity statistics. This is in the form of a likelihood ratio or a standard statistical test, such as the Student t-test. The third approach is to compare images by pixel differences. A simple technique is to subtract the images point for point, and form the metric by summing the differences. The final approach is to compare images by computing changes in image structure, such as changes in edges [15] or by locating discontinuities in interframe motion vectors [1]. Motion vector methods have high computational complexity relative to the others, and were determined to be inferior in [3] to histogram metrics. Therefore, they were omitted from this comparison.

The remainder of this section surveys the metrics applied to this problem, and describes several new ones included in the comparison. The mathematical symbols employed to describe the metrics are summarized as follows: μ = mean intensity value, σ = standard deviation of intensity values, $h_j(\bullet)$ = histogram of image j, M = number of histogram bins, $I(x, y; j)$ = intensity value of point (x, y) of the jth image in the video sequence, (j, k) = indices of two consecutive images in the sequence, and n_j = number of pixels in image j.

3.1 Histogram test statistics

Nagasaka and Tanaka [10] experimented with histogram and pixel difference metrics, and concluded that histogram metrics are the most effective. They concluded that the chi-square test of significance [5] is the best histogram metric. It is computed as follows:

$$\chi^2 = \sum_{i=1}^{M} \frac{\{h_j(i) - h_k(i)\}^2}{\{h_j(i) + h_k(i)\}}, \quad \chi^2 \geq 0, \quad (2)$$

and it can be shown to obey a chi-square probability law under certain conditions. A 6-bit color-coded (M = 64) image was employed using the two most significant bits from each of the three color bands. The authors concluded that comparison based on 6- bit color-coded images was superior to grayscale images.

A problem with χ^2 is that it is not normalized, and therefore the threshold must be adjusted for different values of n.

It can be changed to a normalized measure by using the experimental probability density functions, instead of the histogram values, as follows:

$$\chi^2 = \sum_{i=1}^{M} \frac{\{h_j(i)/n - h_k(i)/n\}^2}{\{h_j(i)/n + h_k(i)/n\}} = \frac{1}{n}\sum_{i=1}^{M} \frac{\{h_j(i) - h_k(i)\}^2}{\{h_j(i) + h_k(i)\}}. \quad (3)$$

The measure in Eq. 3 was utilized in our experiments due to the various video image sizes.

Nagasaka and Tanaka [10] and Zhang et al. [16] both computed the following metric using the 6-bit color code:

$$\delta = \frac{\sum_{i=1}^{M} |h_j(i) - h_k(i)|}{n_j + n_k}, \quad 0 \le \delta \le 1. \quad (4)$$

Zhang et al. concluded that δ is a better metric than χ^2. This was based on the fact that χ^2 is more effective in emphasizing the differences between histograms, but is more likely to generate false alarms. Our experiments indicate little difference in the discrimination power of the metrics.

A histogram comparison metric, referred to as histogram intersection, was introduced in [13], where the objective was to discriminate between color objects in an image database. This measure is defined as

$$I = 1 - \frac{\sum_{i=1}^{M} \min[h_j(i), h_k(i)]}{n_j}, \quad 0 \le I \le 1. \quad (5)$$

The numerator term identifies the number of intensity values the images have in common, while the denominator is a normalizing factor. This metric was applied for shot boundary detection in [6], and its efficacy was tested under a variety of color spaces. It is shown in Appendix A that the metric I is equivalent to δ.

Nakajima et al. [11] proposed a metric based on the inner product of chrominance histograms. The histograms are treated as vectors of length M and projected onto each other as follows:

$$\alpha = 1 - \frac{\vec{h}_j \bullet \vec{h}_k}{\|\vec{h}_j\| \, \|\vec{h}_k\|}, \quad 0 \le \alpha \le 1. \quad (6)$$

If the histograms are similar, the projection is large. (Note: the measures in Eqs. 5 and 6 are subtracted from 1 to follow the convention that large metric values indicate different data sets.) This measure was used in conjunction with DCT differences to detect shot boundaries in MPEG images.

Sethi and Patel [12] proposed the use of the Kolmogorov-Smirnov test [9]. It is the maximum absolute value difference between cumulative distribution functions (CDFs), and is computed as

$$ks = \max_i |CDF_j(i) - CDF_k(i)|, \quad 0 \le ks \le 1. \quad (7)$$

They applied this to DCT-coded images in an MPEG sequence and employed a histogram of the first DCT coefficient of each block (this is the DC gray-level value of each 8×8 block).

3.2 Statistic-based metrics

Considered next are metrics based on statistics derived from the intensity data or histogram. First- and second-order statistics are usually employed, since they are more stable than those of higher order. Likelihood ratios are examined, as well as the Student t-test and Snedecor's F-test. The t-test and F-test are well-known hypothesis tests, but their use has not been reported for shot boundary detection. Finally, two new metrics are considered.

A likelihood ratio test is a standard hypothesis test [17] in which a ratio of probabilities is used as the test statistic. The general form of this test is

$$\lambda = \frac{p(\vec{I}|H_1)}{p(\vec{I}|H_0)}, \quad \lambda \ge 1. \quad (8)$$

$p(\vec{I}|H_1)$ is the probability that the data set $\vec{I} = \{\vec{I}_j, \vec{I}_k\}$ occurred given that H_1 is true, and $p(\vec{I}|H_0)$ is the probability that $\vec{I}$ occurred given that H_0 is true. Expressions for λ are typically found assuming that the data samples are identically and independently distributed. If H_1 is true and $p(\vec{I}|H_1) > p(\vec{I}|H_0)$ and λ is large, while if H_0 is true and $p(\vec{I}|H_1) = p(\vec{I}|H_0)$ and λ is equal to 1.

Jain et al. [8] computed a likelihood ratio test based on the assumption of uniform second-order statistics. Under this assumption the likelihood ratio is

$$\lambda_u = \frac{\left[\frac{\sigma_j+\sigma_k}{2} + \left(\frac{\mu_j-\mu_k}{2}\right)^2\right]^2}{\sigma_j \sigma_k}. \quad (9)$$

Under the assumption of a normal distribution, the likelihood ratio is

$$\lambda_n = \frac{\sigma_0^{n_1+n_2}}{\sigma_j^{n_1} * \sigma_k^{n_2}}, \quad (10)$$

where σ_0 is the pooled variance of both data sets. λ_n is also known as the Yakimovsky Likelihood Ratio and was applied in [12].

The Student t-test [5] tests the hypothesis that two data sets have the same mean, and therefore arose from the same cause, based on an observed difference in their sample means. The t statistic is

$$t = \frac{\mu_j - \mu_k}{\sqrt{\sigma_j^2 + \sigma_k^2}}. \quad (11)$$

This is a simplified expression for t, since $n_j = n_k$. Under certain assumptions t can be shown to obey Student's probability law. Snedecor's F- test [5] tests the hypothesis that two data sets have the same variance, and therefore arose from the same cause, based on an observed difference in their sample variances. The F statistic is

$$F = \frac{\sigma_j^2}{\sigma_k^2}, \quad \text{where } \sigma_j > \sigma_k \text{ and } F \ge 1. \quad (12)$$

Under certain assumptions the F statistic can be shown to obey Snedecor's F-distribution.

None of the statistic-based metrics simultaneously emphasize both the difference between the mean and variance

as intuition would suggest. This led us to two new metrics, which were determined through intuition and trial and error. The first metric is

$$\lambda_1 = \frac{|\mu_j - \mu_k| * |\sigma_j - \sigma_k|}{\sigma_j \sigma_k \left(\frac{\mu_j + \mu_k}{2}\right)}, \quad \lambda_1 \geq 0. \tag{13}$$

The numerator term is the product of the mean and variance differences, and the intent is to enhance the differences in the two statistics. The denominator term provides a normalization based on both the mean and variance. The second is a variation of the F-test. Instead of a ratio of the standard deviations squared, it is a ratio of the product of the mean and standard deviations squared:

$$\lambda_2 = \left(\frac{\mu_j \sigma_j}{\mu_k \sigma_k}\right)^2, \text{ where } \mu_j > \mu_k,\ \sigma_j > \sigma_k \text{ and } \lambda_2 \geq 1. \tag{14}$$

3.3 Pixel differences

Pixel difference metrics compare images based on differences in the image intensity map. Nagasaka and Tanaka [10] computed a pointwise sum of differences between image pairs as

$$\Delta_1 = \sum_x \sum_y |I(x, y; j) - I(x, y; k)|. \tag{15}$$

Jain et al. [8] and Zhang et al. [16] employed a similar measure in which a difference picture (ΔI) is computed as follows:

$$\Delta I(x, y) = \begin{cases} 1, \text{ if } |I(x, y; j) - I(x, y; k)| > \tau \\ 0, \text{ otherwise} \end{cases}, \tag{16}$$

and the metric is computed as

$$\Delta_2(\tau) = \frac{\sum_x \sum_y \Delta I(x, y)}{n}. \tag{17}$$

This requires two thresholds in contrast to the other metrics presented. Another pixel difference metric tested is the absolute value of the sum of pixel differences. It is computed as

$$\Delta_3 = \left| \sum_x \sum_y I(x, y; j) - I(x, y; k) \right|. \tag{18}$$

Images can be considered as one-dimensional vectors of length n. One way to represent the similarity between vectors is to project one vector onto the other, also known as computing the inner product. This led us to use the inner product for comparing image pairs,

$$\gamma = 1 - \frac{\vec{I}_j \bullet \vec{I}_k}{\|\vec{I}_j\| \, \|\vec{I}_k\|}, \quad 0 \leq \gamma \leq 1. \tag{19}$$

3.4 MPEG metrics

Recent attention has focused on the detection of shot boundaries in MPEG sequences due to the attractiveness of processing the compressed data directly. Arman et al. [2] used a measure similar to Eq. 19 using the coefficients of the DCT. The metric computed was

$$\psi = 1 - \frac{\vec{V}_j \bullet \vec{V}_k}{\|\vec{V}_j\| \, \|\vec{V}_k\|}, \tag{20}$$

where $\vec{V}_j$ and $\vec{V}_k$ are selected DCT coefficients from each block for the two images. In our experiments, ψ was computed using all of the non-quantized DCT coefficients.

Yeo and Liu [14] advocated the use of DC images for shot boundary detection in MPEG sequences. They are images where each pixel represents the DC value (or 0^{th} DCT transform coefficient) of each 8×8 block. This results in a significant data reduction. They applied a combination of histogram and pixel difference metrics to the DC images. In this work, all of the aforementioned metrics are computed for the DC image (non- quantized).

3.5 An edge-based metric

Zabih et al. [15] proposed a metric that relies on the number of edge pixels that change in a neighboring images. The algorithm is fairly complex, as it requires computing edges, registering the images, computing incoming and outgoing edges, and finally computing an edge change fraction. Our experiment utilized a much simpler edge comparison algorithm. The edges were computed in each image using a 3×3 Sobel operator (x and y direction), and the total number of edge pixels in each image was determined. The metric used was the modulation of the number of edge pixels:

$$E = \frac{\#edges(I_j) - \#edges(I_k)}{\#edges(I_j) + \#edges(I_k)}. \tag{21}$$

3.6 Intensity vs pixel difference metrics

Histogram- and statistic-based metrics are sensitive to lighting changes; for example, if the light flickers between frames of the same shot. These variations alter the histogram shape, and likewise the gray-level mean and variance. This produces large metric values and false positives. The advantage of these metrics is that they are invariant to large changes in object motion that do not significantly alter the light distribution. The converse is true of pixel difference comparisons. They are more robust with respect to lighting changes, and are sensitive to large interframe changes due to motion and camera zooming and panning.

The metrics may be computed globally (for the entire image) or for O blocks of the image. The latter case complicates the decision-making process, since the decision is made from the O metric values. One approach is to apply a threshold to each of the O values, and if a certain percentage exceeds the threshold decide that a change has occurred. This is the method that is typically applied. We have found that an effective approach is to compute an order statistic of the O values and use it as the test metric. Experiments with 25, 50, and 75 percentile values indicate that the 75% value works the best. This also reduces the number of thresholds required from two to one.

Table 1. Number of operations needed to compute each statistic for global computation. N = number of image pixels and M = number of histogram bins

Metric	Global computation	
	Math operations	Logical operations
χ^2	$5M + n$	M
δ	$3M + n$	–
α	$6M + n$	–
ks	$2M + n$	M
All statistic-based	$5M + n$	–
Δ_1	$3n$	–
Δ_2	$3n$	n
Δ_3	$2n$	–
γ	$6n$	–
E	$36n$	–

3.7 Computational complexity

Table 1 summarizes the computational complexity required to compute each metric. It was assumed that metric computation is optimized so that each histogram and statistic for a given image is computed only once and stored for the subsequent comparison. It was also assumed that μ and σ are computed directly from the histograms, and that n operations are necessary to compute the histogram. Mathematical operations include: $+, -.*, /$, and $|\bullet|$, while logical operations include if statement comparisons to avoid divide-by-zero errors. Extraneous operations that do not significantly add to the computational complexity are omitted. Since $n \gg M$, the histogram and statistic-based metrics have similar computation complexity, and the global computations are relatively higher. These results are all scaled upward for block comparisons.

4 Quantitative comparison of metrics

The metrics were computed for a video database containing a total of 38,858 frames. Of these, 36,288 image pairs were representative of no shot boundary, 1581 were abrupt shot boundaries, and 287 were gradual shot boundaries (each containing multiple frames). The video clips were drawn mainly from the Internet, and included MPEG-1, QuickTime, AVI, and SGI movie formats. The videos were categorized as one of the following: action, animation, comedy, commercial, drama, news, and sports. All videos were decompressed before processing. The complete video database will be made available upon request. The videos and their respective categories are in Appendix B. Many movie trailers were used, which were challenging due to the large number of shot boundaries, fast motion, and special effects. The frames were digitized at frames rates varying from 5 frames per second (fps) to 30 fps, and 8×8 blocks were used.

Three comparison values were computed for each metric, the first being the area above the ROC, A_ζ. Second, a threshold T was selected to minimize the error probability, p_e, (assuming equal prior probabilities), and the minimum value recorded. Third, the false alarm probability, p_{fa}, was determined for a fixed detection probability, p_d = 0.9. This provides a common point on the ROC for comparison, and 90% was selected as reasonable detection probability for a useful system. (Results at the level p_d = 0.95 were consistent.) The metrics were computed for grayscale and 6-bit color-coded images. A_ζ, p_e, and p_{fa} should be minimized for a good metric.

Table 2. Performance of the test statistics applied to the grayscale image database

Metric	Global comparison			Block comparison		
	A_ε	p_e min	p_{fa}@ p_d = 0.9	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram						
χ^2	2.8	7.3	5.9	4.2	6.1	5.1
δ	2.9	7.3	5.7	4.1	6.1	4.8
α	7.0	12.4	16.9	4.8	6.2	5.1
ks	2.4	6.0	5.2	3.7	5.9	5.2
Statistic						
t	4.6	10.0	10.2	3.3	6.2	5.2
F	6.2	11.8	14.3	2.1	4.8	3.4
λ_u	3.8	9.4	8.8	2.2	5.0	3.2
λ_n	3.1	7.4	6.9	2.3	4.6	3.5
λ_1	3.6	8.6	8.3	2.3	4.5	3.5
λ_2	3.1	7.1	7.0	2.4	4.5	3.7
Pixel difference						
Δ_1	4.0	9.8	10.2	4.7	10.0	10.6
Δ_2	3.8	9.6	9.6	4.2	7.5	4.4
Δ_3	4.4	10.2	10.8	3.2	7.4	6.1
γ	3.1	7.7	6.8	4.4	8.2	9.5
DCT and Edge						
ψ	5.8	10.5	11.2	–	–	–
E	9.4	16.0	25.4	–	–	–

Table 3. Performance of the test statistics applied to the color-coded image database

Metric	Global comparison			Block comparison		
	A_ε	p_e min	p_{fa}@ p_d = 0.9	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram						
χ^2	2.0	6.0	3.8	7.9	11.1	15.9
δ	3.5	8.0	6.7	8.3	11.0	17.3
α	5.8	11.6	14.0	9.3	12.0	27.1
ks	3.2	7.8	6.3	7.8	10.4	13.6
Statistic						
t	4.5	10.1	10.5	6.8	9.8	10.5
F	6.6	12.5	15.2	8.1	9.5	9.4
λ_u	3.8	9.4	9.0	6.5	9.8	9.8
λ_n	3.5	8.5	7.7	5.4	7.3	5.3
λ_1	3.6	8.9	8.1	7.9	9.3	9.0
λ_2	3.4	8.2	7.4	5.8	8.1	6.5
Pixel difference						
Δ_1	7.7	14.0	21.6	9.9	14.5	16.7
Δ_2	7.6	14.8	22.8	10.6	14.4	42.9
Δ_3	5.5	11.9	14.9	8.0	12.0	16.7
γ	3.6	8.0	6.7	8.5	15.4	20.8
DCT and Edge						
ψ	10.7	17.0	27.8	–	–	–
E	17.7	19.8	66.1	–	–	–

4.1 Abrupt cuts

The results are summarized in Tables 2 and 3 for grayscale (M = 256) and color-coded images (M = 64), respectively. Only non-shot-boundary points and abrupt cuts were included in these comparisons. First, consider the grayscale

global comparisons. The histogram metrics are the best global performers, and the Kolmogorov-Smirnov metric is the best among these. There has been little reported use of this metric [12], as opposed to the much more popular χ^2 and δ metrics [6, 7, 10, 14, 16]. The results show little difference in the efficacy of χ^2 and δ. The best statistic-based metrics are λ_1, λ_2, and λ_n, which are close in performance to the histogram metrics. The best pixel difference metric is the inner product, γ, which is on par with λ_1, λ_2, and λ_n. The color-coded global metrics do not provide a significant advantage over the grayscale, and in most cases are worse. The notable exception is χ^2 which is the best of all global metrics, color-coded or grayscale.

Now consider the grayscale block comparisons. For the histogram metrics, two of the three values used to compare the metrics show improvement relative to the global case, while A_ε is surprisingly worse. Kolmogorov-Smirnov is the best of the histogram-based metrics. There is a significant increase in performance of the statistic-based metrics, the best of these being F, λ_u, λ_n, λ_1, and λ_2. The pixel difference metrics are generally worse for the block comparisons. Among the block comparisons, the statistic-based metrics are the best. For the color- coded images, the block comparisons are generally worse than the global comparisons.

When comparing global and block results, it is apparent that the block computations are superior for the histogram- and statistic-based metrics and inferior for the pixel difference metrics. The overall best metrics are the block statistic-based metrics F, λ_u, λ_n, λ_1, and λ_2. These are followed by the block histogram metrics, the best of which is the Kolmogorov-Smirnov. Finally, they are followed by the global histogram metrics and the pixel difference metric γ. The 6-bit color-coded images do not provide an advantage over the grayscale images, the one exception being the global χ^2 metric.

The MPEG metric, ψ, using all non-quantized DCT coefficients does not perform particularly well, nor does our edge-based metric. However, it is likely that more complex edge-based algorithms would perform better, and it is possible that quantization of the coefficients would improve ψ.

All of the metrics were applied to the DC image for MPEG processing as proposed by Yeo and Liu [14] (however, the DC coefficients were not quantized as they would be in a MPEG sequence). Only global comparisons were computed, since the DC images are already a block-reduced form of the original image. The results are in Table 4. The histogram metrics all perform poorly relative to the grayscale image case, except for the Kolmogorov-Smirnov test, which performs about as well as before. The statistic-based metrics perform worse than their block grayscale performance. The pixel difference measures Δ_1, Δ_2, and γ perform significantly better. The best overall DC image metric is γ. It should be noted that this is equivalent to ψ if only the DC coefficients are utilized.

Although the performance appears to be good for many of the metrics, it must be kept in mind that simple thresholding produces a large number of false positives. For instance, the block metric λ_u has the lowest recorded false-alarm rate (3.2%) at a detection rate of 90%. This corresponds to 1161 false alarms for 1428 (of 1581) abrupt cuts detected! This indicates that use of a single metric for shot boundary detection is inadequate. The results for each video category are summarized in Tables C.1–C.7 in Appendix C. The metrics follow the same trends as observed for the overall database.

Table 4. Performance of the test metrics applied to the grayscale DC image database

	A_ε	p_e min	p_{fa} p_d = 0.9
Histogram			
χ^2	6.6	13.9	19.6
δ	5.5	12.2	15.4
α	10.9	17.6	34.3
ks	2.4	5.9	5.1
Statistic			
t	4.6	9.7	9.8
F	5.7	11.9	14.0
λ_u	3.6	9.1	8.4
λ_n	2.9	7.0	6.1
λ_1	3.5	8.8	8.3
λ_2	3.0	7.3	6.3
Pixel difference			
Δ_1	2.7	7.6	6.7
Δ_2	2.5	7.4	6.1
Δ_3	4.6	9.8	9.9
γ	1.9	5.7	3.7

Table 5. Performance of the test metrics applied to the grayscale image database, using gradual shot boundaries only

	A_ε	p_e min	p_{fa} p_d = 0.9
Histogram – Global			
χ^2	6.2	12.2	20.3
δ	7.9	12.5	26.2
α	15.1	19.1	67.7
ks	3.4	8.3	7.8
Statistic – Global			
t	3.4	8.0	7.4
F	3.0	7.0	5.9
λ_u	3.5	7.7	7.1
λ_n	2.9	6.5	6.5
λ_1	3.0	6.7	6.3
λ_2	2.9	6.4	6.4
Pixel difference – Global			
Δ_1	14.3	20.7	35.6
Δ_2	13.8	18.3	51.3
Δ_3	4.2	9.2	9.9
γ	12.1	20.0	31.9

4.2 Gradual transitions

The metrics were compared for gradual transitions exactly as done for abrupt cuts in Sect. 4.1. The database employed contained only gradual transitions and non-transition frames. Abrupt cuts were excluded. There were a total of 36,288 no-edit transitions, 145 fade transitions, and 142 dissolve transitions. The results are in Table 5. The global comparisons, were much better than the block comparisons and are the only ones shown. The results indicate that the performance of the metrics is worse for gradual boundaries than abrupt cuts. The statistic-based metrics are the best in terms of performance for gradual boundaries, all of them performing reasonably well. Among the histogram metrics, the Kolmogorov-Smirnov is the best. The pixel difference met-

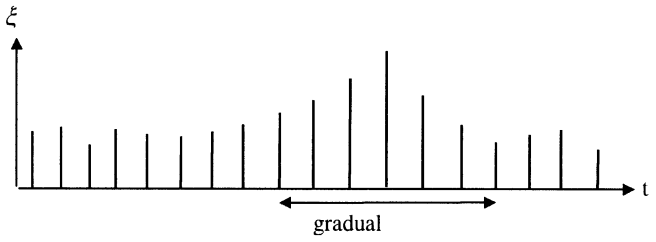

Fig. 3. Test metric behavior during a "typical" gradual transition

Table 6. Comparison ratio for the detection of gradual transitions

	ks	t	λ_u	λ_n	λ_1	λ_2	Δ_3
r – fade-outs and fade-ins	75.2	87.8	96.5	94.9	94.3	91.4	84.8
r – dissolves	67.9	69.8	83.2	85.7	83.7	76.2	67.1

rics do not perform well, the exception being the summed pixel difference, Δ_3.

Due to the difficulty in detecting gradual transitions, more complex methods have been applied to detect them. During a gradual transition, the metrics generally exhibit a slight increase that is sustained over multiple frames, as illustrated in Fig. 3. Zhang et al. [16] applied an approach called twin- comparison to detect these changes. This was accomplished by comparing the metric χ^2 to a low threshold. Then, if this threshold was exceeded, the values of χ^2 following it in the sequence were summed. If the accumulated value exceeded a second threshold, the sequence was labeled as a gradual transition. Any of the test metrics presented can be applied in this manner.

Using the twin-comparison and related approaches, gradual transitions are not detected based on an absolute threshold, but by locating a sustained increase of the metric that occurs during the transition. For reliable detection, the metric values during the transition must be sufficiently large relative to the values neighboring the transition. It was determined which metric values exhibited the largest relative increase during gradual transitions. This was accomplished by computing the average metric value during each transition and comparing it to the average value of the metric preceding and following the transition (five samples directly preceding and following each transition were used). The following ratio was computed for each of the edits:

$$r = \frac{\overline{\xi}_d - \overline{\xi}_{ba}}{\overline{\xi}_d + \overline{\xi}_{ba}}. \tag{22}$$

$\overline{\xi}_{ba}$ and $\overline{\xi}_d$ are the average values of the metric before/after and during the edit, respectively. This ratio is large for metrics that are good for detecting gradual transitions.

All metrics were normalized for a fair comparison; metrics with a non-zero minimum value were normalized to have a minimum value of 0. The average value of the ratio r is shown in Table 6 for the best metrics. Global comparisons are used, because the block comparisons provided no advantage. Seven metrics are shown, since they were clearly the best. The statistic-based metrics are best for both fades and dissolves. The summed pixel difference, Δ_3, performs well, although it was not useful for identifying abrupt cuts.

5 Conclusions

This paper provided a quantitative comparison of metrics for shot boundary detection in digital video sequences. The results are somewhat surprising. The Kolmogorov-Smirnov test is the best histogram metric; it performs better than the more popular chi- square and histogram difference (intersection) metrics. Histogram metrics produce the best results when computed for blocks, rather than globally. Among the statistic-based metrics there are several good choices, likelihood ratios and the two new metrics proposed. Again, these are best computed at the block level. Pixel difference metrics produce the best results when computed globally, the best being the inner product of images. In general, 6-bit color- coded images did not improve the performance of the metrics over the grayscale case. The overall best for abrupt cut detection are the statistic-based metrics computed at the block level.

The metrics were tested using DC images that are used in MPEG sequences. The histogram- and statistic-based metrics degrade in performance when applied to DC images, with the exception of the Kolmogorov-Smirnov metric, and the pixel difference metrics improve in performance.

To detect gradual shot boundaries the statistic-based metrics are clearly the best. A simple thresholding of these metrics produced a fairly low error probability. A relative comparison of the metrics before and during gradual transitions indicates a good response of the metrics to fades and dissolves.

Appendix A

Proof

Show the equivalence of the metrics in Eqs. 4 and 5.

Show that

$$I = 1 - \frac{\sum_{i=1}^{M} \min[h_j(i), h_k(i)]}{n_j} = \delta$$
$$= \frac{\sum_{i=1}^{M} |h_j(i) - h_k(i)|}{n_j}. \tag{A1}$$

I can be expressed as

$$I = \frac{n_j - \sum_{i=1}^{M} \min[h_j(i), h_k(i)]}{n_j}. \tag{A2}$$

The numerator of Eq. A2 is the total number of intensity values in the image minus the total number of intensity values that are the same. This is equal to the total number that are different, which can be written as

$$I = \frac{\sum_{i=1}^{M} |h_j(i) - h_k(i)|}{n_j}. \tag{A3}$$

This is equivalent to Eq. 5.

Appendix B Video database

Action

Video description	Frames	Shot boundaries
Airwolf	70	1
Barbwire movie trailer	726	175
Blade Runner	1112	21
Dune movie	296	22
Eraser movie trailer	997	115
Independence Day movie trailer	2492	97
Star Trek movie	1507	40
Star Wars movie	602	23
Starwars movie trailer	318	29
Terminator	564	4

Animation

Video description	Frames	Shot boundaries
Anastasia movie trailer	1044	101
Comet Animation	524	0
Lion King movie	1390	22
Space animations	338	0
Space probe flight	1491	9
Starwars animation	424	7
Terminator animation	331	0
Winnie the Pooh	1206	12

Comedy

Video description	Frames	Shot boundaries
Friends sitcom	571	10
Ghostbusters movie	147	5
Mighty Aphrodite movie trailer	1841	38
Rockey Horror movie	407	6
Spacejam movie trailer	1079	157

Commercial

Video description	Frames	Shot boundaries
Apple “1984”	722	27
Cartoon ad	95	3
Rice Krispies	81	5

Drama

Video description	Frames	Shot boundaries
A Few Good Men Movie	865	12
Alaska movie trailer	777	60
American President movie trailer	1375	58
Bed Time for Bonzo	436	15
Chung King movie trailer	1225	72
Close Encounters movie	566	9
Crossinguard movie trailer	3038	110
Crow movie trailer	1560	122
First Knight movie trailer	271	15
Jamaica	115	2
My Left Foot movie trailer	449	14
Slingblade movie trailer	1704	101
Titanic movie	1067	14
Titanic movie trailer	1210	173
Truman movie trailer	166	8
Xfiles trailer	429	20

News

Video description	Frames	Shot boundaries
CNN news	169	7
Plane crash newsclip	96	0
Reuters newsclips	1126	11
Ron Brown’s funeral	631	24
San Jose news	1377	6
Singer news clip	350	0
Space shuttle disaster	1411	13
Space shuttle Endeavor astronauts	1190	2
Space station Mir	588	5
Sunrise/sunset	286	2
Weather satellite clips	308	0
White House footage	131	5

Sports

Video description	Frames	Shot boundaries
Basketball	487	6
Hockey	205	11
Rodeo	146	3
Skateboarding	298	6
Sky surfing	74	0
Soccer	87	0

Note: In QuickTime movies, repeat frames were omitted. Therefore, the total frames listed here is greater than that included in results.

Appendix C

Video category results – Abrupt cut comparisons

Table C.1. Category: Action. Frames: 8767. Boundaries: 595

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	5.5	8.2	8.2
δ	5.4	8.3	7.9
α	6.0	8.2	8.7
ks	5.3	8.4	9.0
Statistic – Block			
t	5.3	9.3	10.4
F	2.8	6.5	6.1
λ_u	3.6	7.6	7.4
λ_n	3.6	6.7	7.1
λ_1	3.8	7.4	7.5
λ_2	4.2	7.4	8.1
Pixel difference – Global			
Δ_1	5.2	11.2	12.5
Δ_2	5.5	10.9	12.2
Δ_3	6.7	11.9	14.2
γ	5.2	10.9	12.0
Pixel difference – DC Image			
γ	3.2	6.9	6.5

Table C.2. Category: Animation. Frames: 6495. Boundaries: 133

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	2.6	2.9	4.4
δ	2.7	2.9	4.4
α	2.6	2.8	4.2
ks	2.9	3.2	4.5
Statistic – Block			
t	2.3	3.0	4.1
F	1.2	2.8	2.9
λ_u	1.3	1.7	2.5
λ_n	2.0	3.0	3.5
λ_1	1.1	1.8	1.7
λ_2	1.1	2.1	2.2
Pixel difference – Global			
Δ_1	1.2	4.0	3.6
Δ_2	0.9	2.9	1.7
Δ_3	4.2	10.0	10.7
γ	0.8	2.3	1.6
Pixel difference – DC Image			
γ	0.8	2.1	1.4

Table C.3. Category: Comedy. Frames: 3400. Boundaries: 2

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	3.4	4.9	3.0
δ	3.2	4.8	2.9
α	5.0	4.9	3.1
ks	2.5	4.6	2.5
Statistic – Block			
t	2.5	4.8	2.5
F	2.1	4.6	2.7
λ_u	1.9	3.3	1.7
λ_n	2.0	3.7	2.1
λ_1	2.0	3.3	2.7
λ_2	2.2	3.4	2.9
Pixel difference – Global			
Δ_1	1.8	5.3	4.5
Δ_2	1.6	5.4	2.9
Δ_3	2.5	7.3	5.1
γ	1.7	5.4	3.2
Pixel difference – DC Image			
γ	0.9	3.8	1.5

Table C.4. Category: Commercial. Frames: 844. Boundaries: 23

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	6.2	5.8	2.8
δ	6.1	6.2	2.8
α	5.7	5.7	2.8
ks	5.1	5.0	3.7
Statistic – Block			
t	4.9	4.6	4.0
F	5.1	5.0	1.3
λ_u	4.8	5.8	2.9
λ_n	4.8	6.5	4.5
λ_1	4.9	5.0	1.2
λ_2	5.0	5.0	1.2
Pixel difference – Global			
Δ_1	6.3	5.6	2.6
Δ_2	6.6	5.6	2.6
Δ_3	6.7	12.0	15.2
γ	4.5	3.7	3.0
Pixel difference – DC Image			
γ	3.9	3.7	5.6

Table C.5. Category: Drama. Frames: 11321. Boundaries: 550

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	3.3	4.7	3.2
δ	3.2	4.7	3.1
α	3.9	4.7	3.0
ks	2.8	4.4	3.0
Statistic – Block			
t	2.8	4.9	3.4
F	2.2	3.9	2.7
λ_u	2.0	4.0	2.3
λ_n	2.1	3.9	2.6
λ_1	2.1	3.8	2.5
λ_2	2.0	3.6	2.5
Pixel difference – Global			
Δ_1	4.0	9.3	10.0
Δ_2	3.4	8.1	7.8
Δ_3	3.5	8.6	7.8
γ	2.4	6.1	4.3
Pixel difference – DC Image			
γ	1.6	4.7	3.4

Table C.6. Category: News. Frames: 5106. Boundaries: 42

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	3.1	3.3	0.1
δ	3.1	3.5	0.1
α	3.1	3.9	0.1
ks	2.7	2.8	4.7
Statistic – Block			
t	0.7	1.3	1.5
F	0.2	1.5	0.6
λ_u	0.1	0.3	0.3
λ_n	0.4	0.8	0.7
λ_1	0.3	2.0	0.7
λ_2	0.3	1.5	0.5
Pixel difference – Global			
Δ_1	0.3	1.1	0.6
Δ_2	0.3	1.8	0.2
Δ_3	2.2	5.0	3.6
γ	0.5	1.1	1.1
Pixel difference – DC Image			
γ	0.5	0.9	1.2

Table C.7. Category: Sports. Frames: 1375. Boundaries: 21

	A_ε	p_e min	p_{fa}@ p_d = 0.9
Histogram – Block			
χ^2	1.3	4.2	1.5
δ	1.6	5.7	1.2
α	1.5	5.5	1.5
ks	1.4	5.5	2.0
Statistic – Block			
t	1.1	4.0	2.0
F	3.3	7.3	5.2
λ_u	0.7	2.5	1.5
λ_n	1.8	5.7	1.8
λ_1	3.2	5.2	5.1
λ_2	3.4	5.3	5.5
Pixel difference – Global			
Δ_1	3.6	10.0	17.6
Δ_2	1.1	2.9	1.9
Δ_3	6.2	11.1	22.6
γ	1.7	3.3	1.7
Pixel difference – DC Image			
γ	0.9	3.5	2.0

References

1. Akutsu A, Tonomura Y, Hashimoto H, Ohba Y (1992) Video Indexing Using Motion Vectors. Proc SPIE (Visual Communications and Image Processing) 1818: 1522–1530
2. Arman F, Hsu A, Lee MY (1993) Image Processing on Compressed Data for Large Video Databases. In: Proceedings International Conference on Multimedia, 1993, August 20, 1993, Anaheim, CA, pp 267–272
3. Boreczky JS, Rowe LA (1996) Comparison of Shot Boundary Detection Techniques. Proc SPIE (Storage and Retrieval for Still and Image Video Databases IV) 2664: 170–179
4. Davenport G, Smith TA, Pincever N (1991) Cinematic Primitives for Multimedia. IEEE Comput Graphics Appl 11(4): 67–74
5. Frieden BR (1991) Probability, Statistical Optics, and Data Testing. Springer, Berlin
6. Gargi U, Oswald S, Kosiba D, Devadiga S, Kasturi R (1995) Evaluation of Video Sequence Indexing and Hierarchical Video Indexing. Proc SPIE (Storage and Retrieval in Image and Video Databases III) 2420: 144–151
7. Hampapur A, Jain R, Weymouth TE (1995) Production Model Based Digital Video Segmentation. Multimedia Tools Appl 1: 1–38
8. Jain R, Kasturi R, Schunck BG (1995) Machine Vision. McGraw-Hill, New York, pp 406–415
9. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1993) Numerical Recipes: The Art of Scientific Computing, 2nd edn. Cambridge University Press, Cambridge
10. Nagasaka A, Tanaka Y (1992) Automatic Video Indexing and Full-Video Search for Object Appearances. Visual Database Syst 11: 113–127
11. Nakajima Y, Ujihara K, Yoneyama A (1997) Universal Scene Change Detection on MPEG-coded data Domain. Proc SPIE (Visual Communications and Image Processing) 3204: 992– 1003
12. Sethi IK, Patel N (1995) A Statistical Approach to Scene Change Detection. Proc SPIE (Storage and Retrieval for Image and Video Databases III) 2420: 329–338
13. Swain MJ, Ballard DH (1991) Color Indexing. Int J Comput Vision 7(1): 11–32
14. Yeo B-L, Liu B (1995) Rapid Scene Analysis on compressed Video. IEEE Trans Circuits Syst Video Technol 5(6): 533–544
15. Zabih R, Miller J, Mai K (1995) A Feature-Based Algorithms for Detecting and Classifying Scene Breaks. In: Multimedia Systems 7(2): 119–128 (1999)
16. Zhang HJ, Kankanhalli A, Smoliar SW (1993) Automatic Partitioning of Full-Motion Video. ACM Multimedia Syst J 1: 10–28
17. Trees HL van (1982) Detection Estimation and Modulation Theory: Part I. Wiley, New York

RALPH M. FORD received the B.S. degree from Clarkson University in 1987, and his M.S. and Ph.D. degrees from the University of Arizona in 1989 and 1994, respectively, all in electrical engineering. He was a development engineer at IBM, East Fishkill, N.Y., from 1989 to 1991, where he worked on the development of machine vision systems for industrial inspection. He has been a faculty member in the Electrical and Computer Engineering Program at Penn State Erie since 1994, and holds the rank of assistant professor. His research interests are in the areas of image processing, pattern recognition, and computer vision. He is a member of the Institute of Electrical and Electronics Engineers and the International Society for Optical Engineering.

CRAIG ROBSON received the B.S. degree in Electrical Engineering from Penn Stae Erie in 1996. He is currently pursuing the M.S. degree in Computer Science at the California State University. As a product engineer at Intel from 1996 to 1998 he worked on reverse engineering of flash memory devices. He is currently a design engineer at Intel working on microprocessor design at their Folsom site.

DANIEL A. TEMPLE received the B.S. degree in Electrical Engineering from Penn Stae Erie in 1996. He is currently a custom logic designer at IBM Microelectronics, Austin, Texas. He has been with IBM since 1996 and has worked on various ASIC and card designs, PowerPC bring up at the IBM Motorola Somerset Design Center, and has submitted five multimedia patents. Other work and research has included studies in image processing, stress chains in granular media, digital audio processing, and applications of DVD and MPEG standards. Dan is also a full-time musician, composer, and producer, currently working on his fourth CD release, while pursuing an underground distribution and promotion contract.

MICHAEL GERLACH received the B.S. degree in Electrical Engineering from Penn Stae Erie in 1996. He is currently a hardware design engineer at Raytheon Electronic Systems in Marlborough, Mass., where he designs satellite communication equipment.