Constant Density Displays Using Diversity Sampling

Mark Derthick[1], Michael G. Christel[12], Alexander G. Hauptmann[2], Howard D. Wactlar[2]
[1]Human-Computer Interaction Institute
[2]Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
+1 412 268-8812
{mad, christel, hauptmann, wactlar}@cs.cmu.edu

Abstract

The Informedia Digital Video Library user interface summarizes query results with a collage of representative keyframes. We present a user study in which keyframe occlusion caused difficulties. To use the screen space most efficiently to display images, both occlusion and wasted whitespace should be minimized. Thus optimal choices will tend toward constant density displays. However, previous constant density algorithms are based on global density, which leads to occlusion and empty space if the density is not uniform. We introduce an algorithm that considers the layout of individual objects and avoids occlusion altogether. Efficiency concerns are important for dynamic summaries of the Informedia Digital Video Library, which has hundreds of thousands of shots. Posting multiple queries that take into account parameters of the visualization as well as the original query reduces the amount of work required. This greedy algorithm is then compared to an optimal one. The approach is also applicable to visualizations containing complex graphical objects other than images, such as text, icons, or trees..

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces---Screen design, Evaluation/methodology; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems---Video; H.3.7 [Information Storage and Retrieval]: Digital Libraries---User issues.

General Terms: Algorithms, Human Factors.

Keywords: Information Visualization, Collage

1. Motivation

The goal of Exploratory Data Analysis is to maximize insight into a dataset. Interactive graphics are the most effective way to satisfy this goal [NIST/SEMATECH 2002]. Box plots and histograms are often used to provide quick overviews of quantitative data. However they don't help with unstructured variables such as text, images, and video. For these variables, individual records must be attended to serially [Treisman and Kanwisher 1998]. Scatterplots can be effective, but to maximize the information conveyed, both occlusion and empty areas should be avoided. Figure 1 illustrates both problems.

IEEE Symposium on Information Visualization 2003,
October 19-21, 2003, Seattle, Washington, USA
0-7803-8154-8/03/$17.00 ©2003 IEEE

As the Informedia Digital Video Library (IDVL) grew to include tens of thousands of news stories and hundreds of thousands of individual shots, it became important to structure and summarize query results. Video search engines typically represent video as a sequence of shots, each represented by a keyframe image [Zhang et al. 1995]. (A shot is a continuous sequence from one camera.) Therefore, our goal becomes effective image retrieval interfaces.

Compared to web or other text-based information retrieval algorithms, image retrieval algorithms have much lower precision, so a user must scan many more results to find the same number of useful documents. However there is an advantage to images, in that people can scan them quickly.

Several IDVL visualizations were previously developed that show dozens of result images at once, organized on maps, timelines (see Figure 1), or in a conceptual space defined by the search terms [Wactlar 2001]. Using drill-down or Dynamic Query [Ahlberg et al. 1992], users can focus on a subset of the results. For instance, rather than all 2001 stories, one could focus on March 2001. This allows more March sample results to be shown. The resulting visualizations are called "collages" because they are conceptually like the art form where representative objects are laid out in space to convey an overall message [Christel et al. 2002]. The idea is not to just show the top *n* results, but to convey a summary of all the results.

Summarizing video intelligently is beyond the current state of the art in image understanding and in topic detection and tracking (TDT). The IDVL selection algorithm simply chooses the highest-scoring results for each time interval (on timelines) or for each country (on maps). This ensures that even for countries that do not play a prominent role in a topic, the visualization will suggest the most important role that it does play. This is stratified sampling without accounting for sample weights.

In this paper, we generalize these algorithms to work with any visualization, using diversity sampling in the spatial dimensions of the visualization. Diversity (or heterogeneity) sampling is a non-probability sampling technique to get a broad spectrum of values [Xie et al. 2003]. It is like a random sample of the values, rather than of the population having those values. Since diversity sampling loses distribution information, it is crucial to complement visualizations of diversity samples with those of probability samples like box plots and histograms.

A greedy sampling algorithm is used. Conceptually, the query results are scanned in preference order, and added to the visualization as long as no occlusion is introduced. However sequential scanning for non-occluding results is slower than using database indexes. Therefore, multiple queries are issued sequentially, each ruling out results that occlude those already chosen. This interdependency makes diversity sampling slower than the more usual uniform or stratified sampling.

The next section describes related work, followed by a description of our implementation, detailed comparison to alternative constant density algorithms, future work, and conclusions.

2. Related Work

The following systems address occlusion in visualizations, or else bear a strong surface similarity to the example 2D visualizations of images.

2.1 ThemeView

ThemeView lays out textual news stories in two dimensions according to similarity [Hetzler et al. 1998]. The axes do not encode meaningful variables. Clusters of similar articles are depicted as mountain peaks, and labeled with characteristic keywords. Thus it is showing a small sample of words in a visualization of a large document corpus. It would be interesting to use this layout technique for video, and label the peaks with images rather than text.

2.2 PhotoFinder

PhotoFinder manages personal photo collections [Kang and Shneiderman 2000]. It can show images in a grid, or in a scatterplot. In the scatterplot there is no attempt to reduce occlusion.

2.3 Dynamic Query (DQ)

The most common technique for reducing the number of [visible] objects in a visualization is Dynamic Query [Ahlberg et al. 1992]. A range slider is used to specify a range query on one variable. Only those records satisfying the range query are shown in the visualization. Much effort has been spent on efficient DQ algorithms, so that the visualization can be updated as the range slider is dragged, with no perceptible delay. To make updates fast, an interface object is created for every possible data point. At update time, only the objects' visibility must be changed. This is much faster than creating and destroying objects. However this algorithm breaks down in the case of video keyframe collages. There are 350,000 keyframes in the 2001 CNN news dataset, which is more than 9GB of .gif files. The computer will run out of virtual memory, and the user out of patience, before the slider can ever be manipulated. For the problem considered here, the number of objects on the screen at any time is small, so it is more efficient to create them as needed.

2.4 Visual Information Density Adjuster (VIDA)

DataSplash is a general purpose visualization system that supports semantic zooming [Woodruff et al. 1998]. When users zoom in to a visualization, the objects in the visualization may change form to show more detail. For instance a city may be shown as a dot from a high elevation, as a shaded area from a middle elevation, and a street map at a low elevation. In order to reduce clutter and wasted space, a DataSplash tool called VIDA automatically chooses an appropriate representation at each level. The main drawback is that a person must specify the hierarchy of alternate representations. "Selective omission" is mentioned as one type of representation change, but no algorithm is presented. Thus VIDA has concentrated on the harder problem of choosing among multiple representations, while largely ignoring the simpler problem considered in this paper, sampling. VIDA derives from cartography theory, where this is appropriate. Maps promise completeness: omitting San Diego because it occludes Los Angeles would be lying.

2.5 Astral Telescope Visualiser

Dix and Ellis [Dix and Ellis 2002] use an elevation metaphor similar to VIDA, but focus strictly on the problem of sampling. Sampling parameters are chosen based on local density measurements rather than object-based properties. Their main interest is stability; the system remembers which points have been chosen previously, so that they will be chosen again in similar circumstances. Our system could benefit from this idea. Their motivations for sampling are to speed up rendering and to avoid density saturation. They primarily consider scatterplots of points.

2.6 Video Manga and Other Storyboards

Much work has been done with storyboards, in which each keyframe from a video is laid out in time order on a grid [Zhang et al. 1995; Christel and Warmack 2001]. Since these algorithms display all the keyframes that they are given, they have not addressed the sampling problem.

Video Manga generalizes the grid layout to a comic-book style, and has considered interesting 2D layout issues. It ranks the shots in a video, and generates a visual summary showing the more important ones [Boreczky et al. 2000]. Among those chosen, the size of each image encodes three levels of importance (1x, 2x, and 3x). The summaries have no occlusion or whitespace. For better use of screen space, Video Manga folds the time axis into multiple rows. A single row is always 3x high, so a horizontal interval can show a 3x image, a 2x and a 1x image, or three 1x images. For each successive row in the 2D display, it assigns to that row the next unassigned keyframes in time order until the area of the preferred image sizes equals the row's area. Within a row, it considers all possible layouts of the keyframes, including reordering and resizing. Reordering and resizing each incur penalties, and the minimum-penalty layout is returned.

2.7 Informedia Digital Video Library (IDVL)

2.7.1 Collage Description

Figure 1 is a screenshot from one of the interface tools developed for the Informedia Digital Video Library. The *x*-axis shows the date each news story was broadcast. The *y*-axis shows the vector space similarity score between the transcript of the news story and the query. The scores are normalized so the best match gets 100 and the absolute minimum value is 0.

The IDVL interface is designed to minimize effects of latency between the client interface and a remote server. Therefore, it limits the results of any query to the 1000 most relevant, and it downloads them all at once. There is a delay of up to 30 seconds when a query is changed, but then changing visualizations and filtering do not require additional database accesses. (All times in the paper are for a 1.2GHz Pentium 3 laptop with 1GB of RAM.) For the 1000 results, Figure 1 is actually showing all of them either as a point or as an image. It is attempting to show both density information and selected keyframes in one visualization. There is much occlusion among images in November and December.

Determining whether to show a result as a point or an image is done based on the nominal variable `country` for maps, and based

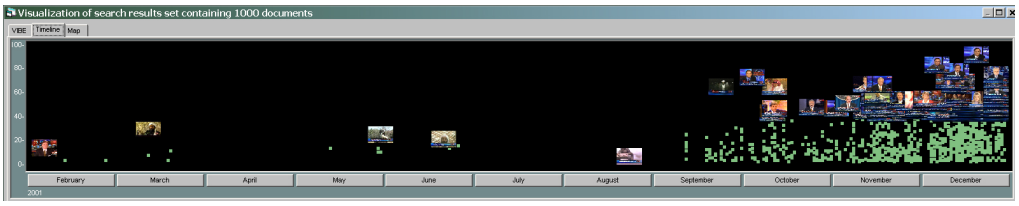

Figure 1 IDVL timeline collage for the query "osama bin laden al qaida" in a database of 2001 CNN news stories.

on the quantitative variable `time` for timelines. There is no general-purpose layout algorithm.

2.7.2 IDVL Collage Evaluation

We conducted an empirical study and found problems with the IDVL interface due to image occlusion, and that users did not take advantage of interactive controls for addressing the problem.

In more detail, collages of the format shown in Figure 1 were used with 20 users. Users were asked to describe visually and with text the newsworthiness of top people in the news from 2001. For this task, they were given collage interfaces that had image overlays with occlusion, and timeline scatterplots with no thumbnail images. There was a significant subjective preference for the image overlay interface. Efficiency and effectiveness metrics, however, showed no significant difference, perhaps in part because of usability issues uncovered during the experiment. Image occlusion and the difficulty of seeing the distribution through the overlaid imagery were two primary comments repeated by many users.

Specifically, four users commented that the thumbnails were too crowded or too overlapping, with another user commenting that "The timeline with dots was easiest to use as it clearly segmented the videos into different times and relevance, and allowed one to get a better idea of when most important events involving the characters took place." Users either had the image overlays locked on (one experimental treatment) or locked off (another treatment). The experimental results and user logs show the importance of allowing both the scatterplot view (for distribution) and the image view (for sampling).

With this same empirical study, users overwhelmingly did not resize the thumbnails to reduce occlusion, and did not zoom into a smaller time interval through dynamic query sliders in order to effectively spread the interface and again reduce occlusion. Users complained about the occlusion, but in the hour spent with the interface during the experiment they did not make use of the features that would have reduced occlusion, even though these features were described in an introductory on-line tutorial to open the experiment. Users overlooked these interface options in light of the other complexities presented with the collage interface. *Hence, for occlusion to be addressed, the system itself should have the means to better cover the space and remove overlapping, especially for novice users who will not know of other ways to address the problem.*

3. Implementation

3.1 Visage Platform

This paper describes interfaces built within the Visage general-purpose data exploration and visualization environment [Roth et al. 1997]. Visage supports visual queries with selects, projects, joins, and aggregation, as well as interactive design of visualizations incorporating multiple maps, charts, and networks; each using multiple sets of points, labels, lines, or images; each of which can encode data with size, shape, color, or position. As a Digital Library interface, Visage would be more appropriate for an expert user like an Intelligence Analyst, in comparison to the largely walk-up-and-use IDVL interface.

With a combinatorial number of possible designs, the occlusion management algorithm must be general purpose. This includes visualizations from any of the domains where Visage has been used – logistics, email, astronomy, and public health – not just video. We also assume a low-latency connection to the database, and rely on this to make multiple queries, each conditioned on the results of the previous ones. Since IDVL typically downloads many more images than will be displayed, these more precise queries use less total bandwidth than the Web-oriented design.

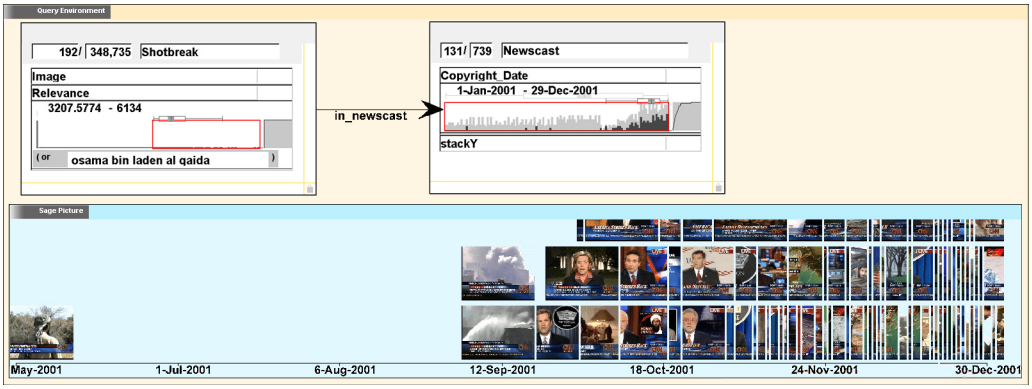

Figure 2 Traditional Visage timeline showing only the highest ranked results for the query "osama bin laden al qaida." There is no *y*-axis; the vertical dimension is used to stack multiple keyframes with similar dates. Note that the leftmost image is from May. If the timeline showed all of 2001 instead of the minimum interval including all the selected images, the occlusion and whitespace problems would be even worse. Background tints have been added in Photoshop to aid explanation here and below.

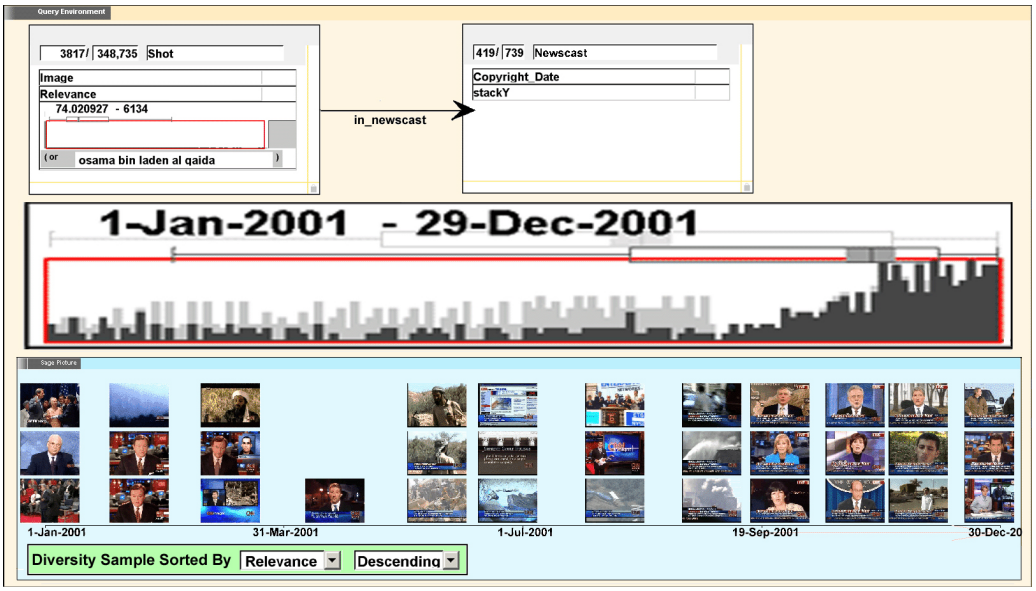

Figure 3 Visage timeline collage for the same query, where results are chosen based on both rank and visualization layout. The histogram and timeline *x*-coordinates were aligned manually.

Queries are represented with a boxes-and-arrows notation similar to that of Microsoft Access. Any query expressible in this visual language can form the basis for a collage. The two white boxes connected by an arrow at the top of Figure 2 and Figure 3 represent joins between a table of individual shots and the newscast of which they are a part. Dynamic Query sliders, such as those on relevance and copyright_date in the figures, allow rapid modification of the queries. In Figure 2, the slider (red outline) filters out all shots with relevance less than 3207 (in unnormalized vector space distance). The fraction at the top of the box shows that this leaves only 192 of the original 348,735 shots. The Copyright_Date histogram shows that the distribution for this query (black bars) is dominated by post-9/11 newscasts. The number of newscasts overall (gray bars) is higher after 9/11 as well.

3.2 Diversity Sampling Interface

There is a collage widget that can be dragged into any interactively designed chart or map. An example can be seen in Figure 3, in green at the lower left. When a visualization is inside the query tool, the presence of the widget causes the visualization to show a diversity sample of the query results rather than all results. There are two menus, which together specify the preference order in which query results are considered. The menu on the left specifies a data attribute to sort by, and the one on the right specifies whether to sort the values of the attribute in ascending or descending order. In the figure, results are sorted from most relevant to least relevant. These simple controls are quite general; one can define new data attributes using arithmetic formulas or attached procedures, and sort on the new attribute.

The Copyright_Date histogram has been dragged out of the Newscast query node and manually rescaled to align with the timeline collage. This makes it easier to correlate what is happening (from timeline) with how much is happening (from histogram). In the future we plan to automate the rescaling by supporting "docking" of visualizations with compatible axes.

3.3 Sampling Algorithm

1. **Compute Spatial Bounds** When the diversity sampling interface is first dragged into the query tool, the original query (which returns all 348,735 shots) is extracted, and used to generate a query for the minimum and maximum bounds of any quantitative variables that determine spatial location. In the example, copyright_date is such a variable. The other spatial variable, stackY, is not quantitative. The corresponding axes are then initialized to show this range. Otherwise, adding a new result could require an axis range to expand, which could invalidate previous non-occlusion determinations.
2. **Retrieve Some Results** Next the current query (which returns 3817 shots in the example) is extracted, and used to retrieve a small number of results, ordered as specified by the collage widget. **Exit if no results.** Otherwise, a grapheme representing each successive result is added to the visualization if it will not result in occlusion. The first result is guaranteed not to occlude results of previous queries, but multiple results of a single query may occlude one another.
3. **Constrain Query to Exclude Previous Results** The pixel bounds of each newly added grapheme are found and converted back to data value ranges. For quantitative variables, the query is modified to exclude results in any of these *x*-range × *y*-range regions. With nominal variables the modification is more complicated. For a visualization with a continuous x-axis, and where *y* is used only for stacking (as in the example), the maximum number of graphemes that will fit in a column, *column_height*, is computed. A result will occlude if there are already *column_height* graphemes with its x-value in the visualization OR if its x-value is not present and its x-range intersects that of an existing grapheme. Separate rules like this are required for each combination of the three axis types, quantitative, nominal, and stack. The algorithm is straightforward but not elegant.
   Return to Step 2.

On exit, it is guaranteed that no more results from the current query will fit in the visualization without occlusion. The algorithm is repeated if the original query is changed.

3.3.1 Dynamic Updates

Interactive visualization systems support dynamic operations like axis rescaling and Dynamic Query, which affect the amount of whitespace. An incremental algorithm is used to update the screen quickly as these parameters change.

1. Given the new layout, find a pair of graphemes with a high degree of occlusion and randomly delete one of them. Iterate until there is no more high degree occlusion. Using a narrower definition of occlusion in this step adds hysteresis and stability. Allowing graphemes to occlude up to 50% has worked well.
2. Go to step 3 of the sampling algorithm.

In our implementation, there is a noticeable delay before new images are added. For layouts similar to Figure 1, the delay ranges from a few hundred milliseconds to a few seconds. However we know of no other system that approaches this performance for arbitrary data. Video editing software uses a similar visualization of the linear sequence of frames, but it only works on one-dimensional layouts, and fast updating is based on an algorithmic correspondence between time and frame number.

3.4 Effectiveness

The greedy algorithm seemed to work well. Unfortunately it is difficult to be more specific. Comparing its effectiveness in minimizing whitespace to other algorithms is not straightforward. Each of the systems described under Related Work is solving a different task. Therefore they can only be compared with respect to a weighting function that trades off occlusion, displacement, resizing, whitespace, query relevance, effectiveness of each semantic zoom level, and whether the visual objects are points or have structure.

Only for comparing the greedy algorithm to an optimal one did a reasonably objective weighting suggest itself. Here we need only weight whitespace vs. query relevance. Since the images are all the same size, this is equivalent to balancing the number of images vs. their relevance. Given a truth set of answers and ranked query results, a common IR metric is "average precision." First, compute a new result ranking by throwing out the incorrect results. Then sum the quotients of the new and old rank for each correct result. Finally, divide by the number of correct answers. For example, if the results are A, B, C, but B is not correct, the average precision is (1/1 [for A] + 2/3 [for C]) / 2 = 5/6. The result is 1 if and only if all correct answers are ranked above all incorrect answers. Adapting this measure to evaluate diversity sampling, the results displayed play the part of the correct answers, and the ranking from the text query is evaluated with respect to that.

As a quick empirical test, we compared the greedy and optimal algorithms' performance on the 2001 newscasts, using the top 10 Google queries for 2002 as an unbiased test set. For each query, the top 2000 matching shots that mentioned geographical locations were laid out on a map. Considering such a large number of matches increases the chances of finding spatial outliers. Figure 4 shows the average precision for the two algorithms, averaged across these 10 queries (blue curves). Each query's average precision was normalized by dividing by the best average precision for that query, considering both algorithms and all result sizes. (This turned out to always be the greedy algorithm on 2000 results.) This prevents a few easy questions from dominating the

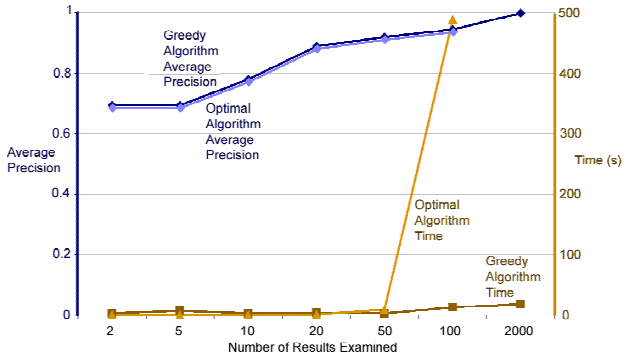

Figure 4 The average precision (left axis) and running time (right axis) of the two algorithms. The optimal algorithm is too slow to run on all 2000 query results. The time explosion is evident at 50 results, where it starts to diverge from the greedy time.

outcome. Unfortunately, the optimal algorithm is NP-hard (it can be reduced to the NP-complete 2D compaction problem [Tarjan 1983]). Therefore we could only evaluate it for the top 100 results, rather than all 2000 returned by the queries. Considering the top 2, 5,10, 20, 50, or 100 results, it *never* outperformed the greedy algorithm. Further, there is a noticeable improvement in average precision for 2000 results over 100 results. This reinforces our intuition that the greedy algorithm is a good choice. In the worst case, one image can occlude a maximum of 4 other images that don't occlude one another. Thus the greedy algorithm must be within a factor of four of optimal.

3.5 Efficiency

The number of results to retrieve in Step 2 can range from one to all. In the former case, many queries will be required, but each result set can be processed quickly. In the latter case, only one query will be required, but processing the result set will be slow. Figure 4 shows the average time required by the greedy algorithm to display the results of the 10 Google queries using an "ORDER BY" SQL query in step 2 (brown curves). The fastest time is when 20 results are processed at once. In the worst case, where each query returns only one non-occluding result, the slowdown is minimal. In the best case where all 20 results are non-occluding, it is faster by almost a factor of 20.

Another possible efficiency improvement is to simplify the constraints that filter out occluding answers. Using Vornoi Diagrams, it will sometimes be possible to combine multiple illegal regions into one larger one, or even to conclude that no more images can fit anywhere and avoid the query completely.

The algorithm scales sub-linearly to large datasets as long as the database is appropriately indexed. It scales linearly with the number of sample records, and it is reasonable to limit this to a few hundred because people are slow to sequentially scan unstructured data.

4. Comparison to Cartographic Generalization

VIDA [Woodruff et al. 1998], which is explicitly modeled on cartographic generalization, is the most closely related system to ours. This section explains in more detail why that approach tends to lead in a different direction from ours.

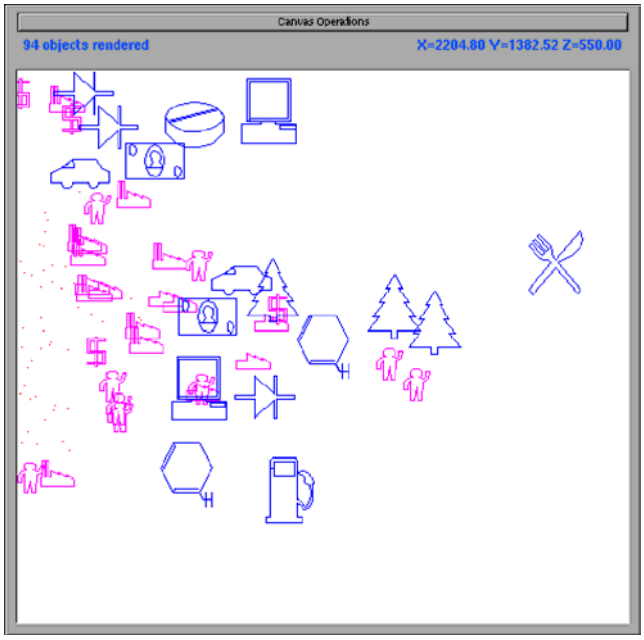

Figure 5 A VIDA visualization from [Woodruff et al. 1998] of the Fortune 500 companies, displaying an icon for the category of each company, laid out according to % profit growth (*x* axis) and income (*y*). There is much occlusion.

Cartographic generalization is the process of making a less detailed map from a more detailed one. For instance, area features can be replaced with point features, or less important features can be omitted, according to some *importance characteristic*. Although algorithmic techniques would be preferred, in practice a hierarchy of alternative representations is built by hand, from which features appropriate for a map of a desired scale can be selected automatically. The Principle of Constant Information Density (CID) [Timpf 1997] states that any sub-area of the map should contain about the same information density. Density depends on the number and complexity of graphemes. The focus on areas rather than objects is a key difference between our approach and cartographic ones.

Diversity sampling, as used here, is a much simpler problem than selecting from multiple representations in cartographic generalization. We make three simplifying assumptions about the data and task: 1) All images have the same complexity; 2) The goal density is the maximum achievable without occlusion; 3) The footprint of the visual object can be found incrementally. The first two seem reasonable, given that automatic algorithms cannot understand images well enough to estimate their semantic complexity. The third prevents the algorithm from laying out movable labels. The menus on the collage widget implement the cartographic importance characteristic.

Since CID is defined for areas, VIDA divides the display into a grid and makes one representation choice for each cell. This does not address non-uniformity within the cell, as seen in Figure 5. Companies are represented by points, small magenta icons, or large blue icons. Some of the small icons would be completely occluded if not for the fact that icon backgrounds are transparent.

VIDA relies on data fitting into memory in order to calculate cell density quickly for tens of thousands of records. Our system relies on efficient database indexing to build collages quickly even for disk-resident data.

5. Future Work

The layout algorithm described here addresses the occlusion problem found in the IDVL experiment, and it works around the fact that collages are poor at showing density by using multiple views. More experiments are needed to compare drill-down techniques, such as Dynamic Query and rubber-band selection. We need to understand what users expect in terms of stability. When do they expect images to reappear, and when do they expect new ones?

Our object-based approach can be extended to choosing among multiple representations as well. It would be applied to each representation in turn. Starting with the most compact, add data in priority order. Then scan the data again at the next more detailed representation, upgrading it as long as it doesn't cause occlusion.

6. Conclusion

6.1 Summary

From our experience with the TREC video retrieval competition, we knew that users can effectively scan large grids of images for those relevant to a query [Christel and Huang 2003]. In a collage, we relax the grid requirement in order to convey structured information, such as copyright_date, through layout. This way the spatial variation tells a story, either in time, space, % profit growth, or any other single or pair of variables. Minimizing empty space and occlusion implies a layout similar to that shown here.

6.2 Evaluation

Collages are meant to summarize large sets of unstructured data. While most attempts to visualize these datasets use aggregation and dimensionality reduction, our collages include a small diversity sample laid out with respect to meaningful axes. In all systems we know of that can show images in charts, data selection is performed before visualization layout. Therefore they cannot take advantage of a particular unused space or repair a particular occlusion.

Dynamic Query is based on attribute value ranges, not objects, so it can't select just one of a pair of objects whose structured attributes are identical (which will perfectly occlude one another). VIDA, Astral Telescope, and IDVL are based on regions, which are equivalent to attribute range values. PhotoFinder does not attempt adaptive density adjustment. Video Manga summaries use layout to encode time order, but no other variables. However the idea of minimizing reordering and resizing can be applied to the more general 2D case as well. If displacing an image by, say, 25% of its horizontal or vertical extent is allowed, then its footprint in the queries can be reduced to the area that it must cover no matter what displacement is used. The first query result will always satisfy the constraints after appropriate displacements.

In the figures, many of the images show news anchors and are therefore uninformative. Commercials and weather reports are also usually uninformative. Errors in speech or image recognition can result in irrelevant images. Diversity sampling worsens this effect by emphasizing outliers, which are often due to processing errors. In general, diversity sampling must be used with care on noisy data. If the noise is modeled explicitly, as it often is in data mining, the objects displayed can be restricted to those likely to be meaningful.

6.3 Generality

The iterative querying algorithm applies to any visualization where the position of a grapheme is not affected by the addition of more graphemes. That is why the maximum and minimum *x* and *y* values must be known ahead of time for quantitative variables. Movable labels, not to mention network visualizations, would require more complicated algorithms than we have considered. In networks, node location depends more on connectivity than on scalar attributes. Sampling is also problematic for networks because it affects global properties like average degree and connectedness.

Interactive visualization of large data sets has usually relied on downloading all the data once. Our algorithm uses a sequence of very specific database queries to interactively sample a large dataset. This approach may be useful in other situations where specialized kinds of sampling are appropriate. For instance, stratified sampling could reduce the variation between samples and make interactive visualizations more stable. For a hierarchical data model, multi-stage sampling and a hierarchical visualization might be appropriate.

For efficiency reasons, sampling is useful for giving quick feedback in visualizations of individual records from large datasets. In the authors' experience, it is reassuring to use collages as rapid feedback while exploring any large dataset, even if it doesn't involve unstructured data.

Apart from efficiency, Figure 2 and Figure 3 suggest that the technique simplifies any task where the relationship between one or two structured variables and a few unstructured variables must be understood. In these figures, the structured variable is copyright_date and the unstructured variable is image. In a map overview showing the effects of El Nino in different parts of the world, latitude and longitude are structured and image is unstructured. Captions could be added to the images, which would be a second unstructured variable. Three other uses are suggested below.

In studying the family tree of human languages, sets of equivalent words in different languages provide important evidence. Often nouns referring to everyday objects, such as "water" are similar across many languages, including long-dead ones. A visualization could show a hypothesized language tree, aligned with an *x*-axis showing time. Words from one of the word sets would appear on the appropriate branch and at the appropriate *x*-coordinate. For every language (and every dialect), there are many known occurrences of the word, and they will not all fit on the screen. Thus diversity sampling would be appropriate. Here time and language are structured, the latter hierarchically. Words are unstructured, at least if there is a sufficiently large number of variations.

Collages can even be used when all the data is unstructured. For instance, in ThemeView clusters of similar articles are depicted as mountain peaks, and labeled with characteristic keywords. Most of the space in the visualization is unlabeled lowlands. What if you are interested in a rarely covered topic that is somewhat related to a few of the peaks? It would seem helpful to label the lowlands to facilitate search for such topics. As long as peak labels are more salient, it seems that diversity sampling might be more effective than peak labeling alone.

By comparing genomes of different species, biologists can estimate probability distributions over possible evolutionary trees. Using tree-similarity metrics, the set of possible trees can be laid out in a plane. Then showing sample trees at various points can convey the layout of the space. This is just like ThemeView, but using trees as labels rather than words.

6.4 Significance

While applicable to trees, icons, and text, the most important application of these ideas is for images. PhotoFinder and numerous image- and video- search interfaces include lists, grids, or scatterplots of images. As datasets get larger, the emphasis will expand from retrieving individual images to exploring patterns. Collages are well suited to showing patterns of variation in image content as a function of the visualization's spatial variables.

The applications mentioned above can benefit immediately from this work. If the application already supports scatterplots, the interface may not have to change at all. More broadly, practitioners should carefully consider tasks that visualizations of visually complex information must support. It may often be effective to support the task considered here, exploring spatial patterns. Other tasks would be supported using multiple views.

As multimedia content becomes more widespread, and as metadata extraction becomes more accurate and links unstructured to structured data, visualizations that combine both will become more effective and more important. Much has been published already about visual layout of short text phrases meant to summarize text databases by topic, as in ThemeView. A number of papers also discuss layout of text and images for image and text retrieval, including organizing them by time (e.g. [Christel and Warmack 2001]). It seems certain that including more structured dimensions in these visualizations of unstructured data, and having a general principle for selection, will become increasingly important.

7. Acknowledgements

This material is based on work supported by the National Science Foundation (NSF) under Cooperative Agreement No. IRI-9817496. This work is also supported in part by the advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037. We thank Allison Woodruff for helpful discussions.

8. References

Ahlberg, C., Williamson, C. and Shneiderman, B. 1992. Dynamic Queries for Information Exploration: An Implementation and Evaluation. In *Human Factors in Computing Systems (CHI)*. Monterey, CA. ACM Press, 619-626.

Boreczky, J., Girgensohn, A., Golovchinsky, G. and Uchihashi, S. 2000. An Interactive Comic Book Presentation for Exploring Video. In *Proceedings of Human Factors in Computing Systems (CHI)*. The Hague, Netherlands. ACM Press, 185-192. http://www.fxpal.com/PapersAndAbstracts/papers/bor00.pdf

Christel, M. and Huang, C. 2003. Enhanced Access to Digital Video through Visually Rich Interfaces. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*. Baltimore, MD. IEEE Press, III.21 - III.24. http://www.informedia.cs.cmu.edu/documents/ChristelICME.pdf

Christel, M., Ng, D., Wactlar, H. and Hauptmann, A. 2002. Collages as Dynamic Summaries for News Video. In *Proceedings of ACM Multimedia*. Juan-les-Pins, France. ACM Press, 561-569. http://www.informedia.cs.cmu.edu/documents/ACMMM02_Collage.pdf

Christel, M. and Warmack, A. 2001. The Effect of Text in Storyboards for Video Navigation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Salt Lake City, UT. IEEE Press, 1409-1412. http://www.informedia.cs.cmu.edu/documents/ICASSP2001_Final.pdf

Dix, A. and Ellis, G. 2002. By chance - enhancing interaction with large data sets through statistical sampling. In *Proceedings of Advanced Visual Interfaces - AVI2002*. Trento, Italy. ACM Press, 167-176. http://www.comp.lancs.ac.uk/computing/users/dixa/papers/avi2002/dixellis-avi2002-v2.5.pdf

Hetzler, B., Whitney, P., Martucci, L. and Thomas, J. 1998. Multi-faceted Insight Through Interoperable Visual Information Analysis Paradigms. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis '98)*. Research Triangle Park, North Carolina, 137-144. http://www.pnl.gov/infoviz/insight.pdf

Kang, H. and Shneiderman, B. 2000. Visualization Methods for Personal Photo Collections Browsing and Searching in the PhotoFinder. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2000)*. New York. IEEE, 1539-1542. http://www.cs.umd.edu/hcil/photolib/paper/ICME2000-final.doc

NIST/SEMATECH. 2002. e-Handbook of Statistical Methods. . http://www.itl.nist.gov/div898/handbook/

Roth, S. F., Chuah, M. C., Kerpedjiev, S., Kolojejchick, J. A. and Lucas, P. 1997. Towards an Information Visualization Workspace: Combining Multiple Means of Expression. *Human-Computer Interaction Journal* 12, 1-2, 131-185. http://www.cs.cmu.edu/~sage/PDF/Towards.pdf

Tarjan, R. E. 1983. *Data structures and network algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Timpf, S. 1997. Cartographic objects in a multi-scale data structure. In *Geographic Information Research: Bridging the Atlantic*, M. Craglia and H. Couclelis, Editors. Taylor&Francis: London. 224-234.

Treisman, A. M. and Kanwisher, N. G. 1998. Perceiving visually presented objects: recognition, awareness, and modularity. *Current Opinion in Neurobiology* 8, , 218-226.

Wactlar, H. 2001. Multi-Document Summarization and Visualization in the Informedia Digital Video Library. In *New Information Technology*. Tsinghua University, Beijing, (Invited Speaker). http://www.informedia.cs.cmu.edu/documents/HDW_NIT2001_Paper.pdf

Woodruff, A., Landay, J. and Stonebraker, M. 1998. Constant Information Density Visualizations of Non-Uniform Distributions of Data. In *Proc. UIST '98*. San Francisco. ACM, 19-28. http://www2.parc.com/csl/members/woodruff/publications/1998-Woodruff-UIST98-Nonuniform.pdf

Xie, D., Singh, S. B., Fluder, E. M. and Schlick, T. 2003. Principal Component Analysis Combined with Truncated-Newton Minimization for Dimensionality Reduction of Chemical Databases. *Math. Program. Ser. B*, 95, , 161-185. http://monod.biomath.nyu.edu/index/papdir/fulllengths/pap_2_92.pdf

Zhang, H. J., Low, C. Y. and Smoliar, S. W. 1995. Video parsing and browsing using compressed data. *Multimedia Tools and Applications* 1, 1, 89-111.