Where Does Computational Media Aesthetics Fit?

Brett Adams
Curtin University of Technology, Australia

The huge volume of multimedia data now available calls for effective management solutions. Computational Media Aesthetics (CMA), one response to this data-management problem, attempts to handle multimedia data using domain-driven inferences. To provide a context for CMA, this article reviews multimedia content management research.

The Multimedia Content Description Interface (MPEG-7) is the Moving Picture Experts Group's ISO standard for describing multimedia content. It provides a rich set of standardized tools for describing multimedia content. The standard's overview contains the rather nostalgic lament, "Accessing audio and video used to be a simple matter—simple because of the simplicity of the access mechanisms and because of the poverty of the sources." This is clearly no longer the case.

Although none of the media in multimedia are new, the sheer volume in which they're stored, transmitted, and processed is, and directly results from the prevailing winds of Moore's law that continue to deny the doomsayers and power a relentless improvement of relevant technologies. The tidal wave of unmanaged and unmanageable data that has developed over the last decade and is outstripping our ability to ultimately use it has motivated the growing drive for management solutions. A book without a contents page or index is an annoyance; a data warehouse with terabytes of video is nearly useless without a means of searching, browsing, and indexing the data. In short, such content is wasted without suitable content management.

Computational media aesthetics (CMA) is one response to the problem posed by multimedia content management (MCM).[1] CMA focuses on *domain distinctives*, the elements of a given domain that shape its borders and define its essence (in film, for example, shot, scene, setting, composition, or protagonist), particularly the expressive techniques used by a domain's content creators. This article seeks to provide a context for CMA through a review of MCM approaches.

The semantic gap

Many approaches to MCM are responses to the much-publicized *semantic gap,* the sharp discontinuity between the primitive features automated content management systems currently provide, and the richness of user queries encountered in media search and navigation, which impact users' ability to comfortably and efficiently use multimedia systems.[2]

Although the semantic gap problem is complex, it essentially results from the connotational relations that human interpretation introduces into a problem's semantic framework, in addition to the already present denotational meanings. Say you want to retrieve an image that contains lush, forested hills. There already exists a many-to-one mapping between the signifier (the image) and the signified (green hills). To capture this relational multiplicity, you must extract the image features that capture the invariant properties of "green hills," such as the color green. If you change the query to "tranquil scenes," the problem becomes many-to-many: the many-to-one denotational link of features to "green hills," and the one-to-many connotational relations of "green hills" to other associated meanings, such as "tranquil" or "beautiful." Figure 1 outlines these relationships.

The presence of a semantic gap invokes a wide variety of policies regarding reasoning framework and semantic authority.

Managing multimedia content

In 1994, Rowe et al. conducted a survey aimed at determining the kinds of queries that users would like to put to video-on-demand (VoD) systems.[3] They identified three types of indexes that are generally required, of which two are of interest:

- Structural (for example, segments, scenes, and shots), and
- Content (for example, objects and actors in scenes).

The third index type, bibliographic—title, abstract, producer, and so on—is too specific for a broad analysis of MCM. Structure-related indexes

1070-986X/03/$17.00 © 2003 IEEE

Published by the IEEE Computer Society

use *primitive* features, while content-related indexes use *abstract* or *logical* features.

Structural indexing

Primitive features infer nothing about the content of a particular cluster—only that the content is different from that in surrounding clusters. Segmenting data into meaningful "blobs"—that is, finding boundaries within the data—is one of the most fundamental requirements of any MCM-related task. Depending on the domain, structural units can be shots, paragraphs, episodes, and so on. Some terminology applies to more than one domain (for example, we can refer to both newscasts and feature films in terms of scenes).

The most broadly applicable structural unit is the *shot*, a piece of film resulting from a single camera run. A shot can be a single frame or many thousands of frames, and as such forms the most basic visual structure for any multimedia data that includes camera footage or simulated camera footage via classical or computer-aided animation. Consequently, the shot is usually the first element detected by a MCM system processing multimedia data.

An *edit* or transitional device joins two consecutive shots. An edit can be a cut, a fade in or out, a dissolve, or a special transition such as a wipe or any number of special effects. Segmenting shots, therefore, involves generating an index of transitional effects. For many applications, cut detection (identifying where two disjoint pieces of footage have been spliced together) is mostly a solved problem, with adequate sustainable precision and recall performance. Detecting other transitional devices remains an active area of research, but the shot index with which dependent processes must work is generally adequate to the task.

Shot segmentation alone is only marginally helpful. For example, assume an average short novel has 10 paragraphs per page, meaning the entire work would have from 1,000 to 2,000 paragraphs. This figure is similar to the number of shots that make up an average feature-length film. If the novel's table of contents listed every paragraph, it would resemble, in usefulness, what we obtain when we segment multimedia data into shots alone. Although it might be useful for a class of readers, it would be inadequate to the needs of most readers.

The inadequacy of purely shot-based indexes has prompted researchers to investigate higher-order taxonomies. Do abstractions above the shot exist? Do even further abstractions exist beyond these? These taxonomies demand methods for clustering shots into hierarchical units, which in turn require a similarity measure. Several routes to a shot similarity measure exist, but nearly all of them start with a simple representation.

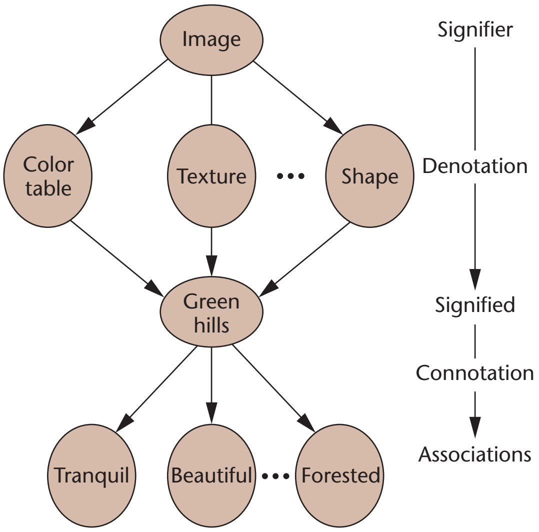

Figure 1. The semantic gap engenders a multiplicity of relations. Denotational links exist between signifier and signified, while connotational relations exist between signified and associated meanings.

Keyframe similarity measure. A keyframe is a common technique for representing a shot. In effect, keyframes reduce a series of shots to a series of images for the purposes of judging similarity.

The simplest policy for obtaining keyframes from a series of shots is to take the first frame, or the first and last frames, of each shot.[4] Zhuang et al., however, note that for a frame to be representative of a shot, it should contain the shot's "salient content."[5] This has led to more complex policies for selecting keyframes with this rather abstract property.

Yeo and Liu and Gunsel and Tekalp extract multiple keyframes by comparing color changes if motion has substantially changed color composition.[6] Zhang et al. also detect cinematic elements, such as zooming and panning, to generate keyframes (first and last frame of a zoom and panning frames with less than 30 percent overlap).[6] Wolf calculates motion estimates based on optic flow and selects local minima as

> **Scenes are … remarkably agile, eluding many schemes formulated to detect them.**

keyframes.[6] He assumes that important content will cause the camera to pause and focus on it.

Even more complex policies use clustering algorithms. Such approaches cluster frames using a similarity metric, such as color histograms, and then select a keyframe from the most significant clusters.

Atemporal shot similarity. Regardless of how you extract the keyframes, you'll end up with a series of representational images for the shot sequence. You can then use *image metrics*—the similarity between shots A and B reduces to the similarity between their respective keyframes.

Pentland et al., for example, use a *semantic preserving* representation to enable image search and retrieval, and note its application to video via keyframe search.[7] They attempt to align feature similarity with human-judged similarity using "perceptually significant" coefficients they extract from the images. In particular, they sort video keyframe similarity using appearance- and texture-specific descriptions.

Other work using image content to determine similarity includes the Query by Image Content (QBIC) system, which also indexes by color (histograms), texture (coarseness, contrast, and directionality), and shape (area, circularity, eccentricity, and so on). VisualSeek uses indexes for region color, size, and relative and absolute spatial locations to find images similar to a user's diagrammatic query.

Chang and Smith bridge image and video domains by basing shot similarity on keyframe image features such as color, texture, and shape, assuming that each video shot has consistent visual feature content.[8] Their work targets art and medical image databases and VoD systems.

Atemporal similarity features found in work explicitly directed at video generally draw from these pioneering sources in the image-similarity-matching domain. *Setting,* a key video feature, provides a correspondence of the general background or objects that make up the viewable area from one frame to the next within a given shot. Developers typically harness setting-based similarity using a color histogram-based feature, which colocates—with respect to a distance metric—keyframes of a similar setting, while remaining largely invariant to common video transformations such as camera angle change.

Gunsel and Tekalp use YUV space color histogram differences to define similarity between shots.[6] They use the equation

$$D_{xy} = \sum_{i=0}^{G} \Big(\left| H_x^Y(i) - H_y^Y(i) \right| + \left| H_x^U(i) - H_y^U(i) \right| + \left| H_x^V(i) - H_y^V(i) \right| \Big) \qquad (1)$$

where G is the number of bins and $H(i)$ the value for the ith Y, U, or V color bin, respectively. Presumably, applying a threshold to the constructed $N \times N$ similarity matrix (where N is the number of shots) results in shot clusters of user-specified density.

You can constrain cluster formation beyond a shot's visual features. Applying time constraints to the shot similarity problem, for example, recognizes the existence of the *scene* or *story unit* structure within a given film, and the binding semantic relationship they impart to the shots within the structure. The assumption here is that true similarity lies not in visual similarity but in the relationships that are formed and mediated by the scenic construct. Part of this construct is the proximity in time of participating shots, which time-constrained models attempt to reflect through shot similarity.

Yeung et al. combine visually similar shot clustering (based on keyframe color histograms) with shot time proximity to obtain the scene's higher-level video structure.[4] They augment their approach with shape and other spatial information.

A scene is a dramatic unit of one or more shots usually taking place during one time period and involving the same setting and characters. Generally considered the most useful structural unit on the next level of the video structure taxonomy, scenes are a popular target for video segmentation. In practice, however, they're remarkably agile, eluding many schemes formulated to detect them.

Hanjalic et al. segment movies into logical story units (LSUs) or *episodes* using a visual dissimilarity measure.[9] The measure is simply a color histogram difference applied to a possibly composite keyframe (in the case of shots with multiple keyframes). Their algorithm, also called the *overlapping links method,*[10] uses three rules to generate the LSU segmentation from the shot visual dissimilarity measures.

Figure 2 shows an example episode these rules detected.

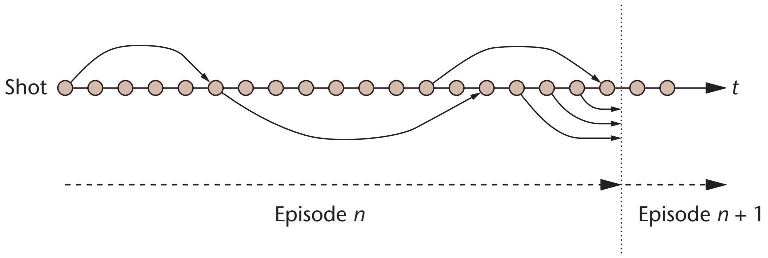

Figure 2. Story unit formation via overlapping links. Arrows indicate visually similar shots, which help form the boundaries of the story unit (or episode).

Unbiased test subjects manually generate scene *groundtruth*—the canonical list and location of story units against which we can assess system performance—and boundaries recorded by all subjects are deemed probable and kept. Hanjalic et al. note that many of the missed boundaries are scenes that form part of a larger sequence, for example, a wedding ceremony, reception, and party.[9]

Zhao et al. measure shot similarity using the weighted sum of a keyframe visual similarity component and a shot temporal distance component, assuming that the visual correlation of scene shots diminishes over time.[11] They then subject the shot similarity sequence to a sliding window, a simpler approach than the overlapping links method. A scene boundary forms wherever the ratio of shot similarities on either side of the middle shot exceeds a threshold. The authors assume that scenes are semantically correlated shots, and therefore boundary detection involves determining two shots' semantic relationship (compounded by means of the sliding window).

Temporal shot similarity. Aside from assumptions such as setting and temporal proximity, we must also consider the full spatial–temporal nature of video and the rich information source it provides. An arbitrary image separated from an inference-enabling context defies useful association, but frame images have a special relationship, dictated by the constraints of the filming process, with the preceding and/or following image.

The most common temporal features for determining shot similarity are shot duration, motion (frame-to-frame activity or optic flow), and audio characteristics (frequency analysis, for example). Veneau et al. include shot duration, perhaps the simplest temporal feature, as one of three shot signatures and use the Manhattan distance to cluster shots into scene transition graphs (STGs).[12] The thrust of their work, however, is the *cophenetic matrix*—a matrix of the similarity values at which a pair of objects, in this case shots, become part of the same cluster—and the user's ability to tune the segmentation threshold.

Rui et al. introduce *time adaptive grouping,*[13] in which shot similarity is a weighted function of visual similarity and time locality. They also include a shot activity temporal feature in the visual component:

$$Act_i = \frac{1}{N_i - 1} \sum_{k=1}^{N_i - 1} Diff_{k,k-1} \tag{2}$$

where Act_i is the activity of the *i*th shot, N_i is the number of frames in shot *i*, and $Diff_{k,\,k-1}$ is a color histogram difference between successive frames. They calculate shot similarity as

$$ShtSim_{i,j} = W_c * ShtClrSim_{i,j} + W_a * ShtActSim_{i,j} \tag{3}$$

where W_c and W_a are color and activity weights, respectively; and *i* and *j* are two shots (every shot *i* is compared with every other shot *j*). *ShtSim* is shot similarity; *ShtClrSim* is shot color similarity; and *ShtActSim* is shot action similarity. They factor each shot similarity component by a *temporal attraction* value, which decreases as the respective frames grow apart. *ShtSim* forms groups of shots, and then their system applies a scene construction phase, similar in effect to the overlapping links method.

Hammoud et al. cluster shots based on color, image correlation, optic flow, and so on.[14] Using an extension of Allen's relations, they form the clusters into a temporal cluster graph that provides semantic information such as "this scene occurs during this one"—that is, the scene is an inset such as a flashback. Mahdi et al. extend this work to remove one-shot scene anomalies.[15] Assuming that similar shot durations belong to the same scene, they add a rhythm constraint to check that the difference between the shot thought to be a scene boundary and the shot previous (minus the mean) are within a certain number of standard deviations from the entire cluster variation.

Huang et al. observe that scene changes are usually accompanied by color, motion, and audio change, whereas shot changes usually produce only visual and/or motion changes.[16] Their feature set includes a color histogram, phase correlation function similar to a motion histogram, and a set of clip-level audio features (including nonsilence ratio, frequency centroid, and bandwidth).

Kender and Yeo seek scenes or story units with a shot-to-shot coherence measure.[17] Using frame similarity metrics, they aim to transform the shot sequence rather than parse it, thus leaving room for a user-specified sensitivity level. Their algorithm includes a human memory retention model that seeks to capture the extent to which we can perceive and assimilate temporally near and visually similar stimuli into higher-order structures. Coherence is essentially how well a shot recalls a previous shot in terms of its color similarity and the time between the two shots. Candidates for scene segmentation appear where this recall is at a local minimum.

Sundaram and Chang extend this concept of coherence, coupling it with audio segmentation.[18] They define a video scene as a collection of shots with an underlying semantic, and assume the shots are chromatically consistent. Audio scenes contain a number of unchanging dominant sound sources, and scenes are shot sequences with consistent audio and video characteristics over a specified time period. Hence, they label scene boundaries where a visual change occurs within an audio change's neighborhood.

Vendrig et al. note that previous approaches fail to achieve truly robust results because "visual similarity as computed by image-processing systems can be very different from user perception."[19] Some features might segment only part of a film, or some films and not others. They throw the problem back into an interactive setting. In their approach, the LSU segmentation groundtruth depends on users who terminate the session after attaining the desired segmentation. After an initial automatic segmentation, consecutive LSUs that might have resulted from overclustering (through a shot number threshold) are subjected to a number of automatically selected features. The user then rates the features' effectiveness in terms of the shot similarity results.

Like Vendrig and Worring,[10] Truong et al.[20] mention the two major trends in scene boundary extraction:

- time-constrained clustering and
- time-adaptive grouping.

They note that time-constrained clustering depends on clustering parameters, and that clustering inhibits a system's ability to observe shot progression, which helps it find scene boundaries. Time-adaptive grouping depends on finding local minima within a noisy signal, and refers to viewer perception rather than cinematic convention. The authors also assert that neither technique adequately deals with at least one of two issues:

- Researchers should model shot color similarity as continuous rather than discrete, because changing camera angles or motion might result in filming shots within a scene with different lighting or shading.
- Fast motion or slow disclosure shots can cause only part of a shot to be similar to another, and developers should therefore use the same number of frames to evaluate this similarity.

Their shot similarity metric addresses the first issue using an algorithm that gradually computes, then excludes, regions with the highest color shade similarity by recursively adding component color similarities from the most similar to the least for a given representative frame pair. Truong et al. address the second issue by applying this color similarity metric to any two representative shot frames from a pair of shots and recording the maximum similarity found.[20] Film convention is explicitly the dominant force behind algorithmic decisions.

Wang et al. introduce a scene-extraction method based on a shot similarity metric that includes frame feature (color moments and a fractal texture feature) substring matching to detect partial similarity.[21] A sliding pair of tiles, similar to Zhao et al.'s window,[11] generates a shot-by-shot visual dissimilarity measure, with local minima consequently deemed *scene segments*. They then merge scene segments into more complex scene types based on the number of visually similar threads in a segment and the camera focal length behavior.

The authors classify five scene types: parallel, concentration, enlargement, general, and serial. The approach is currently of limited practical use because they must manually generate camera focal length information. What's enlightening, however, is the attention to general filmic techniques, and the attempt to detect them.

All of the approaches detailed thus far are founded on some measure of shot similarity. Regardless of explicit domain, the shot construct is key to unlocking views of the veiled semantic landscape. Shot similarity measures can drive inferences of the form, "the texture or colors of this shot is like this other shot, and not like that shot," and it's this power that the software harnesses, initially for simple clustering, and then with greater domain directedness toward scene segmentation, and so on. Given content management's semantic nature, however, simple shot-similarity-based methods can't address some of the most useful questions, such as Where is the film's climax? or Is this the sports section of a newscast?

Content indexing

Logical or abstract features map extracted features to content. Although these features can address the segmentation problem, they naturally target a new problem class—content—and applications that depend on that knowledge, such as genre recognition or scene classification.

Another way to compare the emphases of primitive feature-based work and abstract feature-based work is to consider characterizing functions based on similarity or discrimination. Similarity seeks to determine objects' relations to each other. Discrimination aims to determine if an instance object qualifies as a member of a particular class. A discriminant function might detect a face within a shot whereas a similarity function might capture how two faces are alike. Obviously one type can include part of the other.

Abstract features for explicit indexing

Beyond supporting similarity and segmentation, abstract features enable powerful explicit indexing. In the image retrieval realm, some researchers claim that the only route to semantically rich indexing ("this image contains a dog," for example) is through human annotation. Is this also true for the larger multimedia domain, and for film in particular? Many researchers are seeking the filmic analog of tools to find the aforementioned dog—that is, content-related information meaningful in the context of film.

Semantic indexing and scene classification. Nam et al. apply a toolbox of feature sets for characterizing violent content signatures—for example, an activity feature detects action, color-table matching detects flame, and an energy entropy criterion captures sound bursts.[22] The authors gathered their data sets from several R-rated movies and graph a sampling of their results. They note that "any effective indexing technique that addresses ... higher-level semantic information must rely on user interaction and multilevel queries."

Given content management's semantic nature, simple shot-similarity-based methods can't address some of the most useful questions.

Yoshitaka et al. mix shot length, summed luminance change (shot dynamics), color histogram similarity, and shot repetition patterns to classify scenes as conversation, increasing tension, or hard action.[23] They classify scene type using a rule hierarchy, from less to more strict. For example, the least strict conversation scene detection rule simply requires a shot pattern of either *ABA′B′* or *ABB′A′*, whereas the most strict requires (*ABA′B′* or *ABB′A′*) and (visual dynamics of each shot < σ) and (shot length of each shot > τ). Film grammar—the body of rules and conventions for the filmmaking craft—explicitly motivates this approach, unlike the implicit approach of Yoshitaka's more recent work.

Saraceno and Leonardi also propose a scene classifier.[24] Their system identifies four scene types: dialog, story, action, and generic (not belonging to the first three types, but with consistent audio characteristics). Like Yoshitaka et al., they separate audio from visual processing and then use a rule set to recombine them, but they also classify scenes by audio type (silence, speech, music, and miscellaneous), leveraging these types to distinguish the scene classes.

With a broader domain and an accordingly altered scene definition, Huang et al. classify television-derived data as news, weather, basketball, or football.[25] They attempt to capture the different genres' timeliness by exploring competing hidden Markov model (HMM) strategies.

In one strategy, they combine all features in a super vector that they feed to the HMM, which is an effective classifier but training-data hungry. Another, extensible, strategy recognizes the lack of correlation among modal features (audio, color, and motion) and trains an independent HMM for each mode. The authors note that all strategies provide better performance than single modalities, as multimodal features can more effectively resolve ambiguities.

Alatan et al. address scene classification by reflecting content statefulness.[26] Their system classifies audio tracks into speech, silence, and music coupled with visual information such as face and location to form an audio–visual token, which it passes to an HMM. They identify useful properties of statistically based approaches, particularly as they relate to natural language, which they view as similar to film. They attempt to model dialogue scenes, action scenes, and establishing shots to create a dialogue/nondialogue classification. The system can only split the given data into three consecutive scenes, however. They obtain groundtruth subjectively—from the first words of a conversation to the last.

Assuming that semantic concepts are related, and hence their absence or presence can imply the presence of other concepts, Naphade and Huang seek to model such relations within a probabilistic framework.[27] Their system contains *multijects*, probabilistic multimedia objects, connected by a *multinet*, which explicitly models their interaction. The system can then exploit the existence of one object (whose features are perhaps readily recognizable) to detect related concepts (whose features are not so invariant) via these associations. In such a setting, the system can use prior knowledge (such as the knowledge that action movies have a higher probability of explosions than comedies) to prime the belief network. The aim of their work is semantic indexing, and they use the multiject examples of sky, snow, rocky terrain, and so on.

Roth also considers concepts within contexts, rather than in isolation.[28] His system represents knowledge about a given film using a propositional network of semantic features. *Sensitive regions*, or hot spots delineating regions of interest in successive frames, represent information of interest—that is, “principal entities visible in a video, their actions, and their attributes.” The system doesn’t attempt to determine hot spots automatically; rather, the main thrust is querying such representations. Roth’s attempt to couple a knowledge base containing an ontological concept hierarchy to sensitive region instances perhaps nears the extreme of envisaged semantic representation for film.

Genre discrimination. Fischer et al. classify video by broad genre using style profiles developed inductively via observation.[29] Profiles include news, tennis, racing, cartoons, and commercials. The authors build style profiles for each genre based on shot length, motion type (panning, tilting, zooming, and so on), object motion, object recognition (specifically logo matching, with particular application to newscasts), speech, and music. Each style attribute detector reports the likelihood of the video belonging to each genre based only on its style attribute. The system then pools the detectors’ results using weighted averages and produces the winning classification. The authors conclude that even within this limited context, no single style attribute can distinguish genre; rather, fusing attributes produces a much more reliable classification. They also note that “film directors use such style elements for artistic expression.”

Sahouria and Zakhor’s principal components analysis- (PCA-)based work classifies sports by genre.[30] Arguing that motion is an important attribute with the desirable property of invariance despite color, lighting, and to a degree scale changes, they develop a basis set of attributes for basketball, ice hockey, and volleyball. They stress the motions inherent to each—for example, “hockey shows rapidly changing motions mostly of small amplitude with periods of extended motion, while volleyball exhibits short duration, large magnitude motions in one dimension.” In effect, the content bubbles to the surface through the grammar of the coverage.

As a first step in constructing semantically meaningful feature spaces to capture properties such as violence, sex, or profanity, Vasconcelos and Lippman categorize film by “degree of action.”[31] They begin with the premise that action movies involve short shots with a lot of activity. Then, they map each movie into a feature space composed of average shot activity based on *tangent distance,* a lighting and camera-motion invariant, and average shot duration. They obtain genre groundtruth from the Internet Movie Database (http://www.imdb.com), segmenting their results into regions, with comedy/romance at one extreme and action at the other. The authors suggest a simple Gaussian classifier based on the mapping would achieve high classification accuracy.

In other work,[32] Vasconcelos and Lippman present their Bayesian modeling of video editing and structure (Bmovies) system. They summarize video in terms of the semantic concepts action, close-up, crowd, and setting type, using the structure-rich film domain in the form of priors for the Bayesian network. Sensors that detect motion energy, skin tones, and texture energy feed the network at the frame level. Because it uses a Bayesian framework, the system can infer a concept's presence given information regarding another. Importantly, the authors refer to film's production codes when choosing semantic features to capture, and hint at their use for higher-level inferences—for example, the close-up effectively reveals character emotions, facilitating audience–character bonding, and is therefore a vital technique in romances and dramas.

Vasconcelos and Lippman present semantic concept timelines for two full-length movies.[32] Such a representation gives the user an immediate summary of the video, and lets the user interactively scrutinize the video for higher-level information. For example, the timeline might indicate that outdoor settings dominate the movie, but a user might want further detail—for example, What sort of setting, forest or desert?

Complex applications. Pfeiffer and Effelsberg combine many of the techniques previously discussed to perform a single complex task—to automatically generate movie trailers or abstracts.[33] An abstract, by definition, contains the essential elements of the thing it represents, hence the difficulty of the task. To create a trailer, the system must know the film's salient points. Moreover, it must create an entertaining trailer without revealing the story's ending.

Pfeiffer and Effelsberg's approach consists of three steps:

- *Video segmentation and analysis*, which attempt to discover structure, from shots to scenes, and other special events, such as gunfire or actor close-ups.

- *Clip selection*, which attempts to provide a balanced coverage of the material and any identified special events.

- *Clip assembly*, which must seamlessly meld the disjoint audio–visual clips into a final product.

If the units and structures that we want to index are author derived, they must be author sought.

The authors found that film directors consider constructing abstracts as an art, and abstracts differ depending on the data's genre. Feature film abstracts attempt to tease or thrill without revealing too much, documentary abstracts attempt to convey the essential content, and soap opera trailers highlight the week's most important events. Accordingly, the authors suggest that abstract formation be directed by parameters describing the abstract's purpose.

Wactlar et al. take a retrospective look at the Informedia project, another complex system embracing speech recognition, shot detection using optical flow for shot similarity, face and color detection for richer indexing, and likely text location and optical character recognition (OCR).[34] The Informedia project included the automatic generation of video skims, which are similar to Pfeiffer and Effelsberg's video abstracts,[35] but emphasize transmitting essential content with no thought of viewer motivation.

Video skim generation uses transcriptions generated by speech recognition. The authors' stated domain is broad and includes many hours of news and documentary video. Notably, they found that using such "wide-ranging video data," was "limiting rather than liberating." In other words, the system often lacked a sufficient basis for domain-guided heuristics. They go on to say that "segmentation will likely benefit from improved analysis of the video corpus, analysis of video structure, and application of cinematic rules of thumb."

Computational media aesthetics in MCM

Evaluating MCM approaches in general is difficult, and it's often exacerbated by small data sets. In particular, no standard test sets exist for automated video understanding, as they do for image databases and similar domains, against which developers can assess approaches for their relative strengths. The sheer number of as yet unenumerated problems of interest to the multimedia community, a direct result of the number of subdomains (such as video) exacerbates this problem. For example, unlike the shot extraction problem, which is fundamental to the entire domain and hence supported with standard test sets, the film subdomain brings with it a plethora of useful indexes, with many still to be identified.

A deeper cause for the difficulties in evaluating and comparing the results of different approaches relates to *schematic authority*, which should prompt questions such as Is this interpretive framework valid for the given data? Schematic authority is most appropriate to the class of problems that have been examined in this article, rather than consciously user-centric frameworks, which often involve iterative query and relevance feedback. In short, if the units and structures that we want to index are author derived, they must be author sought. Neither the researcher nor the end user can redefine a term at will if they want to maintain consistency, repeatability, and robustness.

Final thoughts

What does the CMA philosophy bring to this situation? Does systematic attention to domain distinctives, such as film grammar, address these issues? With regard to evaluation, CMA might more clearly define a baseline for comparison—that is, it may clarify the groundtruth source. To a small degree, CMA also alleviates the need for larger data sets. Film grammar embodies knowledge drawn from wide experience with the domain; it's the distillate of a very large data set indeed.

Film grammar also provides the reference point for deciding the most appropriate terminology from a number of options. For example, Is the scene an appropriate structure? What does it mean? Does a *strata* (a shot-based contextual description) properly belong to film, or is it a secondary term more suited to user-defined film media assessment? As for questions regarding the use of different feature sets, film grammar informs us of the many techniques available to the filmmaker that manifest differently, hinting that we may require multiple feature sets in different circumstances and at different times to more reliably capture the medium's full expressiveness. **MM**

References

1. C. Dorai and S. Venkatesh, "Computational Media Aesthetics: Finding Meaning Beautiful," *IEEE MultiMedia*, vol. 8, no. 4, Oct.–Dec. 2001, pp. 10-12.
2. R. Zhao and W.I. Grosky, "Negotiating The Semantic Gap: From Feature Maps to Semantic Landscapes,"*Pattern Recognition,* vol. 35, no. 3, Mar. 2002, pp. 51-58.
3. L. Rowe, J. Boreczky, and C. Eads, "Indexes for User Access to Large Video Databases," *Proc. Storage and Retrieval for Image and Video Databases,* The Int'l Soc. for Optical Eng. (SPIE), 1994, pp. 150-161.
4. M. Yeung, B.-L. Yeo, and B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation," *Proc. Int'l Conf. Multimedia Computing and Systems*, IEEE Press, 1996, pp. 296-305.
5. Y. Zhuang et al., "Adaptive Key Frame Extraction Using Unsupervised Clustering," *Proc. IEEE Int'l Conf. Image Processing*, IEEE Press, 1998, pp. 886-890.
6. A. Girgensohn, J. Boreczky, and L. Wilcox, "Keyframe-Based User Interfaces for Digital Video," *Computer*, vol. 34, no. 9, Sep. 2001, pp. 61-67.
7. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. Storage and Retrieval of Image and Video Databases II*, SPIE, 1994, pp. 2185-2205.
8. S. Chang and J. Smith, "Extracting Multidimensional Signal Features for Content-Based Visual Query, *SPIE Symp. Visual Comm. and Signal Processing*, SPIE, 1995, pp. 995-1006.
9. A. Hanjalic, R. Lagendijk, and J. Biemond, "Automated High-Level Movie Segmentation for Advanced Video-Retrieval Systems," *IEEE Trans. Circuits and Systems For Video Technology*, vol. 9, no. 4, June 1999, pp. 580-588.
10. J. Vendrig and M. Worring, "Evaluation of Logical Story Unit Segmentation in Video Sequences," *IEEE Int'l Conf. Multimedia and Expo 2001* (ICME 2001), IEEE CS Press, 2001, pp. 1092-1095.
11. L. Zhao, S.-Q. Yang, and B. Feng, "Video Scene Detection Using Slide Windows Method Based on Temporal Constraint Shot Similarity," *Proc. IEEE Int'l Conf. Multimedia and Expo 2001* (ICME 2001), IEEE CS Press, 2001, pp. 649-652.
12. E. Veneau, R. Ronfard, and P. Bouthemy, "From Video Shot Clustering to Sequence Segmentation," *IEEE Int'l Conf. Pattern Recognition*, vol. 4, IEEE Press, 2000, pp. 254-257.
13. Y. Rui, T.S. Huang, and S. Mehrotra, "Constructing Table-of-Content for Videos," *Multimedia Systems*, vol. 7, no. 5, 1999, pp. 359-368.
14. R. Hammoud, L. Chen, and D. Fontaine, "An Extensible Spatial–Temporal Model for Semantic Video Segmentation," *Proc. 1st Int'l Forum Multimedia and Image Processing*, 1998, http://citeseer.nj.nec.com/hammoud98extensible.html.
15. W. Mahdi, L. Chen, and D. Fontaine, "Improving the Spatial–Temporal Clue-based Segmentation by

the Use of Rhythm," *Proc. 2nd European Conf. Digital Libraries* (ECDL 98), Springer, 1998, pp. 169-181.

16. J. Huang, Z. Liu, and Y. Wang, "Integration of Audio and Visual Information for Content-Based Video Segmentation," *IEEE Int'l Conf. Image Processing* (ICIP 98), IEEE CS Press, 1998, pp. 526-530.
17. J. Kender and B.-L. Yeo, *Video Scene Segmentation via Continuous Video Coherence*, tech. report, IBM T.J. Watson Research Center, 1997.
18. H. Sundaram and S.-F. Chang, "Video Scene Segmentation Using Video and Audio Features," *Proc. Int'l Conf. Multimedia and Expo*, IEEE Press, 2000, pp. 1145-1148.
19. J. Vendrig, M. Worring, and A. Smeulders, "Model-Based Interactive Story Unit Segmentation," *IEEE Int'l Conf. Multimedia and Expo* (ICME 2001), IEEE CS Press, 2001, pp. 1084-1087.
20. B.T. Truong, S. Venkatesh, and C. Dorai, "Neighborhood Coherence and Edge-Based Approach for Scene Extraction in Films," *Proc. Int'l Conf. Pattern Recognition* (ICPR 02), IEEE Press, 2002.
21. J. Wang, T.-S. Chua, and L. Chen, "Cinematic-Based Model for Scene Boundary Detection," *Proc. Int'l Conf. Multimedia Modeling* (MMM 2001), 2001, http://www.cwi.nl/conferences/MMM01/pdf/wang.pdf.
22. J. Nam, M. Alghoniemy, and A. Tewfik, "Audio Visual Content-Based Violent Scene Characterization," *Proc. IEEE Int'l Conf. Image Processing* (ICIP 98), IEEE CS Press, 1998, pp. 353-357.
23. A. Yoshitaka et al., "Content-Based Retrieval of Video Data by the Grammar of Film," *IEEE Symp. Visual Languages*, IEEE CS Press, 1997, pp. 314-321.
24. C. Saraceno and R. Leonardi, "Identification of Story Units in Audio Visual Sequences by Joint Audio and Video Processing," *Proc. Int'l Conf. Image Processing* (ICIP 98), IEEE CS Press, 1998, pp. 363-367.
25. J. Huang et al., "Integration of Multimodal Features for Video Classification Based on HMM," *Proc. Int'l Workshop on Multimedia Signal Processing*, IEEE Press, 1999, pp. 53-58.
26. A. Alatan, A. Akansu, and W. Wolf, "Multimodal Dialogue Scene Detection Using Hidden Markov Models for Content-Based Multimedia Indexing," *Multimedia Tools and Applications*, vol. 14, 2001, pp. 137-151.
27. M. Naphade and T.S. Huang, "A Probabilistic Framework for Semantic Video Indexing, Filtering, and Retrieval," *IEEE Trans. Multimedia*, vol. 3, no. 1, Jan. 2001, pp. 141-151.
28. V. Roth, "Content-Based Retrieval from Digital Video," *Proc. Image and Vision Computing*, vol. 17, Elsevier, 1999, pp. 531-540.
29. S. Fischer, R. Lienhart, and W. Effelsberg, *Automatic Recognition of Film Genres*, tech. report, Univ. of Mannheim, Germany, 1995.
30. E. Sahouria and A. Zakhor, "Content Analysis of Video Using Principal Components," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 9, no. 8, Dec. 1999, pp. 1290-1298.
31. N. Vasconcelos and A. Lippman, "Toward Semantically Meaningful Feature Spaces for the Characterization of Video Content," *Proc. Int'l Conf. Image Processing* (ICIP 97), IEEE CS Press, 1997, pp. 25-28.
32. N. Vasconcelos and A. Lippman, "Bayesian Modeling of Video Editing and Structure: Semantic Features for Video Summarization and Browsing," *Proc. Int'l Conf. Image Processing* (ICIP 98), IEEE CS Press, 1998, pp. 153-157.
33. R.L.S. Pfeiffer and W. Effelsberg, "Video Abstracting," *Comm. ACM*, vol. 40, no. 12, Dec. 1997, pp. 54-63.
34. H. Wactlar et al., "Lessons Learned from Building a Terabyte Digital Video Library," *Computer*, vol. 32, no. 2, Feb. 1999, pp. 66-73.
35. B. Adams, C. Dorai, and S. Venkatesh, "Toward Automatic Extraction of Expressive Elements from Motion Pictures: Tempo," *IEEE Trans. Multimedia*, vol. 4, no. 4, Dec. 2002, pp. 472-481.

Brett Adams received a PhD from the Curtin University of Technology, Perth. His research interests include systems and tools for multimedia content creation and retrieval, with a particular emphasis on mining multimedia data for meaning. Adams has a BE degree in information technology from the University of Western Australia, Perth, Australia.

Readers may contact Brett Adams at adamsb@cs.curtin.edu.au.