ImageHive: Interactive Content-Aware Image Summarization

Li Tan, Yangqiu Song, and Shixia Liu ■ *Microsoft Research Asia*

Lexing Xie ■ *Australian National University*

ImageHive communicates information about an image collection by generating a summary image that preserves the relationships between images and avoids occluding their salient parts. It uses a constrained graph-layout algorithm first, to preserve image similarities and keep important parts visible, and then a constrained Voronoi tessellation algorithm to locally refine the layout and tile the image plane.

Finding interesting content in a large image collection can be difficult, whether it's photos of a friend's international trip or a presidential archive. One typical solution is to select a smaller set of representative images and compile them into a compact image summary for subsequent exploration, understanding, communication, and storytelling. Organizing images by content similarity can help users quickly locate information of interest.[1] However, users are distracted when the informative regions of images overlap, so summaries should avoid this. Furthermore, to deeply explore an image collection and exchange ideas, users must be able to interact with the summary in real time.

Existing image summarization algorithms take one of two general approaches (see the sidebar):

- Building exploratory image browsers that use layout to reflect the images' relationships, while allowing them to overlap arbitrarily.
- Building a compact, space-tiling image layout—a picture collage—while ensuring that the regions of interest in images don't overlap.

Little work has focused on combining these approaches to both maintain image relationships and avoid occlusion, let alone offering real-time user interaction to explore and understand the collection.

To address these problems, we developed ImageHive, a content-aware image summarization tool. ImageHive generates a compact Voronoi-like image layout that both preserves image relationships and avoids occluding salient image parts. In addition, it lets users interact with the image summary in real time to explore the collection and modify the summary to better communicate their ideas and stories. (An audiovisual presentation that augments this article's description of ImageHive is available at http://doi.ieeecomputersociety.org/10.1109/MCG.2011.89.)

ImageHive Design and Implementation

"Overview first, zoom and filter, then details on demand" is the mantra for seeking visual information. We followed this mantra, along with the observations reported by Kerry Rodden and her colleagues,[1] to establish three governing principles in the ImageHive design:

1. *The image summary should be compact and make the most salient regions in the collection visible.* Different parts of an image can invoke varying levels of visual attention, which can be encoded in image saliency maps and used to avoid occluding the most salient image regions and to preserve the maximum amount of information.
2. *The image layout should reflect pairwise content relationships.* A good summary communicates the correlations existing in the whole collection. Placing similar images adjacent to each other effectively minimizes the collective perceptual overhead.[1] It also facilitates further filtering and interaction with the collection.

 Published by the IEEE Computer Society

Related Work in Summarizing Image Collections

Researchers have proposed two distinct objectives for summarizing image collections.

Similarity-based summarization renders visualizations based on pairwise image content similarities. Its implementation has many variations, such as principal component analysis,[1] multidimensional scaling,[2] and manifold-based methods such as Isomap.[3] These approaches focus on the global image distribution and are optimized for placing the image centers without explicitly considering how to avoid overlapping and content occlusion.

Compactness-based summarization constructs a collage to avoid occluding salient image regions. AutoCollage uses constraint packing for image layout and graph cuts and Poisson blending for boundary matting.[4] Picture collage optimizes a Markov random field for salient-region placement,[5] and dynamic collage uses loopy belief propagation to update an image layout upon an incremental addition.[6]

In comparison, our approach relies on graph layout algorithms for global placement and Voronoi tessellation for local adjustment, both of which are very efficient. We ensure salient regions' visibility by incorporating them as constraints on both the global and local layouts, formulated with stress majorization and constrained Voronoi tessellation. Furthermore, our method can take into account any perceptual or semantic image similarity measure, such as object class, metadata, or content feature distances.

Some recent research addresses visual representation using Voronoi metaphor. *Voronoi treemaps* focus on two problems: the aspect ratio of rectangles and easily recognizing a graph's hierarchical structure.[7] The *Voronoi-diagram-buster algorithm* leverages the Voronoi diagram to evenly distribute a graph's nodes while maintaining the user's mental picture of the original graph drawing.[8]

Compared with these two methods, our method incorporates constraints into a Voronoi tessellation to avoid the occlusion of the salient image parts.

References

1. B. Moghaddam, Q. Tian, and T.S. Huang, "Spatial Visualization for Content-Based Image Retrieval," *Proc. 2001 IEEE Int'l Conf. Multimedia and Expo* (ICME 01), IEEE Press, 2001, pp. 22–25.
2. K. Rodden et al., "Does Organisation by Similarity Assist Image Browsing?" *Proc. 2001 SIGCHI Conf. Human Factors in Computing Systems* (SIGCHI 01), ACM Press, 2001, pp. 190–197.
3. G.P. Nguyen and M. Worring, "Interactive Access to Large Image Collections Using Similarity-Based Visualization," *J. Visual Languages and Computing*, vol. 19, no. 2, 2008, pp. 203–224.
4. C. Rother et al., "AutoCollage," *ACM Trans. Graphics*, vol. 25, no. 3, 2006, pp. 847–852.
5. J. Wang et al., "Picture Collage," *Proc. 2006 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 347–354.
6. Y. Yang et al., "An Improved Belief Propagation Method for Dynamic Collage," *The Visual Computer: Int'l J. Computer Graphics*, vol. 25, nos. 5–7, 2009, pp. 431–439.
7. M. Balzer and O. Deussen, "Voronoi Treemaps," *Proc. 2005 IEEE Symp. Information Visualization* (InfoVis 05), IEEE CS Press, 2005, pp. 49–56.
8. K.A. Lyons, "Cluster Busting in Anchored Graph Drawing," *Proc. 1992 Conf. Centre for Advanced Studies on Collaborative Research* (Cascon 92), vol. 1, IBM Press, 1992, pp. 7–17.

3. *The summarization tool should let users examine the image collection from multiple perspectives.* Users might have a specific goal or information need in mind that's not easily factored into a computation-similarity metric for principle 2. Interactive editing features help fulfill this need. Interactive exploration features such as zooming and filtering help users seeking further information.

Following these design principles, our image summarization pipeline first automatically clusters images and selects representative images for each cluster. Then we lay out the selected images in two interrelated steps, shown in Figure 1:

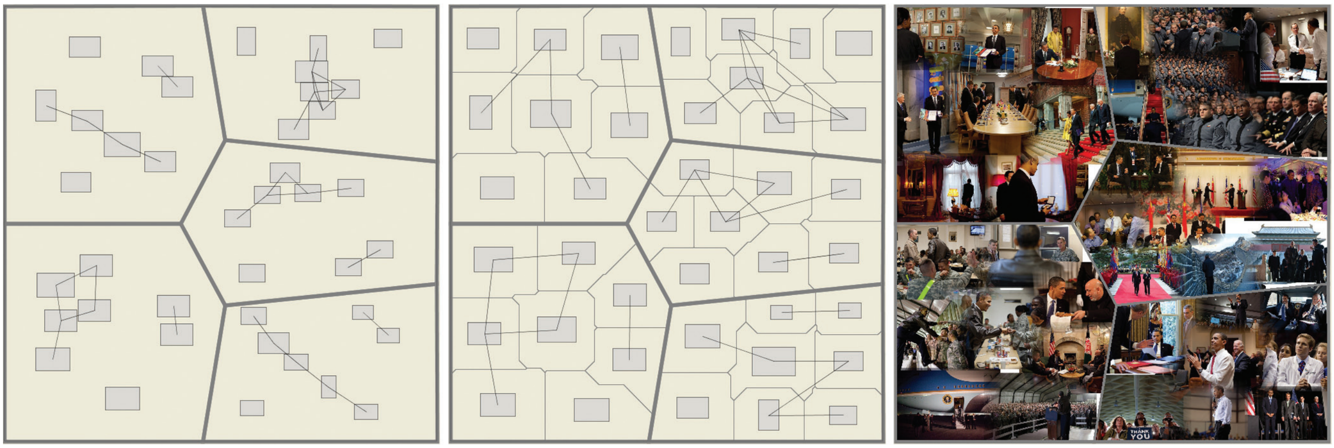
- *Graph based layout.* A similarity-preserving global-placement algorithm correlates the mutual distances between images and avoids occlusion between salient image regions. The choice of correlation metrics varies among collections and usages, ranging from the WordNet distance of image tags and descriptions to the vector distance between image features.
- *Online Voronoi tessellation.* A local adjustment to refine the layout uses a constrained Voronoi tessellation that minimizes image overlaps and maximizes the 2D layout area's coverage.

Finally, we provide real-time interaction to support editing and exploration.

In the whole process, we assume the images' aspect ratios and relative sizes are fixed, although users can resize the images in a preprocessing step that applies different scale factors. ImageHive's pipeline is flexible enough to support this extension.

Image Clustering and Representative Selection

A meaningful summary for a large image collection requires data transformation techniques to reduce the number of visualized images. In ImageHive, we first cluster images according to their correlation.



(a) (b) (c)

Figure 1. ImageHive's two-step layout. (a) Step 1 establishes a global placement. (b) Step 2 applies a local adjustment. (c) The resulting layout evenly distributes the images and maintains their relationships.

Taking the content-similarity relationship as an example, we obtain image correlations by calculating the similarity between various features, such as the scale-invariant feature transform, color histogram, and edge histogram. We leverage image category information or general clustering methods (such as *k*-means) to generate clusters.

We perform a Voronoi tessellation on these clusters, then correlate the images to build an image graph of each cluster. For example, an edge would connect two images with a similarity relationship, such as a shared tag. In each cluster, ImageHive selects several representative images—for example, the images nearest the calculated cluster center that also show some diversity. This lets users recursively navigate a large image collection.

Graph-Based Layout

To generate a global placement of images, we apply a constrained graph layout algorithm on the image graphs, which embeds them into a 2D plane (see Figure 1a), where the selected images' salient regions are visible.

The graph layout model. Consider a set of N images $\{\mathcal{I}_i\}_{i=1}^{N}$. To represent the relationships between images, we construct a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which each node corresponds to an image and the edge represents the similarity correlation between the images. Our method augments traditional graph layout methods by placing the image collection on a plane according to this representation.

Among various graph layout methods developed to transform a graph into a visual representation, one of the most popular is the one that Tomihisa Kamada and Satoru Kawai devised.[2] It aims to minimize the difference between the geometric distance in the layout and the ideal graph-theoretic distance. Users can also incorporate constraints into the original energy function for different visualization purposes.[3]

We can formulate the general constrained graph layout problem mathematically as

$$\min_{\mathbf{x}} \sum_{i<j} w_{ij} \left(\left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{L_2} - d_{ij} \right)^2, \tag{1}$$

subject to certain constraints being satisfied.

In Equation 1, d_{ij} is the graph-theoretic distance—the shortest path—between nodes v_i and v_j, $\mathbf{x}_i = \left(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\right)^T$ is the vector denoting the 2D coordinates of v_i, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ is an $N \times 2$ dimensional matrix composed of coordinates of all nodes, $\|\cdot\|_{L_2}$ is an L_2 norm of a vector, and w_{ij} is a normalization constant typically equal to d_{ij}^{-2}.

Constraint definition. We constrain the layout to satisfy the three design principles we described earlier. We encode principle 2 in the objective function of the graph layout model (Equation 1) and support principle 3 by user interactions, which we describe later. Here we mainly discuss the implementation of principle 1.

A natural way to model image saliency is to ensure that salient regions in adjacent images don't occlude each other. This presents two challenges in practice. First, preserving saliency while guaranteeing a compact layout is difficult. Second, constraining only the detected regions might not fully use the given 2D plane region.

So, to satisfy principle 1's two requirements, we derive two constraint types:

- *Hard constraints* guarantee that the given regions in adjacent images don't overlap and that salient regions display legibly.
- *Soft constraints* let regions have some overlap

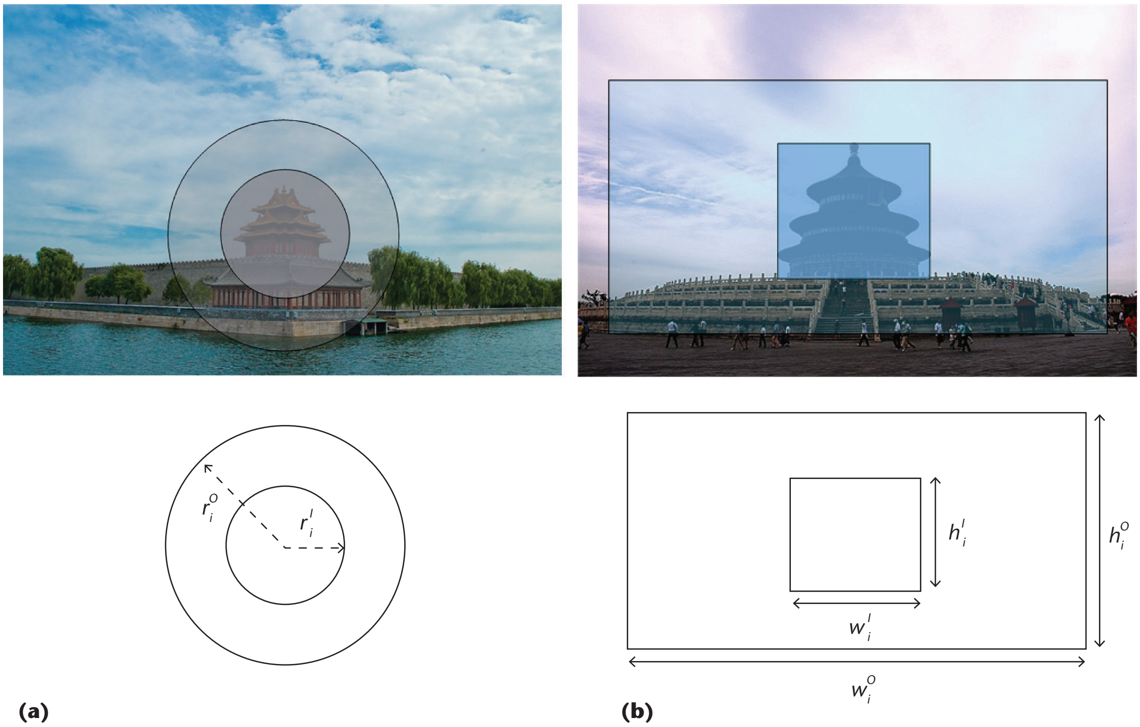


Figure 2. The two most common shapes to mark an image's salient regions: (a) circles and (b) rectangles. ImageHive formulates inner and outer bounding regions for each type.

and guarantee that marginally informative regions maximally use the unoccupied space.

Our system uses Laurent Itti and his colleagues' saliency-detection algorithm,[4] which usually produces satisfactory results. We also provide for user interaction to refine the results.

We then define two types of salient regions in each image, corresponding to the modeling of the two constraint types:

- *Inner salient regions* contain the most salient part, such as a scene's central object.
- *Outer salient regions* bound the marginally informative region around the central region.

Various methods exist for defining these two types of regions—for example, automatic saliency estimation, object detection, and user input. Typically, we define them as two concentric regions and incorporate them into the constrained layout model. The semantics of salient regions can address an entire object, a crop of an object, or different regions of the same image to reflect alternative photographic and perceptual interpretations.

Circles and rectangles are the two most common shapes to mark an image's salient regions (see Figure 2). We now illustrate how to formulate these two types of bounding regions for both hard and soft constraints.

For *circular inner regions*, we formulate the hard constraints as

$$\left(\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right)^2 + \left(\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right)^2 \geq \left(r_i^I + r_j^I\right)^2,$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the centers of two regions in two images, and r_i^I and r_j^I are the corresponding radii (see Figure 2a). This inequality indicates that for inner regions, the distance between two centers should be larger than the sum of two radii.

For *circular outer regions*, the soft constraints are

$$\left(\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right)^2 + \left(\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right)^2 + \xi_{ij} \geq \left(r_i^O + r_j^O\right)^2,$$

where $\xi_{ij} \geq 0$ is a relaxation parameter. This inequality indicates that for outer regions, by adding a small positive real value ξ_{ij}, the distance between two centers should be larger than the sum of two radii. We therefore expect ξ_{ij} to be as small as possible.

Because all the constraints are quadratic, we can easily formulate the optimization problem (Equation 1) as

$$\min_{\mathbf{X},\Xi} \sum_{i<j} \left(w_{ij} \left(\left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{L_2} - d_{ij} \right)^2 + C \cdot \xi_{ij} \right),$$

subject to (2)

$$\left(\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right)^2 + \left(\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right)^2 \geq \left(r_i^I + r_j^I\right)^2,$$

$$\left(\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right)^2 + \left(\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right)^2 + \xi_{ij} \geq \left(r_i^O + r_j^O\right)^2,$$

$$\xi_{ij} \geq 0,$$

where C is a trade-off parameter that controls the penalization of ξ_{ij}. A sequence of quadratic-programming procedures can solve this objective function.

For constraints based on rectangular regions, we further extend the regions to cases of different heights and widths. For example, Figure 2b represents the inner rectangle by its width and height $\left(w_i^I, h_i^I\right)$ and denotes the outer rectangle as $\left(w_i^O, h_i^O\right)$.

To ensure that *rectangular inner regions* don't overlap, we formulate the hard constraints as

$$\left|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right| \geq \frac{w_i^I + w_j^I}{2}$$

or

$$\left|\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right| \geq \frac{h_i^I + h_j^I}{2}.$$

Unlike the circular regions, the rectangular regions won't overlap if and only if one dimension satisfies the constraint.

For the *rectangular outer regions*, we formulate the soft constraints as an optimization problem:

$$\min_{\mathbf{X},\Xi} \sum_{i<j} \left(\xi_{ij}^{(1)} \cdot \xi_{ij}^{(2)}\right),$$

subject to

$$\left|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right| + \xi_{ij}^{(1)} \geq \frac{w_i^O + w_j^O}{2},$$

$$\left|\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right| + \xi_{ij}^{(2)} \geq \frac{h_i^O + h_j^O}{2},$$

$$\xi_{ij}^{(1)}, \xi_{ij}^{(2)} \geq 0.$$

where $\xi_{ij}^{(1)}$ and $\xi_{ij}^{(2)}$ are the relaxation parameters for different dimensions.

In this case, we expect the relaxation parameters for both dimensions to be minimized, so we can minimize the outer rectangles' overlap:

$$\min_{\mathbf{X},\Xi} \sum_{i<j} \left(w_{ij} \left(\left\| \mathbf{x}_i - \mathbf{x}_j \right\|_{L_2} - d_{ij} \right)^2 + C\left(\xi_{ij}^{(1)} \cdot \xi_{ij}^{(2)}\right) \right),$$

subject to

$$\left|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right| \geq \frac{w_i^I + w_j^I}{2} \text{ or } \left|\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right| \geq \frac{h_i^I + h_j^I}{2}, \quad (3)$$

$$\left|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}\right| + \xi_{ij}^{(1)} \geq \frac{w_i^O + w_j^O}{2},$$

$$\left|\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}\right| + \xi_{ij}^{(2)} \geq \frac{h_i^O + h_j^O}{2},$$

$$\xi_{ij}^{(1)}, \xi_{ij}^{(2)} \geq 0.$$

We could solve this problem by converting it to a sequential quadratic-programming problem with some relaxations, which could then be easily solved by standard quadratic-programming tools such as Mosek (www.mosek.com) and Cplex (http://www-01.ibm.com/software/integration/optimization/cplex-optimizer). However, to handle the constraints more efficiently, we solve the optimization problem through a heuristic gradient-projection method,[3] which has proven faster in practice.

Online Voronoi Tessellation

The global placement preserves the image similarity and the salient image regions' visibility, but it doesn't tessellate the layout area to maximize its usage. So, we refine this layout by introducing a constrained local adjustment algorithm that applies the online Voronoi tessellation (see Figure 1b). This generates a summary that maximizes the coverage of the 2D layout area, distributing the images evenly while preserving their relationships (see Figure 1c).

This fast tessellation method enables real-time user interaction with the summary image to reflect individual information needs.

Central-region-constrained Voronoi diagram. A Voronoi diagram is a decomposition of a metric space determined by the distances to a specified discrete set of objects in the space.[5] Typically, a Voronoi diagram's objects are a set of points. In our case, to make the salient regions visible, the objects can be a circle or a rectangle.

Two preliminary definitions are useful for the discussions here.

Definition 1. *Central-region-constrained Voronoi cell.* Given a set of separate central regions S, which are called Voronoi sites, the Voronoi cell associated to each region $S \in \mathcal{S}$ is an area $V(S) \subseteq \mathbb{R}^2$ such that each point $\mathbf{x} \in V(S)$ is closer to shape S than to any other one in $S \in \mathcal{S}$:

$$V(S) = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid d(\mathbf{x}, S) \leq d(\mathbf{x}, S'), \forall S' \in \mathcal{S} \setminus \{S\} \right\},$$

where $d(\mathbf{x}, \cdot)$ denotes the distance from $\mathbf{x}$ to the central region's boundary. The line segments forming the Voronoi cells' boundaries are called *Voronoi edges*.

Definition 2. *Central-region-constrained Voronoi diagram.* A Voronoi diagram is the union of Voronoi cells in an area on a 2D plane, and a regional constrained Voronoi diagram (RCVD) meets the additional objective

$$RCVD(\mathcal{S}) = \left\{ V(S_i) \middle| S_i \in \mathcal{S} \right\}.$$

Because we typically represent salient regions of images in circles or rectangles, we make the Voronoi cell's boundary piecewise linear by using a weighted Voronoi diagram (power diagram) for circles and a supremum metric (L_∞ norm) for rectangles.

Specifically, for circles (S is a circle C), we adopt the power metric

$$d(\mathbf{x}, C) = \|\mathbf{x} - \mathbf{c}\|_{L_2}^2 - r^2,$$

where r is the radius and $\mathbf{c}$ is the center of C. For rectangles (S is a rectangle R), we introduce the supremum metric

$$d(\mathbf{x}, R) = \min_{\mathbf{r} \subset R} \|\mathbf{x} - \mathbf{r}\|_{L_\infty},$$

where $\mathbf{r}$ is an arbitrary point belonging to R.

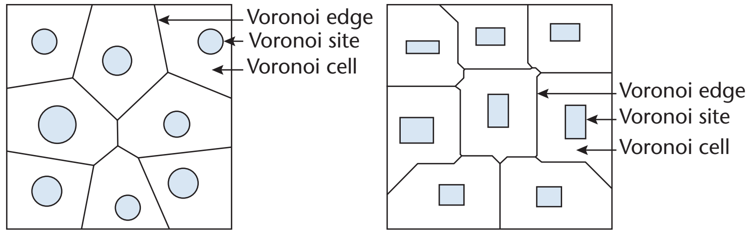
Figure 3 shows examples of circular and rectangular cases.

Central-region-constrained centroidal Voronoi tessellation. An intuitive way to generate a compact image summary that fully utilizes the layout space is to generate an evenly distributed image layout, in which the Voronoi cells' sizes are linearly proportional to the images' salient-region sizes. The ideal situation locates every Voronoi site in the Voronoi cell's centroid and is called a centroidal Voronoi tessellation (CVT).

One popular method for constructing a CVT is Lloyd's iteration procedure,[5] which performs two steps in each iteration:

1. Generate the Voronoi diagram on the basis of the Voronoi sites.
2. Recalculate the Voronoi cells' centroids, and move the Voronoi sites to the new centers.

Figure 4 shows an example of the iteration in the rectangle case. The circular case is similar.



(a) **(b)**

Figure 3. Central-region-constrained Voronoi diagrams: (a) a circular diagram constrained by a power metric and (b) a rectangular diagram constrained by a supremum metric. The constraints generate piecewise linear Voronoi cell boundaries.

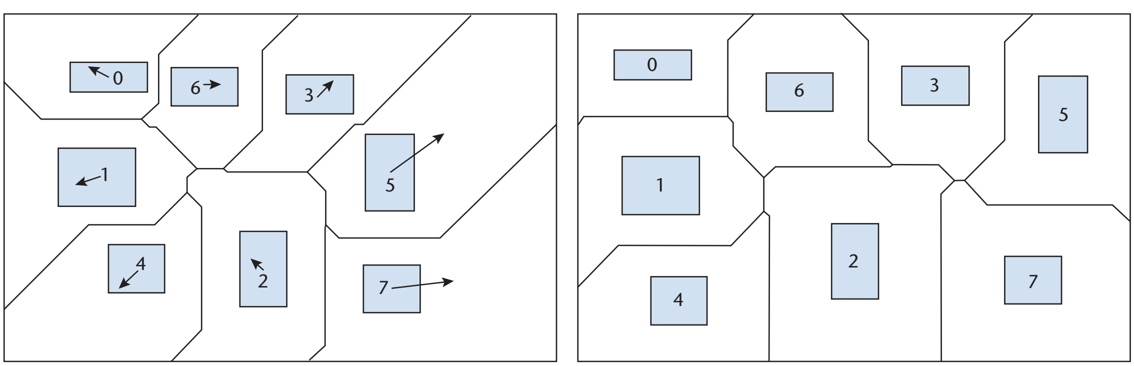
For the first step, we can construct the central-region-constrained Voronoi diagram by extending existing construction algorithms, such as the incremental-construction, divide-and-conquer, and sweeping-line algorithms. We use the incremental-construction algorithm for constrained Voronoi diagrams with circular sites and the sweeping-line algorithm for Voronoi diagrams with rectangular sites. For rectangular sites, prior sweeping-line algorithms deal only with point sites, weighted-point sites, and line sites. To overcome this limitation, we extended the sweeping-line algorithm for rectangular sites under the supremum metric.

In the second step, we develop a constrained optimization problem to move the sites to the new centers, under both the hard and soft constraints. Denoting the centroids by 2D vectors $\mathbf{C} = \{\mathbf{c}_i | 1 \le i \le N$, where N is the number of sites in $\mathcal{S}$, we formulate the problem as

$$\min_{\mathbf{X}} \sum_i (\mathbf{x}_i - \mathbf{c}_i)^T (\mathbf{x}_i - \mathbf{c}_i) + C \sum_{i<j} \left(\xi_{ij}^{(1)} \cdot \xi_{ij}^{(2)}\right),$$

subject to the hard and soft constraints.

The method to reformulate the constraints is equivalent to Equation 2 or Equation 3. So, we solve the problem by converting the equation to a sequential quadratic-programming problem such as



(a) **(b)**

Figure 4. Lloyd's iteration procedure. (a) Arrows indicate the direction for the Voronoi diagram's recalculated centroids. (b) The layout reflects the resulting centroidal Voronoi tessellation after the movement.

(a) (b)

Figure 5. Interactive editing with ImageHive: (a) before and (b) after adjustment. Users can refine the image summary results to highlight the healthcare reforms of Obama's presidency.

Equation 2 and 3, which we solve using a heuristic gradient-projection method.[3]

User Interaction

So that users can efficiently examine large-scale image collections, we implemented interactions for exploration at different levels of detail. The implementation rests on the image clusters' hierarchical structure. We gradually display the hierarchical structure between the adjacent levels by recursively partitioning the visual space as a treemap.[6] Because drilling down to see details is central to visual exploration, the constrained graph model incorporates layout techniques that keep the existing images stable during the exploration.

The editing function lets users further polish the generated image summary. Users can manually adjust the summary layout and add or delete images or clusters. Figure 5 shows the result of a user's interactive adjustment to focus on particular pictures. ImageHive updates the layout during these interactions by iteratively and incrementally recalculating the Voronoi cells and selectively moving the Voronoi sites to new centers. This gives the system a responsive feel (see the audiovisual Web Extra at http://doi.ieeecomputersociety.org/10.1109/MCG.2011.89).

Color blending is useful for eliminating joints between images and producing more appealing results, so we provide it as an additional postprocessing function.

Experimental Results

We quantitatively and qualitatively compared ImageHive with other semantic image summarization methods based on principal component analysis (PCA),[7] Isomap,[8] and picture collage.[9]

We measured the time the layout algorithms consumed, varying the image collection's size. All experiments executed on a PC with a 2.4-GHz Intel Pentium 4 processor. As Figure 6 shows, our method is slightly slower than the unconstrained methods, PCA and Isomap, because of the additional computations in our constraint optimization. For more than 400 images, ImageHive's speed is comparable to that of the Isomap-based method but is still slower than that of the PCA-based method. Our algorithm's complexity is $O(N^3)$ for the shortest-path computation. The quadratic programming's cost is comparable to Isomap's cost of computing geodesic distances and the cost of eigenvalue decomposition in both Isomap and PCA.

Nevertheless, our method is significantly faster than picture collage.[9] For a collection of 60 images, ImageHive took 0.14 sec., whereas picture collage took 140.82 sec. For a collection of 100 images, ImageHive took 0.39 sec., whereas picture collage took 715.44 sec. The estimation procedure for picture collage is the Metropolis-Hastings algorithm. Deciding on a good stopping criterion or diagnosing when the sampling has converged is difficult. The sampling used in Markov random-field inference, as in the case of picture collage, is empirically much slower than deterministic optimization methods such as the constrained graph layout.

To compare the visual summary results, we chose 50 images from the IAPR TC-12 image dataset (www.imageclef.org/photodata) with the labels including "sky" (blue and sunset), "cloud," "bicycle and highway," and "floor tennis court." We constructed the image graph from a similarity matrix computed over the concatenated color correlogram and color histogram features extracted by Imars (IBM Multimedia Analysis and Retrieval System; https://researcher.ibm.com/researcher/view_project.php?id=877).

Figure 7 shows the results. For the PCA and Isomap layouts (see Figures 7a and 7b), approximately one-half of the images are occluded, and much space is left unused. The collection's overall information is difficult to see and comprehend from the overviews. Picture collage (see Figure 7c), on the other hand, utilizes space efficiently but doesn't preserve the relationships, which makes gathering a collection-level summary difficult. ImageHive (see Figure 7d) laid out the similar images together without overlapping, making it easy to locate and compare similar images and different content clusters.

ImageHive Applications

From both our observations and informal user surveys, we found that ImageHive has many uses for communication and recollection. For example, a typical use is to share photos of scenes or events with others. This application is comparable to storyboarding, which shows plot scenarios.

Figure 8 shows the application of ImageHive to summarize views of some Chinese cities through famous landmark buildings and views that symbolize the location. The left cluster depicts famous structures in Beijing: Tiananmen Square, the Summer Palace, the Temple of Heaven, and the Great Wall. The top-right cluster shows Shanghai landscapes, and the bottom-right cluster represents the vibrancy of Hong Kong.

In this summary, we adopted rectangular constraints for their flexibility in shaping the salient regions of representative structures. We generated the correlations between images in each cluster by the similarity of color features to minimize color variation between adjacent images. To support different salient regions, we applied different weights to different pictures so that the most significant pictures are highlighted in larger Voronoi cells.

Another important use of images is to summarize photographed events. For example, Figure 9 shows events related to Barack Obama's US presidency. In this example, we used circle constraints to capture salient regions and the photo tag hierarchy to map the cluster structure. We generated the correlation between images by the cosine distance of histogram features. The results offer an anecdotal summary of Obama's early tenure.

Because ImageHive produces fast, compact layouts, it's suitable for deployment on small devices, including mobile phones and tablets. Figure 10 displays a single-cluster summary of animals photographed on a trip to Kenya and displayed on a mobile phone. We calculated the image correlations from their color histograms.

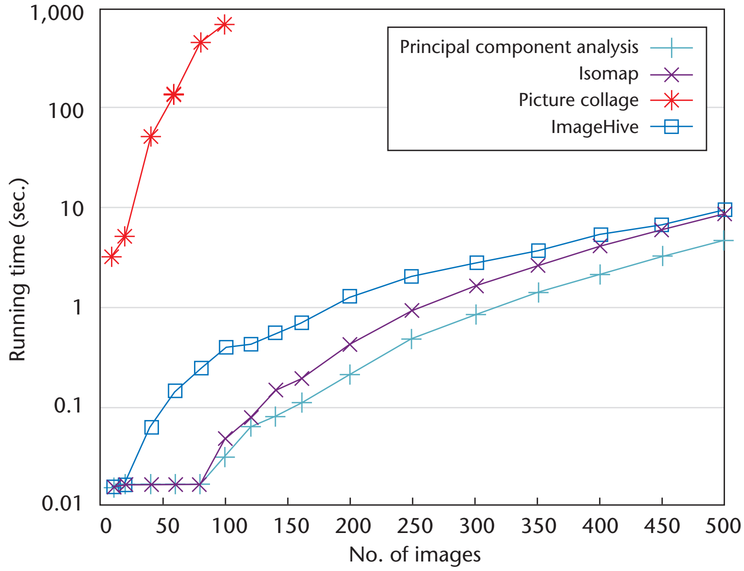


Figure 6. The running time of four image layout algorithms. In practice, our method is comparable to principal component analysis (PCA) and Isomap and outperforms picture collage.

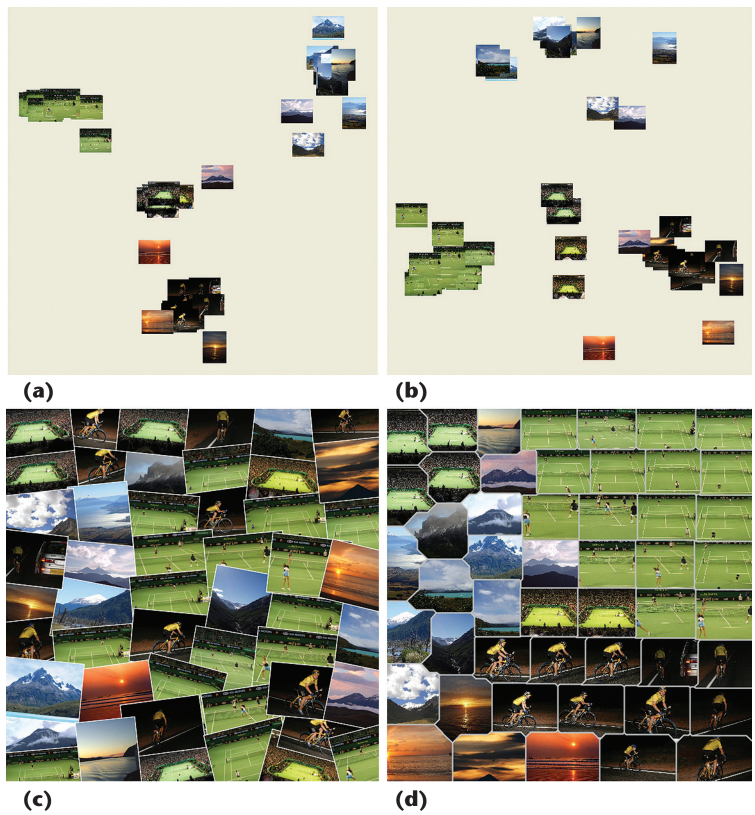


Figure 7. Visual summarization results for (a) PCA, (b) Isomap, (c) picture collage, and (d) ImageHive. Our method both preserves the relationship (compared to PCA and Isomap and utilizes space efficiently (compared to picture collage).

ImageHive creates an immediate impression of image collections that preserves facts, tags, and events embedded in metadata. At the same time, it's efficient enough to facilitate real-time user interactions and to be suitable for editing, filtering,

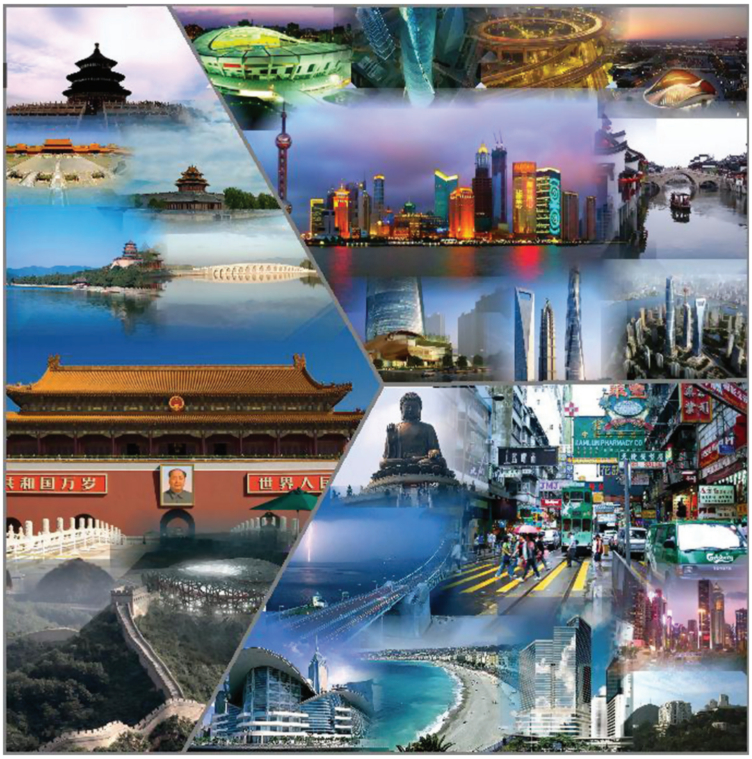


Figure 8. An image summary of Beijing (left), Shanghai (top right), and Hong Kong (bottom right). The image summarized these places with landmark buildings and views.

or interaction on mobile devices. Our future research includes tightly integrating image analysis and interactive visualization to support progressive analytic processes. We plan to implement this functionality as an underlying similarity metric or salient-region detection rule, on the basis of user feedback.

Acknowledgments

We thank Jingdong Wang for his kind help with setting up the experimental comparison with the picture collage method.[9]

References

1. K. Rodden et al., "Does Organisation by Similarity Assist Image Browsing?" *Proc. 2001 SIGCHI Conf. Human Factors in Computing Systems* (SIGCHI 01), ACM Press, 2001, pp. 190–197.
2. T. Kamada and S. Kawai, "An Algorithm for Drawing General Undirected Graphs," *Information Processing Letters*, vol. 31, no. 1, 1989, pp. 7–15.
3. T. Dwyer and K. Marriott, "Ipsep-Cola: An Incremental Procedure for Separation Constraint Layout of Graphs," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 5, 2006, pp. 821–828.
4. L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, 1998, pp. 1254–1259.
5. F. Aurenhammer et al., "Voronoi Diagrams," *Handbook of Computational Geometry*, Elsevier, 1996, pp. 201–290.
6. M. Balzer and O. Deussen, "Voronoi Treemaps," *Proc. 2005 IEEE Symp. Information Visualization* (InfoVis 05), IEEE CS Press, 2005, pp. 49–56.
7. B. Moghaddam, Q. Tian, and T.S. Huang, "Spatial Visualization for Content-Based Image Retrieval," *Proc. 2001 IEEE Int'l Conf. Multimedia and Expo* (ICME 01), IEEE Press, 2001, pp. 22–25.
8. G.P. Nguyen and M. Worring, "Interactive Access to Large Image Collections Using Similarity-Based

Figure 9. A summary of several events in Barack Obama's US presidency: a trip to Afghanistan (bottom left), winning the Nobel Prize (top left), working on healthcare reform legislation (bottom center), delivering a speech at West Point (top center), and traveling to Asia (right). ImageHive users can interactively generate a content-aware summary from image collections. (Photos publicly available at www.whitehouse.gov.)

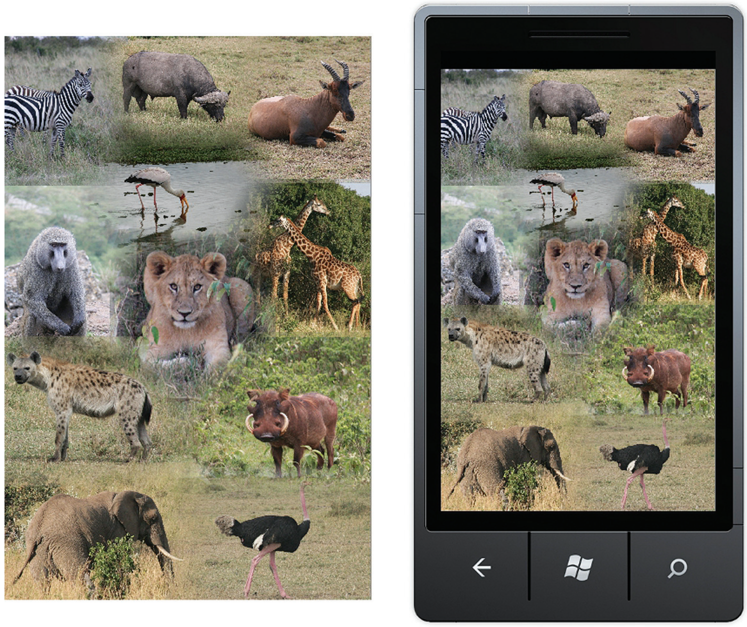
Visualization," *J. Visual Languages and Computing*, vol. 19, no. 2, 2008, pp. 203–224.
9. J. Wang et al., "Picture Collage," *Proc. 2006 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 347–354.

Li Tan *is an associate researcher at Microsoft Research Asia. His research interests include graph visualization and text visualization. Tan has an MS in computational mathematics from NanKai University. Contact him at lit@microsoft.com.*

Yangqiu Song *is an associate researcher at Microsoft Research Asia. His research interests include machine-learning algorithms and applications. Song has a PhD in control science and engineering from Tsinghua University. Contact him at yangs@microsoft.com.*

Shixia Liu *is a lead researcher at Microsoft Research Asia. Her research interests include visual text analytics and visual social network analysis. Liu has a PhD in CAD and computer graphics from Tsinghua University. Contact her at shliu@microsoft.com.*

Lexing Xie *is a lecturer in the Australian National University's Research School of Computer Science. Her research interests include multimedia mining and social-media analysis. Xie has a PhD in electrical engineering from Columbia University. Contact her at lexing.xie@anu.edu.au.*



(a) **(b)**

Figure 10. A summary of animals photographed on a Kenya trip. (a) The single cluster correlated images on the basis of their color histograms. (b) The image display on a smartphone. ImageHive generates compact layouts fast enough for handheld devices.