

MediaTable: Interactive Categorization of Multimedia Collections

Ork de Rooij and Marcel Worring ■ *University of Amsterdam*

Jarke J. van Wijk ■ *Eindhoven University of Technology*

Digital multimedia archives from various providers, including broadcasters, news agencies, and social-media sites such as Flickr and YouTube, are growing rapidly. However, the tools for managing such multimedia collections can't keep up. So, the content is often irretrievable at a later stage.

MediaTable helps users categorize image or video collections. A tabular interface gives an overview of multimedia items and associated metadata, and a bucket list lets users quickly categorize materials. MediaTable uses familiar interface techniques for sorting, filtering, selection, and visualization.

We've designed a tool that lets users quickly view and categorize large multimedia collections. *MediaTable* combines automatic content-analysis techniques with an interface optimized for visually categorizing large sets of results.

The Challenge of Multimedia Categorization

Typical scenarios for multimedia categorization include

- sorting images from a camera's memory card into good and bad pictures;
- labeling old photos with the names of the people in them; or
- screening news, Web, or surveillance data for terrorist activity.

A more detailed example comes from digital forensics. For example, police inspectors might need to determine whether a suspect's computer stores child pornography. They must therefore view ev-

ery photo or fragment of video on all drives of the suspect's computer. These inspectors are often forensics experts, who can categorize images immediately upon seeing them. Unfortunately, visually inspecting every fragment of multimedia is impractical and labor intensive.

Metadata-based retrieval engines, such as those used by Flickr, YouTube, and Google Video, are the most common means of accessing a multimedia collection. The metadata provided with the collection can include file names, the type of codec used, video length, titles and descriptions, or social tags. Because most of this metadata must be manually entered, the textual descriptions don't necessarily capture the multimedia's visual content adequately, either because the annotation is too sparse or because the annotators focused on different aspects of the multimedia.

To assess multimedia, we must look at it on all levels—from the collection level, to the file level with associated metadata, to the individual frames or images themselves. Automatic content-analysis techniques provide metadata based on the multimedia's visual content. This content-based metadata in turn helps us understand and organize the multimedia on the basis of the content. (See the "Multimedia Content Analysis" sidebar for more information.) However, unlike a regular metadata search, content-based image retrieval (CBIR) techniques often produce far from perfect results. In addition, previous multimedia categorization tools haven't effectively overcome this inaccuracy.

Multimedia Content Analysis

Behind content-based image retrieval lies a series of content-analysis algorithms.¹⁻³ In the early years, content analysis was based on low-level feature comparisons, and many of these systems required specialized user input,² such as sketching or providing example images. Providing this input isn't always possible or practical. Moreover, the low-level visual-feature representation used for querying often doesn't correspond to the user's intent, a problem called the *semantic gap*.²

Researchers have recently proposed and examined several solutions to this problem,³ with varying performance. One such solution is generic concept detection that allows automatic labeling of people, objects, settings, or events in video content. Such algorithms first segment multimedia into individual *shots*—fragments of video from a single camera capture and extract low-level features from their frames. For images, extra segmentation is not needed, and the algorithms can directly extract low-level features.

On the basis of these extracted features, a supervised machine-learning algorithm determines the presence of a certain semantic concept. This yields scores between

0 and 1 indicating the system's view on that concept's absence or presence in each shot in the collection. For example, to provide these scores, MediaTable (see the main article) uses the MediaMill Semantic Video Search Engine, but other video-analysis systems are available.

These algorithms produce a new form of metadata that can characterize individual shots. However, because automated detection's quality varies, humans must still visually inspect all possible results. So, MediaTable combines automated analysis with visual inspection.

References

1. R. Datta et al., "Image Retrieval: Ideas, Influences, and Trends of the New Age," *ACM Computing Surveys*, vol. 40, no. 2, 2008, pp. 1–60.
2. A.W.M. Smeulders et al., "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Dec. 2000, pp. 1349–1380.
3. C.G.M. Snoek and M. Worring, "Concept-Based Video Retrieval," *Foundations and Trends in Information Retrieval*, vol. 4, no. 2, 2009, pp. 215–322.

Most successful interactive interfaces deal with the inaccuracy of CBIR by letting users quickly browse through hundreds of results. For example, Alex Hauptmann and his colleagues used rapid serial visual presentation to allow categorization of automatic results organized in batches of two to nine images.¹ Mike Christel and Rong Yan used a storyboard-based user interface particularly suited to categorize several shots in one video.² Eric Zavesky and his colleagues used a grid-based structured visualization of a ranked list, in which the presentation of the images in each grid allows quick recognition of categories.³ Other visualizations have used spatial placement of images to convey information. For example, Jing Yang and his colleagues used multidimensional scaling to organize images on the basis of visual aspects.⁴

As an alternative to browsing, other systems, such as IBM Marvel, allow tagging of results.⁵ Users can add large sets of relevant concepts, or tags, to single items, instead of assigning a single concept to a large set of items. All these systems aim to aid video retrieval, so they're less suited for categorizing collections or gaining insight into them. Furthermore, they sometimes use radically novel user interfaces, which inhibits novice users from understanding them.

Researchers have developed many categorization methods for enormous collections of abstract multivariate data—a classic problem in information

visualization. Examples include scatterplots, dense pixel displays, parallel coordinates, treemaps, and glyphs. Various applications use these methods to organize or gain insight into multimedia collections. For example, PhotoMesa uses a treemap to visualize an insightful hierarchy, one multimedia aspect at a time.⁶

Tables are a classic, useful approach. They are familiar, allow the display of many attributes, and enable easy interaction. The spreadsheet is a well-known table-based application. Jacques Bertin showed that reordering columns and rows can lead to more insight.⁷ One application in the multimedia domain, PhotoSpread, lets users drag and drop images into table cells to categorize them.⁸ For larger tables that don't fit in a single screen, scrolling can let users browse through all the available information. However, these techniques can result in the loss of visual relations between parts of the collection. A solution is the table lens, which makes individual items in large tables clearly visible, while maintaining context by showing condensed versions of many other columns and rows.⁹ The Focus¹⁰ and InfoZoom (<http://infozoom.com>) systems extend this idea. In such systems, users can interact with entire sets of summarized data at once. For multimedia, however, this poses a problem. You can categorize an individual image by looking at it, but no direct means of summarizing hundreds of images exists.

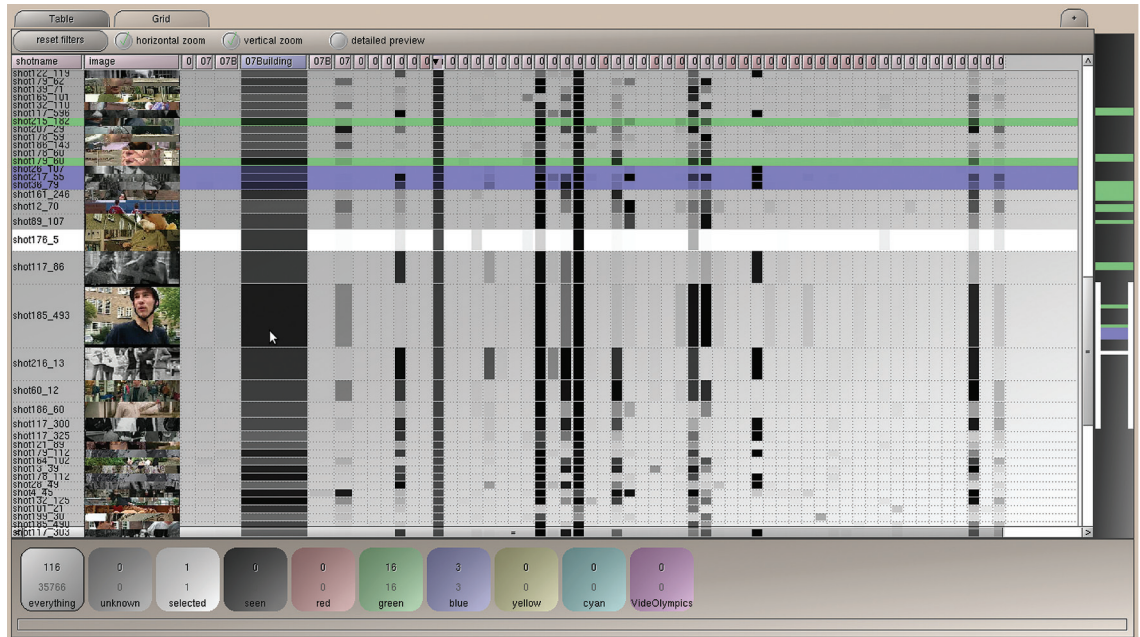


Figure 1. The MediaTable interface. The table interface takes up most of the figure, with the overview pane to the far right and the bucket list on the bottom. The table shows a collection of 200 hours of video, with a part expanded through the lens effect. The series of filled cells indicate each detected concept’s presence or absence in each video fragment. The colors in the table and in the overview indicate in which buckets each fragment is currently categorized.

Effective multimedia categorization requires a tool that combines smart, recognizable interface elements with content-based metadata. Such a tool must give users insight into the collection and let them easily add categorizations to the collection using that insight.

Those basic requirements led us to establish four more requirements. First, the tool must allow and enforce visual inspection of results to compensate for the varying quality of content-based metadata. Second, it must provide overview and detail, as well as a seamless zoom between them, so that users can inspect both individual items and large collections. Third, it must allow for categorization of fragments in a collection. Finally, it must be easy to use for both expert and nonexpert users.

Categorizing a Multimedia Collection

To meet those requirements, MediaTable exploits techniques from metadata-based retrieval, CBIR, and existing multivariate-visualization techniques. As Figure 1 shows, MediaTable provides different views of the same collection. Its primary components are the *table interface*, *overview pane*, *bucket list*, and *point cloud viewer*.

The Table Interface

This interface is MediaTable’s main interaction component. Initially, it displays all the media items in the collection. Each row yields information

about an individual video shot, with a representative frame. To fit each frame in a row, MediaTable cuts off the top and bottom parts. Because the most important information is usually centered in the image, users can still recognize most scenes even though half of the frame is missing.

The depicted columns vary depending on user search need, but they typically contain metadata based on both user-supplied information and results from automated CBIR. Users can interact with the table in several ways. They can sort or filter the collection by individual columns or rows. They can also order table rows by visual similarity to the image in the chosen row, which is placed on top. Finally, they can select one or more rows by clicking, dragging, or shift-clicking, and add those rows to buckets.

Each table cell represents one piece of metadata for a shot. MediaTable supports the following four representations, depending on the input data.

Text. A pure-text field can, for example, display file names or other textual metadata.

Dots. A colored dot’s presence or absence in the cell’s center can represent Boolean values. The dot’s color can represent a series of nominal values.

Fill. A cell’s fill can range from fully transparent to opaque, depending on the data’s numeric value. Filled cells are a powerful tool when used

with concept-detection results. In most cases, a video shot contains few semantic concepts. With the filled-cell visualization, this will be obvious because the cells containing concepts will be the only ones that aren't completely transparent. When many concepts appear in a table, user attention will be immediately drawn to those concepts in the shot. Furthermore, results obtained from semantic concept detection contain some uncertainty. The filled-cell representation mitigates this uncertainty by letting users focus on visual differences between scores instead of exact numerical values.

Images. A cell can display an image. An expert can determine much information with a single glance at an image. By exploiting this ability, we help the expert with classification.

Furthermore, the table allows four types of interactions that let users see individual images more clearly. The first is the *row lens effect*. When the mouse hovers over a row, MediaTable slightly enlarges the row, similarly to the table lens effect.¹¹ This lets the user view the frame in full detail.

The second type is *column condensation*. When a table contains many columns, MediaTable compacts them into one visual layout, so that correlations between columns are visible. The condensation hides details such as the column names, which aren't needed at that time, but still lets users see the column contents visualized as fills or dots. When the mouse hovers over a column, MediaTable seamlessly enlarges the column to show details.

The third type is *mouse-over previews*. When the mouse hovers over a row, a preview window appears near the mouse. This window shows either an enlarged version of the selected item or, for video, a window that repeatedly plays the entire shot at faster-than-real-time speeds. This gives the user an impression of the video content in each shot in the collection.

The fourth type is a *grid preview*. To support categorizing small sets on a detailed scale, MediaTable can also display items in a grid formation (see Figure 2a). This hides all extra metadata and lets users focus on the images themselves. With the grid preview, all categorization-based interactions are also available. Furthermore, the user can interactively change the shown frames' size. In addition, mouse-over previews are available.

Because users have varying preferences for user interfaces, these interactions are on-demand only; users can switch them on or off. For example, for some users the row lens effect gives enough detail of the image, while others might prefer the mouse-over preview.

The Overview Pane

The table shows only part of the collection. The overview pane displays the entire collection and acts as a scroll bar for the table. It gives users an overview of where they are in the collection, including what they've categorized so far. Currently, categorized items are also highlighted in the overview pane using different colors. When items are placed in multiple categories or when there are more items than one pixel can represent, the overview pane shows a color mixture.

The tool must allow and enforce visual inspection of results to compensate for the varying quality of content-based metadata.

The overview pane's most important role is to give a visual representation of seen and categorized parts and as-of-yet unseen parts in the collection. This lets users visually estimate whether they've examined a collection closely enough or it merits further examination. For excessively large collections, the overview can also help users estimate how many human-hours might be required to accurately categorize the entire collection.

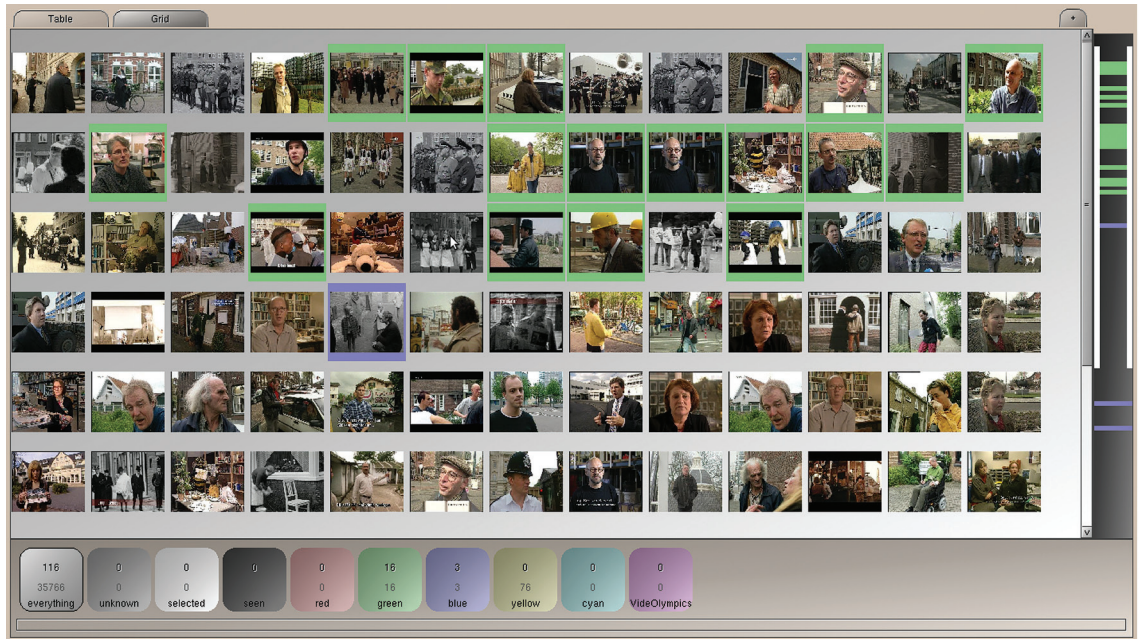
The Bucket List

This list forms the basis for categorization-based browsing. Users categorize items by placing them into buckets, so a bucket can be seen as a category. Users can place items in multiple categories by putting them in multiple buckets.

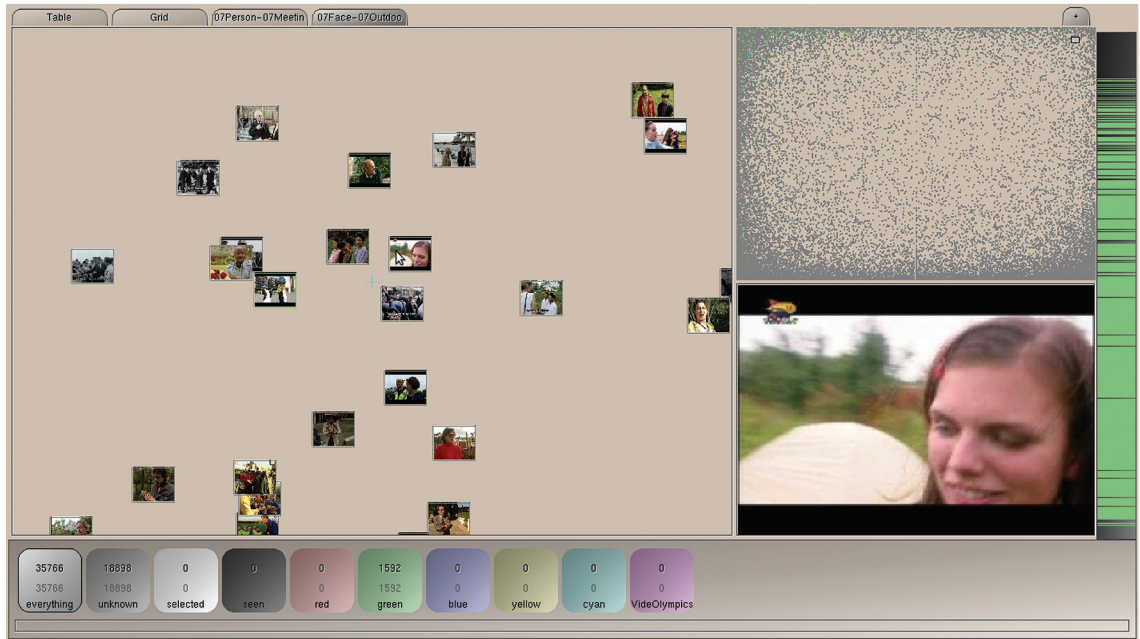
We distinguish between *special buckets* and *user buckets*. Special buckets automatically update their contents; users manually name and fill user buckets. The number of user buckets can vary; in our prototype, we fixed the number at six.

There are five types of buckets:

- The *everything bucket* contains all media items in the collection.
- The *unseen bucket* contains all uncategorized media items in the collection.
- The *seen bucket* contains every media item that has ever appeared in the detail pane.
- The *selected bucket* contains only the current selection.
- *User-specified buckets* must be explicitly filled by users. Users can have as many buckets as they require.



(a)



(b)

Figure 2. Two available views in MediaTable. (a) The grid preview for detailed inspection of part of the table. The colored borders indicate to which bucket a media item is assigned. (b) The point cloud viewer for detailed inspection of two columns. The scatterplot lets users seamlessly zoom in and out anywhere in the plot, from an overview of every item up to an individual image. The top-right corner shows the entire space with all items; the bottom-right corner shows the image under the mouse cursor. At any zoom level, the user can drag-select batches of results to place in any bucket.

Seen, selected, and user-specified buckets start out empty. User buckets can contain any selection of items, and individual items can be in multiple user buckets. Users add items by selecting them in any other view and assigning them to a bucket using a right-click context menu or a keyboard shortcut. This creates the colored fills in the table interface and the colored backgrounds in the grid

preview and the point cloud viewer. The colors are chosen such that items placed in multiple buckets will often still have a unique color.

Clicking a bucket shows its contents in the table, with all other media items filtered out. By right-clicking on a bucket, users access a context menu that lets them show or hide the contents of one or more buckets simultaneously, empty buckets, or

load lists of media items from disk or save them to disk.

The Point Cloud Viewer

When users see interesting correlations between two columns in the table, they can open the point cloud viewer for a more detailed inspection. MediaTable then depicts the relations between the selected columns in a scatterplot of images, similarly to Yang and his colleagues' work.¹² To keep a contextual mapping between the table and the scatterplot, we map each axis directly to a column, rather than using a similarity-based visualization in which axes have no meaning. Users can therefore map spatial coordinates with column-sort directions.

Furthermore, to ensure that we don't lose overview and detail, the point cloud viewer allows seamless zooming from a full overview of every available result in the collection to an individual image (see Figure 2b). The bucket list is still visible, and users can select regions of results and add them to any bucket.

Bucket-Based Workflow

Before categorization is possible, the user must define a set of buckets. For example, in a forensic investigation such as we mentioned earlier, categories could include "illegal material," "suspect present," "children present," "nude present," and "harmless." The categorization needn't be exclusive (users can place media items in multiple buckets), but it must be complete. That is, every shot must fit into at least one bucket. So, typically MediaTable also provides a default or "not relevant" bucket.

Looking at just one bucket, we can think of the entire collection as relevant or irrelevant to it. Thus, we can view filling a bucket as a binary-classification problem. So, we define three actions.

First, users can explicitly categorize relevant items. If a sorting or filtering method is available that can rank items' relevance to a bucket, users can batch-select relevant items and put them in the bucket.

Second, users can explicitly categorize irrelevant items. A collection might contain many items that are clearly visually irrelevant to the bucket. Using sorting and filtering techniques to first capture these items helps because the remaining items will be less cluttered and potentially more relevant. For example, a forensics expert can first sort and batch selection results that aren't suspect, by filtering out frequently occurring material such as cartoons. The expert can place these items in a "harmless" bucket. Eventually, the unseen bucket will contain only potentially harmful materials.

Finally, users can employ intermediate categorization. For example, when both relevant and irrelevant items share a characteristic, users can sort by this characteristic and place the entire set of those items in a temporary bucket. By selecting that bucket, the user can determine whether another characteristic would help sort the relevant from the irrelevant media items in that bucket only. If so, the user can select the relevant items and add them to the target bucket.

Users can choose which bucket to fill first, and they don't need to go through the entire collection to fill one bucket before filling others.

Users can choose which bucket to fill first, and they don't need to go through the entire collection to fill one bucket before filling others. Typically, users will explore available metadata columns by first finding and then sorting on the column that provides them with the most distinctive segmentation of items for any set of buckets. Then users can select large batches of mostly consecutive items at once and place them in buckets. They use the contents of these buckets as inclusive or exclusive filters while sorting on other columns, and then continue the categorization process from there. Figure 3 gives an overview of this scenario.

For example, by sorting on cityscape (see Figure 4), a user sees a list of images showing cityscapes in decreasing order of probability. By selecting all rows up to the point at which the images no longer represent cityscapes, the user can create a bucket that classifies the collection into "contains cityscape yes/no." The user might further categorize the items in this bucket by selecting the bucket and sorting on "people visible." Alternatively, users can look at all items they didn't place in the "cityscape" bucket (if they'd identified "cityscape" as irrelevant) and determine a segmentation for that data. This process continues, each step significantly decreasing the number of items until all items are in one or more buckets.

Depending on the task, the user can choose between several visualizations. The table interface is ideal for collection overview and for discovering new relevant concepts to aid collection categorization. The point cloud viewer lets users quickly select large correlated subsets of the collection for more detailed analysis later, and the grid preview

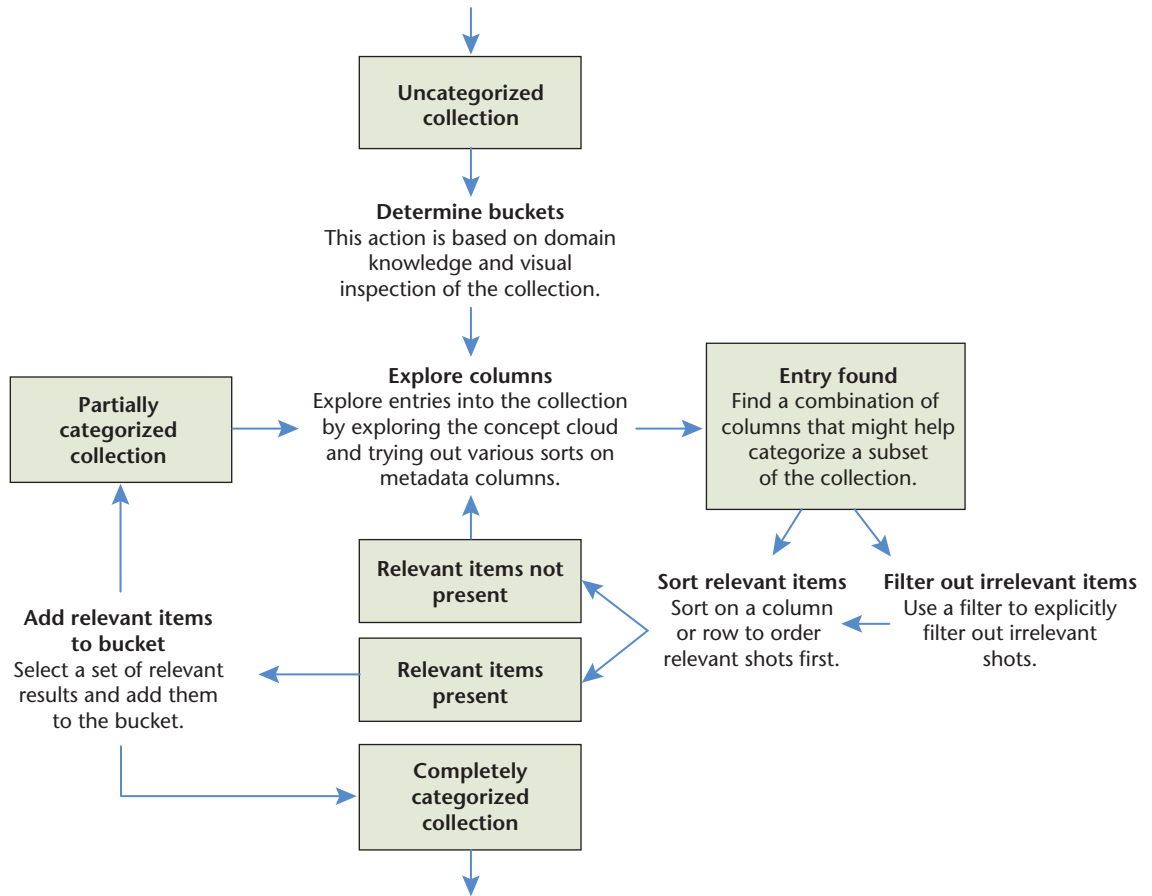


Figure 3. A typical workflow for collection categorization using MediaTable. First, the user explores the collection briefly to find relevant buckets. Entry points to the collection yield sorted lists of items that would all fall in the same bucket. The user selects these and places them in a bucket, which is rendered into the overview. The user continues looking for entry points in the remainder of the collection.

helps with categorizing small sets of filtered results. For example, users can quickly tag all faces into a temporary bucket using the point cloud viewer and then use the grid preview for detailed inspection of the resulting selection.

During categorization, the overview pane gives a complete overview of the bucket distribution in the entire collection or the current selection. The bucket list shows the actual number of elements in each bucket.

MediaTable Implementation

We designed and built MediaTable from scratch, so issues and bugs are bound to surface. In addition, an interface’s original design might not be the most efficient, and it’s better to allow for evolutionary design.

So, we built a preliminary version of MediaTable, which we tested during several design cycles with different user groups. This testing yielded a basic set of recommendations for the table interface and bucket list. Later, we set up a bug-reporting tool and gave a group of 24 computer science students access to it to help them solve a multimedia data-

mining project as part of a lab course. During this course, we listened to their suggestions and bug reports, and provided incremental releases of MediaTable to solve outstanding issues.

User Studies

We performed a small-scale user study with digital-forensics experts on a dataset taken from home video. We also performed two larger-scale studies with a group of students: one on a collection of Flickr images and the other on a dataset taken from broadcast video.

Each study followed a similar strategy. First, we introduced the participants to MediaTable and gave them a categorization task to apply to the collection. They had limited time to solve this problem and some time to ask questions about the system. Next, we gave them more categorization tasks. During each task, we logged all user actions and the categorization results.

Home Video

The five digital-forensics experts design large-scale forensic image-categorization software used by sev-

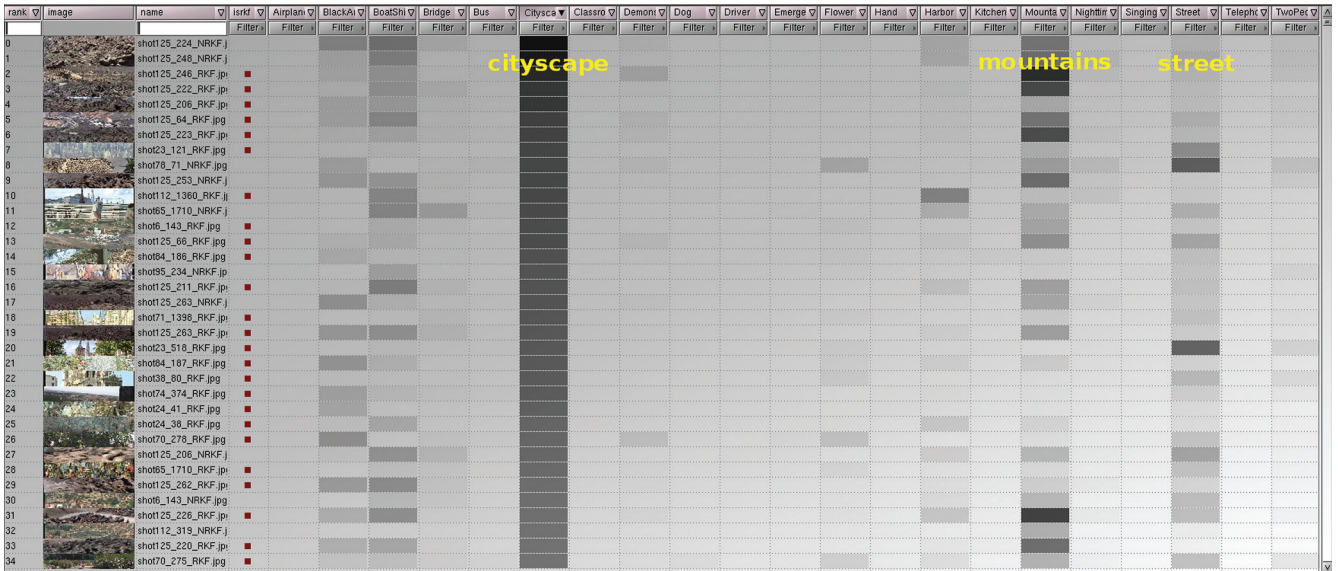


Figure 4. The MediaTable table sorted on “cityscape” in the 200-hour home video collection. The visualization shows a correlation between “cityscape” and “mountains” and, to a lesser extent, “street,” “boats and ships,” and “black and white.”

eral police forces in the Netherlands. For this study, we used a collection of 200 hours of home video recordings from various individuals, together with various television recordings. A series of 21 generic semantic concepts based on automated video analysis served as metadata to this collection. Participants couldn’t access the point cloud viewer.

After the participants familiarized themselves with MediaTable, we had them perform two categorization tasks aimed at letting them find and use relations between concepts. Results indicate that explicitly visualizing a series of at-first-unrelated concepts in a table let the participants find concept relations interactively, which they then used to help categorize the collection. Furthermore, all participants showed a usage pattern similar to the bucket-based categorization workflow we described earlier. First, they explored the top set of items found by individual concepts. They started categorizing when they found a set of items resembling shots they expected to find for the task.

Owing to this early prototype’s limitations, further examination of the results was limited. The participants did, however, show enthusiasm about the possible ways in which MediaTable allowed interaction with video collections and how this could help users explore and categorize large video collections.

Broadcast Video

This study involved 12 teams of two participants to provide insight into individual-component use. Although MediaTable is essentially a multicategory categorization tool, we expected to gain the most insight by keeping the tasks relatively simple. So, we used only binary categorization tasks. Because

we wanted to measure bucket use, we instructed the participants to use as many buckets for storing intermediate results as they saw fit.

We asked each participant to perform four tasks, with the other team member present to help or guide. Table 1 lists the tasks. We used a dataset of 200 hours of video, split into 35,766 shots, with 57 associated metadata columns with semantic annotations ranging from “wildlife” to “face” to “people marching,” all automatically extracted on the basis of content analysis.

All 12 teams started the same task simultaneously; each task ended after exactly five minutes. To ensure active, serious participation, we introduced a game element, similar to the VideOlympics,¹³ by projecting each team’s resulting categorization in real time on a large projection screen. The participants could use the table interface, grid preview, and point cloud viewer in conjunction with the bucket list to store categorizations.

Despite the difficult tasks and the limited time frame, participants found a significant portion of the available items (see Table 1), showing the interface’s effectiveness. Consider, for example, task 7 in Table 1. Here, three participants used grid- and table-based browsing alone and didn’t need any other buckets to store intermediate information. Nine participants used other buckets and achieved on average 123 percent more relevant results.

To understand how participants used the point cloud viewer and the grid, we focus on three participants (see Figure 5). Participant 5 shows the typical pattern of someone who doesn’t use other buckets or visualizations. After 300 seconds (5 min.), participant 5 had retrieved 48 relevant items.

Table 1. Descriptions and high-level results for the categorization tasks for broadcast video.*

Task no.	Images to categorize	No. of relevant items	Using a single bucket		Using multiple buckets	
			No. of participants	Avg. no. of items found, with std. deviation	No. of participants	Avg. no. of items found, with std. deviation
1	People with mostly trees and plants, and no buildings	454	7	80 ± 36	5	40 ± 23
2	People playing with children	210	5	20 ± 11	7	31 ± 36
3	Pieces of paper with writing	634	6	156 ± 144	6	102 ± 82
4	Food or drinks on the table	94	7	19 ± 16	5	24 ± 5
5	Interviewed woman talking to the camera	429	5	55 ± 41	7	71 ± 63
6	A geographical map	154	7	68 ± 20	5	54 ± 43
7	People with a body of water visible	274	3	44 ± 26	9	98 ± 35
8	A face filling more than half of the frame	963	3	155 ± 55	9	96 ± 45

*For each task (except task 3), the bold number indicates which technique (single or multiple buckets) was used most. In most cases, the technique that was used most found the most items.

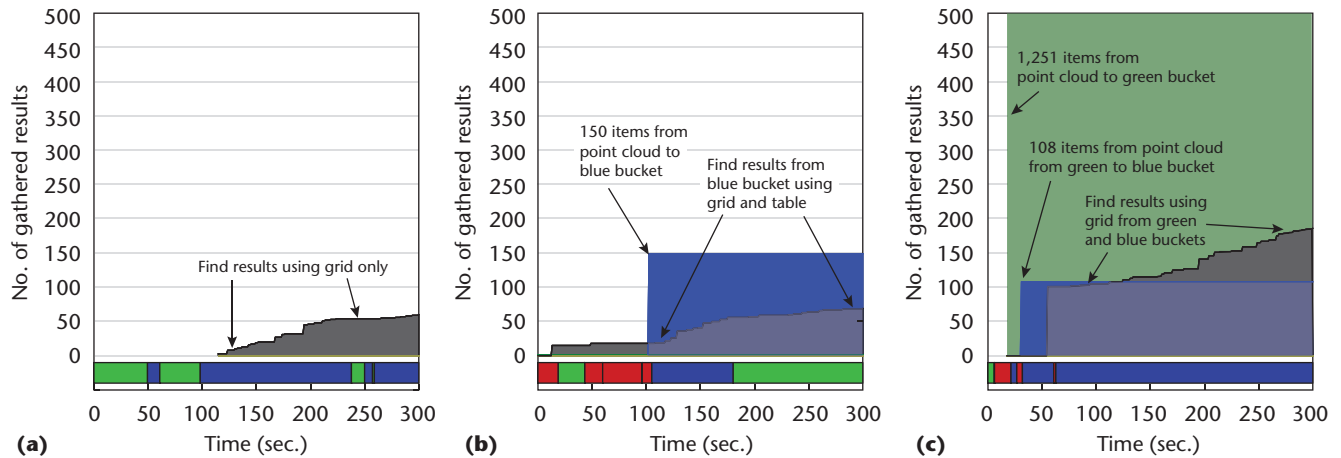


Figure 5. The results for a broadcast-video categorization task for participants (a) 5, (b) 7, and (c) 8. Each graph shows the number of gathered items for each bucket over time. The green and blue areas indicate use of temporary buckets. The black graph indicates the number of items that participants submitted as “final and positive.” Below each graph is a bar indicating which visualization the participants used at each point in time (green: table; blue: grid; and red: point cloud). The annotations indicate what the participants did at certain points in time.

Participants 7 and 8 show a more advanced usage pattern. Both used the point cloud viewer to seed an initial bucket with items and then browsed through those items using the grid preview (participant 8) or table interface and grid preview (participant 7) to find relevant items in only that subset. Participant 7 used other buckets but only when the items had a high chance of being relevant. Using additional buckets took a lot of time; after five minutes, participant 7 had found only 59 relevant items. Participant 8 chose to explicitly fill the bucket with 1,251 items and then search through them only. Although only approximately 15 percent of the green temporary bucket was relevant, this strategy reduced the number of items to search such that participant 8 still had time to gather 130 relevant items.

The overall results for all tasks reveal a similar pattern. For four categorization tasks, most participants used multiple buckets. Those who did found on average 58 percent more relevant items. Results of the four other tasks indicate that participants using single buckets obtained on average 38 percent more relevant items than those using extra buckets. In these cases, most participants used only a single, task-suited metadata column, such as using “face” for task 8. So, they didn’t need other buckets. When participants used multiple buckets, we saw more advanced segmentation patterns, similar to those in Figure 5, with users first segmenting the dataset into smaller task-suited chunks and then selecting relevant items from the most relevant chunk only. Overall results indicate that bucket-based browsing enhances multimedia categorization.

MediaTable lets users search through a collection without losing the connection to the dataset. It shows results of multiple content-analysis algorithms simultaneously, which helps user see patterns in the data. The bucket list summarizes the search process, gives users a temporary working memory, and provides a way to classify results without losing the original collection's context. Our future work includes further investigation of emergent user-segmentation patterns and investigation into automating MediaTable on the basis of relevance feedback from user interaction. ■■

Acknowledgments

We acknowledge the ZiuZ company for their support and comments during the evaluation.


References

1. A.G. Hauptmann et al., "Extreme Video Retrieval: Joint Maximization of Human and Computer Performance," *Proc. 14th Ann. ACM Int'l Conf. Multimedia*, ACM Press, 2006, pp. 385–394.
2. M.G. Christel and R. Yan, "Merging Storyboard Strategies and Automatic Retrieval for Improving Interactive Video Search," *Proc. 6th ACM Int'l Conf. Image and Video Retrieval*, ACM Press, 2007, pp. 486–493.
3. E. Zavesky, S.-F. Chang, and C.-C. Yang, "Visual Islands: Intuitive Browsing of Visual Search Results," *Proc. 2008 Int'l Conf. Content-Based Image and Video Retrieval (CIVR 08)*, ACM Press, 2008, pp. 617–626.
4. J. Yang et al., "Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis," *Proc. 2006 IEEE Symp. Visual Analytics Science and Technology (VAST 06)*, IEEE Press, 2006, pp. 191–198.
5. R. Yan, A. Natsev, and M. Campbell, "An Efficient Manual Image Annotation Approach Based on Tagging and Browsing," *Proc. Workshop Multimedia Information Retrieval on the Many Faces of Multimedia Semantics (MS 07)*, ACM Press, 2007, pp. 13–20.
6. B.B. Bederson, "PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps," *Proc. 14th Ann. ACM Symp. User Interface Software and Technology (UIST 08)*, ACM Press, 2001, pp. 71–80.
7. J. Bertin, *Semiology of Graphics*, Univ. of Wisconsin Press, 1983.
8. S. Kandel et al., "PhotoSpread: A Spreadsheet for Managing Photos," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 08)*, ACM Press, 2007, pp. 1749–1758.
9. R. Rao and S.K. Card, "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 94)*, ACM Press, 1994, pp. 318–322.
10. M. Spenke, C. Beilken, and T. Berlage, "Focus: The Interactive Table for Product Comparison and Selection," *Proc. 9th Ann. Symp. User Interface Software and Technology (UIST 96)*, ACM Press, 1996, pp. 41–50.
11. R. Rao and S.K. Card, "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 94)*, ACM Press, 1994, pp. 318–322.
12. J. Yang et al., "Semantic Image Browser: Bridging Information Visualization with Automated Intelligent Image Analysis," *Proc. 2006 IEEE Symp. Visual Analytics Science and Technology (VAST 06)*, IEEE Press, 2006, pp. 191–198.
13. C.G.M. Snoek et al., "VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems," *IEEE MultiMedia*, vol. 15, no. 1, 2008, pp. 86–91.

Ork de Rooij is a PhD student in computer science at the University of Amsterdam, where he works on advanced visualization systems for large video collections. His main research interests cover multimedia information visualization, user computer interaction, and large-scale content-based multimedia retrieval systems. Contact him at orooij@uva.nl

Marcel Worring is an associate professor in the University Amsterdam's Intelligent Systems Lab Amsterdam. His research interests are in multimedia analytics, particularly the integration of computer vision, machine learning, and information visualization. Worring has a PhD from the University of Amsterdam. Contact him at m.worring@uva.nl

Jarke J. van Wijk is a full professor in visualization at the Technische Universiteit Eindhoven. His main research interests are information visualization and flow visualization, focusing on the development of new visual representations. Van Wijk has a PhD in computer science from Delft University of Technology, with honors. Contact him at vanwijk@win.tue.nl

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.