Discovering Favorite Views of Popular Places with Iconoid Shift

Tobias Weyand and Bastian Leibe
UMIC Research Centre
RWTH Aachen University, Germany
{weyand,leibe}@umic.rwth-aachen.de

Abstract

In this paper, we propose a novel algorithm for automatic landmark building discovery in large, unstructured image collections. In contrast to other approaches which aim at a hard clustering, we regard the task as a mode estimation problem. Our algorithm searches for local attractors in the image distribution that have a maximal mutual homography overlap with the images in their neighborhood. Those attractors correspond to central, iconic views of single objects or buildings, which we efficiently extract using a medoid shift search with a novel distance measure. We propose efficient algorithms for performing this search. Most importantly, our approach performs only an efficient local exploration of the matching graph that makes it applicable for large-scale analysis of photo collections. We show experimental results validating our approach on a dataset of 500k images of the inner city of Paris.

1. Introduction

Community photo collections have become a valuable source for large amounts of tourist photos, densely covering entire cities. In particular, they provide rich imagery of the world's landmark buildings, statues, monuments, and paintings. Our goal in this work is to automatically discover popular objects in such image collections and to find a representative and *iconic* view for each of them. Additionally, we aim at finding all images corresponding to those iconic views in order to structure the image data. Such a grouping enables many interesting applications, such as scene summarization [20], landmark recognition for image auto-annotation and visual search [11, 25, 8, 2], and 3D building reconstruction for use in virtual city models [12, 1, 10].

Several previous approaches have addressed this grouping task as a clustering problem [20, 16, 15, 12]. Similar to [8], we argue that a hard clustering is however the wrong task here, since there are many images that show multiple objects or buildings of interest together. In a hard clustering, those would be arbitrarily assigned to one of the clusters, when instead they should be assigned to both. In contrast, we propose to consider the task as a mode estimation problem. Instead of a hard partitioning into clusters, we aim at finding local attractors that summarize a number of images, but whose influence areas may overlap.

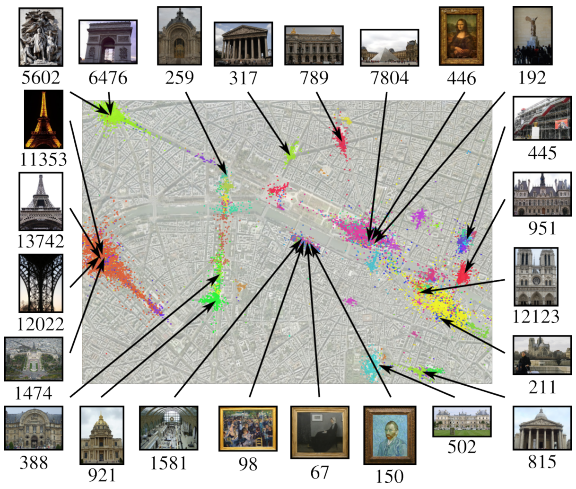


Figure 1: Map of iconic images (iconoids) and their cluster sizes discovered automatically by our algorithm in a set of 500k images from Paris. Our approach is designed for finding *central views* of a *single object or building* through efficient *local exploration*.

Many criteria have been proposed for structuring photo collections, including GPS tag proximity [8], global image similarity [12], the number of local feature matches [16, 15], or the number of shared 3D points of a reconstruction [20]. Our goal is to obtain a grouping that is defined on a building or object level and that can be computed efficiently by *local exploration*. That is, we want to only group images that show the same building, facade, or object, and we want to consider only images from the object's immediate vicinity in order to make the grouping decision.

An interesting question lies in the task definition itself: What properties should we aim for that make an image an iconic view? Intuitively, the iconic image should be similar to most other images in the cluster (the "likelihood" criterion from [23]), but we additionally impose the constraint that it should show the most central viewpoint on the object from those sampled by the human photographers. Our goal is therefore to find photos that have maximal mutual overlap with other images (*i.e.*, photos showing a similar view should have low distance, while panning and zooming

should be penalized). Therefore, distances based only on feature-based similarity are unsuitable, since they have no geometric interpretation. We instead propose a novel geometric distance measure, the *homography overlap distance*, and a corresponding mode search algorithm, called *Iconoid Shift*, derived from medoid shift [19, 22]. Seeking to maximize the mutual overlap with a large number of images in the neighborhood, this mode search procedure shifts the kernel window to images which show the depicted structure from the most central (and therefore iconic) viewpoint among their neighboring images, called *iconoids*, as visualized in Fig. 1. Additionally, Iconoid Shift produces a soft clustering of the image set by associating each image with all iconoids showing the same scene. We present an efficient linear-space algorithm to perform the medoid-shift search on large image databases. In addition, we show how the approach can be combined with Geometric min-Hash [6] to efficiently select promising seed points for exploration.

Our novel view of the problem has several important advantages: (1) It draws direct parallels to the well-understood problems of density estimation and mode search. This makes it possible to draw upon the proven theoretical properties of mode search algorithms and their intuitive parameters. In particular, we show that a hierarchical organization of iconic views can be obtained by varying the bandwidth of the kernel. (2) The iconoid kernel has a direct visual interpretation and delivers good empirical results. We experimentally demonstrate that both the selected iconic images and their supporting image groups make intuitive sense. In addition, we show that our procedure results in more centered iconic views than criteria based on the valence of nodes in the matching graph [15, 8]. (3) Our approach performs a local exploration of the matching graph. That is, we do not need to precompute the full pairwise matching graph (in contrast to [15, 1]), but our approach can be applied to many seed images in parallel. This makes the approach attractive for large-scale analysis of photo collections.

Related Work. A number of approaches have already been proposed that try to find iconic summary images for object categories [3], general visual concepts [17], or buildings [12, 20, 15, 4, 16]. This problem is closely related to the one of finding canonical views of 3D objects (*e.g.* [9]). In our approach, we follow the strategy of [20, 8] in relying on a population of photographers to provide a distribution over camera viewpoints and in searching for iconic images as peaks of the distribution. Our approach however differs from [20, 8] in several respects. We do not define image similarity over the number of shared 2D or 3D points, but over the size of the 2D inlier region of a homography relating the images. This allows our approach to be specific to individual buildings, but also to exploit transitivity to images of the same building for which there are no direct feature correspondences. In addition, our algorithm is optimized for efficient local exploration of large image collections, while [20, 15] work on precomputed connected components of the matching graph. Finally, [20] performs a greedy optimization to obtain a hard clustering, while our approach obtains stable results with a mode search.

The techniques we use for landmark building discovery also set us apart from other approaches with similar goals. [15, 8] apply spectral clustering on connected components of the matching graph, which is computationally expensive [24]. [16, 11] use agglomerative clustering on images in the same geospatial grid cell instead, which results in hard clusters, but does not find representative images. [5, 4] apply min-Hash together with query expansion in order to find clusters of partially overlapping images. However, the extracted clusters are not restricted to individual objects or buildings, but may extend to entire connected components of the matching graph [24].

2. Iconoid Shift

To lay out the foundations of the Iconoid Shift algorithm, we first give a brief review of visual word based image retrieval and medoid shift. We then define the homography overlap distance which enables applying mode search to find popular views in image collections and propose an efficient propagation scheme for the quick computation of pairwise distances in an image graph. Finally, we introduce the Iconoid Shift algorithm itself.

Problem Definition. We aim to find a subset of images that each have locally maximal mutual overlap with their neighboring images (the *iconoids*). Additionally, we are interested in each iconoid's corona of supporting images, which we call *support set*, that have non-zero overlap with it.

Visual Word based Image Retrieval. Our method uses the vector space model for image retrieval [21, 14]. Images are represented as bags of vector-quantized SIFT [13] features using a visual vocabulary size of 1M. Retrieval is performed using an inverted file based voting scheme and results are ranked by the cosine distance of their $tf * idf$ vectors to the query. The top-k matches are verified by fitting a homography using SCRAMSAC [18]. A match is accepted if it has more than 15 inliers.

Medoid Shift Review. Medoid shift [19] is an iterative mode search algorithm based on the classical mean shift [7]. Mean shift finds modes in a data set $\{\mathbf{x}_i\}$ by searching for local maxima in the kernel density

$$f(\mathbf{x}) = c \sum_i \Phi(\mathrm{d}(\mathbf{x}, \mathbf{x}_i)). \quad (1)$$

Here, Φ is a kernel, d a distance function, and c a normalization constant, such that Φ integrates to 1. Mode search is performed efficiently by iteratively shifting the kernel center $\mathbf{y}_k$ in the direction of the gradient:

$$\mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \left\{ \sum_i \mathrm{d}(\mathbf{y}, \mathbf{x}_i) \varphi(\mathrm{d}(\mathbf{y}_k, \mathbf{x}_i)) \right\}, \quad (2)$$

where $\varphi(\mathbf{x})$ is a kernel such that $\varphi(\mathbf{x}) = -\Phi'(\mathbf{x})$, *i.e.* Φ is the *shadow* of φ [7]. This minimization is iterated until $\mathbf{y}_k = \mathbf{y}_{k+1}$. The initial point $\mathbf{y}_0$ is then associated with the mode $\mathbf{y}_k$. Clustering is performed by applying this procedure once to each point in a dataset.

In medoid shift, the only formal difference is that the kernel center must always lie on a data point [19]:

$$\mathbf{y}_{k+1} = \arg\min_{\mathbf{y} \in \{\mathbf{x}_i\}} \left\{ \sum_i \mathrm{d}(\mathbf{y}, \mathbf{x}_i) \varphi(\mathrm{d}(\mathbf{y}_k, \mathbf{x}_i)) \right\}. \quad (3)$$

The advantage of this small modification is the generalization from Euclidean spaces to general metric spaces (*e.g.*, fully connected graphs). This means the algorithm only requires a distance measure to be defined between each pair of data points. In the following, we introduce an overlap-based distance measure for images.

Homography Overlap Distance. Since the modes we are searching for are images having maximal overlap with their neighborhood, we need a distance measure that rewards similar views while penalizing view changes like panning and zooming. To determine the *overlap region* between two images i and j, we estimate a homography H_{ji} that maps from image i to j. We now define the overlap regions x_{ij} and x_{ji} as the axis-aligned bounding boxes around the inlier features of the homography. Here, x_{ij} is the bounding box around the inlier features in image j and x_{ji} is the bounding box around the inlier features in image i.

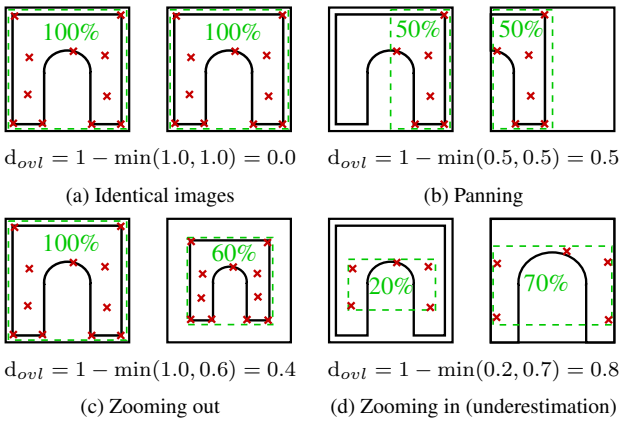
We then compute the relative size of the overlap regions in both images and define the *homography overlap distance* as one minus the minimum of these relative sizes:

$$\mathrm{d}_{ovl}(i,j) = 1 - \min \left\{ \frac{||x_{ji}||}{||R_i||}, \frac{||x_{ij}||}{||R_j||} \right\}. \quad (4)$$

Here, $||R_i||$ and $||R_j||$ denote the area of image i and j, respectively.

Properties. The effect of this definition is visualized in Fig. 2. If the images are identical (Fig. 2a) $\mathrm{d}_{ovl} = 0$, since both inlier bounding boxes fill the whole images. Now, if we pan the view (Fig. 2b), the size of the overlap region decreases equally in both images and d_{ovl} increases. In the case of zooming out (Fig. 2c) or in (Fig. 2d), the relative size of the smaller overlap region determines the value of d_{ovl}. As illustrated in Fig. 2d, this method sometimes underestimates the overlap due to homogeneous image regions where no interest points are present.

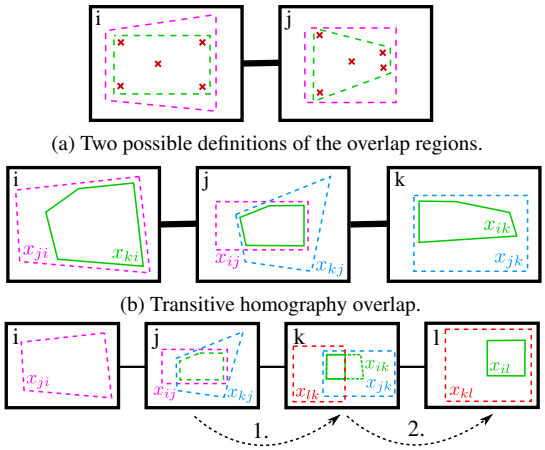
It is easy to verify that d_{ovl} is positive and symmetric. However, due to the limited repeatability of feature detectors and descriptors, the triangle inequality does not hold in general. In the following, we propose a transitive extension of d_{ovl} that fulfills the triangle inequality by construction.



(a) Identical images (b) Panning

(c) Zooming out (d) Zooming in (underestimation)

Figure 2: Illustration of the homography overlap distance.



(a) Two possible definitions of the overlap regions.

(b) Transitive homography overlap.

(c) Homography overlap propagation along a path.

Figure 3: (a) Direct, (b,c) transitive homography overlap.

Transitive Homography Overlap Distance. To determine local modes, medoid shift requires the pairwise distances of all images within the kernel radius. However, computing these pairwise distances by direct feature matching (as done, *e.g.*, in [16]) is very costly. Instead, we represent the local neighborhood by a tree and at first only compute distances along edges. Then, we infer the distances of all other pairs using their connecting path in the tree.

A simple approach for this would be to multiply all homographies along the path and to use the inliers of this homography to determine the overlap region. However, our experiments (Sec. 4) show that this method is not robust due to the limited invariance of the interest point detectors and descriptors and thus often underestimates the overlap.

We therefore propose a propagation scheme that is independent of feature matches and avoids the costly step of determining the homography inliers. In the following, assume there exists a direct correspondence between the image pairs (i,j) and (j,k) and our goal is to infer the homography overlap distance of (i,k). As a simple example, consider Figure 3b. We estimate the overlap regions x_{ki}

and x_{ik} (green regions in images i and k) by intersecting x_{ij} with x_{kj} (the magenta and blue regions in image j), and projecting the intersected region into images i and k using the known homographies. The homography overlap distance can then be computed as in Eq. (4) without explicitly matching i and k. Likewise, we can easily compute the homography from i to k as

$$H_{ki} = H_{kj} H_{ji}. \tag{5}$$

In order for this scheme to work, we need to be able to transform overlap regions between images. This is not possible when defining both overlap regions as axis-aligned bounding boxes, as we did above. We therefore define only one overlap region as an axis-aligned bounding box and the other as its image w.r.t. the homography. As shown in Figure 3a, we have two choices for this. We choose the pair of boxes that encloses the set of inliers better, *i.e.* the pair whose sum of areas is smaller (the green boxes in Fig. 3a). This makes the overlap regions consistent with the homography, *i.e.*

$$x_{ji} = H_{ij} x_{ij} \text{ and } x_{ij} = H_{ji} x_{ji}. \tag{6}$$

To define this scheme formally, let $x \cap y$ be the intersection between two regions x and y. We then define the overlap region of image i in image k as

$$x_{ik} := H_{kj}(x_{ij} \cap x_{kj}). \tag{7}$$

Now, x_{ki} can either be computed analogously, or by back-projecting x_{ik}:

$$x_{ki} = H_{ik} x_{ik} \tag{8}$$

By applying this scheme recursively, we can propagate overlap regions along paths. As an example, consider a path of four images (i, j, k, l) with correspondences only between adjacent images (Fig. 3c). We can compute x_{il} using Eq. (7) twice:

$$x_{il} = H_{lk}(H_{kj}(x_{ij} \cap x_{kj}) \cap x_{lk}) \tag{9}$$

Here, we intersect the magenta and blue regions in image j, project the intersection to image k, yielding the green region, which we intersect with the red region. Finally, we project the resulting solid green region to image l to get x_{il}. By alternating intersection and projection in this way, we can determine the transitive homography overlap distance between any pair of images that are connected through a path of pairwise correspondences.

Using this procedure, the triangle inequality is fulfilled, because the minimum spanning tree fulfills it by definition and all remaining distances are inferred using this propagation scheme which fulfills it by construction. In practice, violations only occur in cases where the polygon intersection algorithm fails due to degenerated polygons. Those cases are so rare that they do not have any effect on the algorithm's convergence and can easily be filtered out.

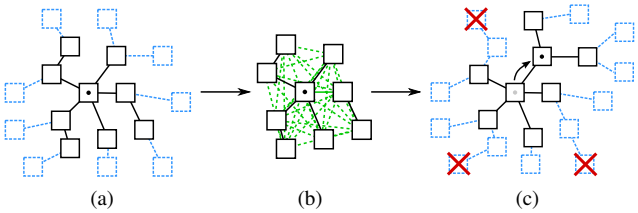


Figure 4: Steps of the Iconoid Shift procedure. (a) Exploration of the minimum spanning tree. Dashed boxes denote images outside the current kernel window, (b) Computation of pairwise distances, (c) Shifting the medoid.

Algorithm 1 Iconoid Shift.

```
// Given a collection C of tourist photos
// Compute the set of O iconoids and their support sets
S ← draw a set of seed images from C
for s ∈ S do
    y_0 ← ∅, y_1 ← s, k ← 1
    while y_{k-1} ≠ y_k do
        Build minimum spanning tree T starting from y_k
        Complete missing edges in T by propagating overlaps
        y_{k+1} ← the image in T minimizing Eq. (3)
        k ← k + 1
    Add (y_k, T) to O
```

Hinge Kernel. Having introduced our distance function on images, we define the shadow kernel Φ as a hinge function that is 0 for all distances above a threshold β. The kernel $\varphi(d) = -\Phi'(d)$ then becomes a piecewise constant function that cuts off all distances greater than β:

$$\Phi(d) = (1 - \frac{d}{\beta}) \text{ if } d < \beta \text{ , 0 otherwise} \tag{10}$$

$$\varphi(d) = \frac{1}{\beta} \text{ if } d < \beta \text{ , 0 otherwise} \tag{11}$$

Iconoid Shift. Now, we have all the components that are needed to define the Iconoid Shift procedure. The algorithm follows the principle of medoid shift, but incorporates some modifications to make it applicable to our problem.

Starting from an initial center image, we construct a minimum spanning tree of images overlapping with it by locally exploring the neighborhood of the central image using recursive image retrieval (Fig. 4a). Matches are verified by computing their homography overlap distance (Eq. (4)) with the root node. For the children of the root, this is done directly (Fig. 3a). For nodes further away, the overlap is propagated transitively (Fig. 3c). In the second step, we compute the pairwise distances between all images in the graph (Fig. 4b). This can be performed very efficiently (*cf.* Sec. 3) by again exploiting the transitive definition of the homography overlap distance (Fig. 3c). Finally, we compute the next medoid (Fig. 4c) using the standard medoid

shift minimization (Eq. (3)) and iterate this procedure until a convergence point (the *iconoid*) is reached. This mode search is performed for a previously selected group of *seed* images and the set of resulting iconoids and their minimum spanning trees is returned. The overlapping clustering is then given by the images contained in the minimum spanning trees. The full algorithm is shown in Alg. 1. Note that it can easily be parallelized by distributing the mode search for different seeds to multiple threads or compute nodes.

The algorithm's steps have an intuitive interpretation: Starting with a seed image, we explore the set of images overlapping with it and compute their pairwise distances. In the medoid shift step, we compute weighted sums of overlap distances (Eq. (3)) which rate how well each image represents all other images in the neighborhood. The best representative is then chosen as the new iconoid, which tends to be the most central view on an object (see Fig. 6).

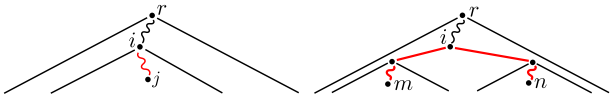
In contrast to medoid shift, our approach produces an overlapping clustering, since the clusters are the iconoid influence areas. Also note that unlike, *e.g.*, [15, 1] we do not compute the full matching graph but only the local neighborhoods of the points along the convergence paths.

3. Efficient Implementation

We now introduce efficient algorithms for both the exploration and the distance computation steps of Iconoid Shift. The proposed exploration procedure builds a minimum spanning tree of images overlapping with a central image, which enables an efficient implementation of the pairwise distance computation by homography overlap propagation (HOP). In particular, by interleaving the distance computation and medoid shift minimization steps, the memory requirements of our algorithm are linear in the number of images within the kernel window.

Local Exploration and Minimum Spanning Tree Construction. The exploration procedure works by querying an image retrieval system (Sec. 2) with the root node r to obtain potentially matching images. Each match i is verified by computing the homography overlap distance with the root node $\mathrm{d}_{ovl}(i, r)$. If this distance is within the kernel radius, *i.e.* $\varphi(\mathrm{d}_{ovl}(i, r)) > 0$, the match is accepted and added to the graph. This procedure is executed recursively, building up a minimum spanning tree in a breadth-first manner (Fig. 4a). In order to efficiently compute the homography overlap distances with the root node, each node i stores its overlap region with the root x_{ri}. After a set of potential matches has been retrieved, their homography overlap distances to the root can then efficiently be computed by propagating the overlap region of the query node (Eq. (7)). This way, only $O(N)$ propagation steps have to be performed, where N is the number of images within the kernel radius.

Homography Overlap Propagation (HOP). Having constructed the minimum spanning tree, we now compute the distances between all pairs of nodes (Fig. 4b) and determine the medoid using Eq. (3). A naive implementation of this step would require runtime and storage cost in $O(N^2)$ which quickly becomes infeasible for large image sets. Instead, we propose an efficient divide-and-conquer algorithm that requires only linear space.



(a) (b)

Figure 5: Homography overlap propagation. (a) The lowest common parent i propagates its overlap region to its subtree. (b) Distances between nodes in different subtrees are propagated via their lowest common parent i.

Algorithm 2 Homography Overlap Propagation (HOP).

// Given a minimum spanning tree T with root r
// Compute the medoid m
var $\{D_i\}$ // Sums of weighted distances for all images i
$D_i \leftarrow 0$ for all images i in T
for all images i in T pre-compute $\varphi(\mathrm{d}_{ovl}(r, i))$ (Eq. (11))
for all images i in T traversed breadth-first starting at r **do**
  // **Step 1**: Propagate root overlap (Fig. 5a)
  for all images j under i traversed breadth-first **do**
    Compute H_{ij}, x_{ij}, x_{ji} by recursive propagation (Eq. (7))
    $D_j \leftarrow D_j + \mathrm{d}_{ovl}(i, j)\varphi(\mathrm{d}_{ovl}(r, i))$ (Eq. (3))
    $D_i \leftarrow D_i + \mathrm{d}_{ovl}(i, j)\varphi(\mathrm{d}_{ovl}(r, j))$ (Eq. (3))
  // **Step 2**: Compute pairwise distances (Fig. 5b)
  for all image pairs (m, n) in different subtrees of i **do**
    $D_m \leftarrow D_m + \mathrm{d}_{ovl}(m, n)\varphi(\mathrm{d}_{ovl}(r, n))$ (Eq. (3))
    $D_n \leftarrow D_n + \mathrm{d}_{ovl}(m, n)\varphi(\mathrm{d}_{ovl}(r, m))$ (Eq. (3))
$m \leftarrow \arg\min_m \{D_m\}$

Our algorithm exploits the transitive homography overlap distance to propagate overlaps in the minimum spanning tree. The central idea is that for two images in different subtrees, propagation always goes through the lowest common parent. The overlap with this parent can be pre-computed and re-used for all pairs of images below it.

For each lowest common parent i, we proceed in two steps: First, we propagate the homography overlap of i to all nodes j in its subtree (Fig. 5a). Then, we use the transitive propagation scheme (Eq. (7)) to compute the distances between all nodes n and m that have the lowest common parent i. The full algorithm is given in Alg. 2.

This algorithm has $O(N)$ memory complexity in the size of the tree, because we directly accumulate the kernel-weighted sums of distances (Eq. (2)) instead of storing all $N^2/2$ pairwise distances and computing the weighted sums in a separate step. The time complexity of this algorithm is $O(N^2)$ in the tree size. Each overlap propagation enables us to compute the homography overlap distance between two nodes. There are N propagation targets and N lowest common parents overall. Thus, $O(N^2)$ top-down propagation steps are performed. The number of pairwise distance cal-

culations in different subtrees is also $O(N^2)$, because each distance calculation is done for a different pair of nodes. In order to increase the efficiency further, we propose the following speedups.

Tree Re-use. We can avoid repeating retrieval and propagation steps by memorizing the images within the kernel window, including the *border* images (blue dashed boxes in Fig. 4c). After shifting the medoid, we re-build the minimum spanning tree re-using previously determined homographies and overlap regions. In general, it will be necessary to expand the tree beyond the border, but the known nodes can be processed at much lower cost.

Basin of Attraction. A common speedup used in mean shift is to associate the *basin of attraction*, *i.e*. the points within a narrow radius around the mode, with the mode directly instead of performing an extra mode search for them. This is feasible because a mode search from a point very close to a mode will likely converge to the same mode. This speedup can be used in Iconoid Shift by removing the basin of attraction images of each iconoid from the seed set.

4. Experimental Results

We now present results achieved applying our approach on a challenging large-scale dataset. We show that our definition of iconoids suits the notion of iconic images and demonstrate that Iconoid Shift can automatically discover meaningful object clusters in a fully unsupervised way. Furthermore, we show how Iconoid Shift can be used to perform hierarchical scene summarization.

Dataset. We use a dataset of 500k images of Paris [24] collected from Flickr and Panoramio. The images were retrieved using a geographic bounding box query. As a result, they have a "natural" distribution as opposed to images retrieved using keyword queries.

Min-Hash Seed Generation. Chum *et al*. [4] propose to cluster touristic image collections by iterated query expansion [6] using min-Hash collisions as *seeds* from which clusters are grown. We use their approach to select the seed set of images S (Alg. 1), since it has been shown to yield good starting points for growing image clusters [24].

Does Iconoid Shift actually select iconic views?
Our first question is if our definition of iconoids fits the intuitive concept of an iconic image. Fig. 6 shows four typical runs of Iconoid Shift starting with views of landmark buildings taken at oblique angles or large distances (left column). Each run took three iterations to converge to an iconoid (right column), which typically show a frontal, centered and full view. Starting with a given view of a landmark building, Iconoid Shift tends to tilt, zoom and orbit around the object until it reaches a view that is favored by human photographers. As an interesting side effect, it automatically selects whether a portrait or landscape format photo fits the object

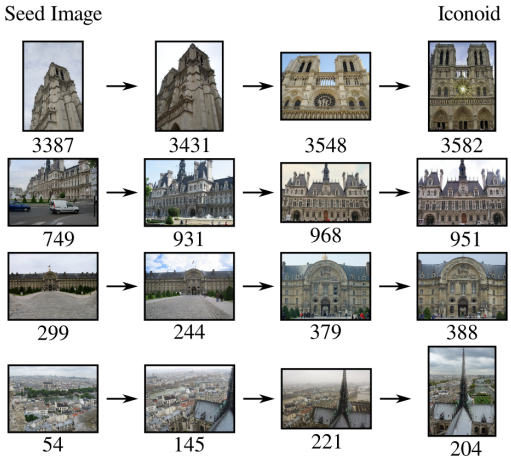


Figure 6: Examples of Iconoid Shift sequences with the support set sizes at each step.

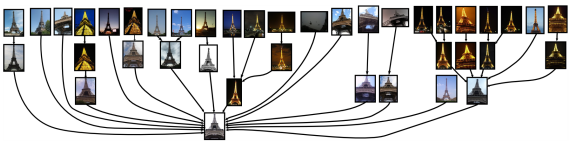


Figure 7: Iconoid Shift runs converging in the same iconoid.

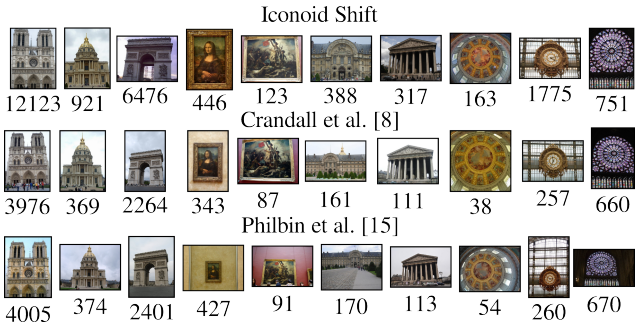
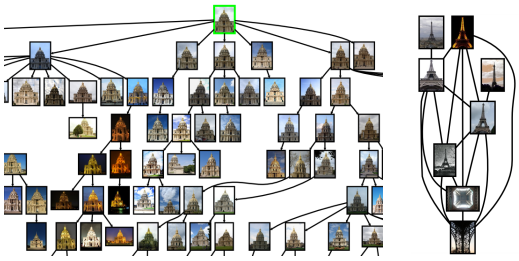


Figure 8: Comparison of iconoids (top) and [8] and [15]'s spectral clustering approaches, selecting the images with maximum (weighted) valence. Numbers denote the number of images associated with the iconic by membership of the support set (top) and adjacency in the matching graph (middle and bottom).

better, because a photo that is completely filled by the object has higher mutual overlap with its neighbors. The support set size (given below the images) is often higher for more "iconic" views, because the more typical a view is the more images overlap with it. However, since we optimize the mutual overlap and not the number of images in the support set, this number does not increase consistently.

Fig. 7 shows a number of Iconoid Shift runs starting from different views of the Eiffel Tower (leaves) that each converge in the same iconoid (root). The path from a leaf to the root shows the iterations of Iconoid Shift.

How does it compare to feature-based iconic selection?
A very popular approach for finding iconic images is to select the images with the highest feature-based similarity in their neighborhood [2, 8, 12, 15, 25]. We compare



(a) (b)

Figure 9: (a) Part of the minimum spanning tree of Les Invalides (921 images) with its iconoid (green border) at the root. (b) Overlapping clusters (represented by their iconoids) of the Eiffel Tower.

our work with two approaches: [8] build the full matching graph, weight each edge by the number of homography inliers, segment it using spectral clustering and select the image with the highest weighted valence in each cluster. [15] additionally merges the spectral clusters showing the same building by trying to propagate a homography between their images with maximum unweighted valence. A qualitative comparison is given in Fig. 8. In general, Iconoid Shift tends to select more central views than the (weighted) valence criteria, which are based only on feature similarity and does not have a geometric interpretation. The numbers show the neighborhood sizes w.r.t. the respective neighborhood criterion (membership in the support set vs. adjacency in the matching graph). We define neighborhood using the geometric overlap that is propagated independently of feature matches and thus discover more images of the same object than a plain feature-based matching.

How are the minimum spanning trees structured? Iconoid Shift returns both the iconoid and its minimum spanning tree (Fig. 9a). Since this tree was constructed by recursive image retrieval, its branch structure reveals the structure of the iconoid's neighborhood. For example, branches may contain specific views of the object or depict the object in certain lighting conditions. This has interesting applications such as navigating the scene of an iconoid by following paths in the tree.

Is the simple propagation scheme insufficient? In order to verify that the transitive overlap propagation scheme (Fig. 3b) is necessary to fully explore an iconoid's neighborhood, we compare it to the simple scheme (Sec. 2) that multiplies homographies along the path and determines the inlying feature matches. We use a smaller dataset of 100k images of Paris and initialize Iconoid Shift with a set of 25 seed images generated by Geometric min-Hash. The simple scheme discovered 17 clusters with an average size of 137.9, while the transitive scheme discovered 16 clusters with an average size of 230.8. Visual inspection showed that in general, the images discovered by the simple method cover a lower variety of viewpoints, because it relies on direct feature matches and thus on the invariance of the interest point descriptor and detector, while the transitive scheme propagates the overlap region independently of direct feature matches. Furthermore, computation using the simple scheme took 24x longer than the transitive scheme, because direct overlap computations are more costly and require local features to be loaded from disk.

(a) Grand Palais Entrance (b) La Madelaine (c) Notre Dame

Figure 10: 3D reconstructions from iconoid support sets.

Large-Scale Evaluation. In order to show that Iconoid Shift can automatically identify the tourist hotspots of an entire city, we apply it to the full 500k images of Paris. We generate a seed group S with Geometric min-Hash [5] using 5 min-Hash sketches of size 2, yielding 10,487 colliding images. We remove duplicates by applying a tf*idf threshold and filter out multiple seeds of the same object by building a pairwise matching graph of the min-Hash seeds, identifying its connected components and choosing one representative for each. Since we only build such components on the seeds, this step is inexpensive. We use this reduced set of 477 images as the seed set. We use the hinge kernel (Eq. 11) and set $\beta = 0.9$, *i.e.* all images in the support set of a medoid need to have at least 10% overlap with it.

Iconoid Shift identified 349 iconoids with a mean support set size of 627, covering a total of 76,787 photos. Due to the lack of a suitable ground truth, we cannot provide precision and recall statistics, but by visual inspection we did not find any false positives in the support sets except for those caused by timestamps, borders and logos that some users have added to their photos. Fig. 1 shows a map of the discovered iconoid support sets and some example iconoids of varying types and support set sizes. The top landmarks are the Eiffel Tower (7 iconoids covering 16,342 images), Notre Dame (4 iconoids covering 13,369 images) and the Arc the Triomphe (4 iconoids covering 7,764 images).

Because landmark buildings typically have several popular photo taking spots, Iconoid Shift often discovers multiple iconoids of the same building or object. These can easily be merged, *e.g.* for reconstructing buildings in 3D (Fig. 10), since their clusters typically overlap. An example is given in Fig. 9b. In contrast to a hard clustering on a landmark level, Iconoid Shift produces a much richer structure: For each landmark, we get a set of popular views and their relationships, as well as a fine-grained tree structure of different aspects of the photographed scenes (see above).

Figure 11: Hierarchical scene summarization by varying the kernel bandwidth. Each row shows the iconoids for a particular bandwidth.

Scene Summarization. When decreasing the kernel bandwidth, mean shift converges to smaller, more local modes. Similarly, in Iconoid Shift, a smaller kernel bandwidth usually leads to smaller, less important iconic images, such as certain details of a facade or entrance. Increasing bandwidth shifts the iconoids to more global iconic views. Fig. 11 shows a hierarchical summary created this way. Each row shows the set of iconoids for a given kernel bandwidth β.

5. Conclusion

In this paper, we proposed a novel algorithm for discovering popular views of landmark buildings and other objects in image collections. Our approach considers the task as a mode estimation problem, which is solved by applying a medoid shift search with our newly proposed homography overlap distance. Our experiments have shown that our approach discovers meaningful iconoic images and produces an overlapping cluster structure that gives rise to many interesting applications.

Acknowledgements This project has been funded by the cluster of excellence UMIC (DFG EXC 89). We thank Jan Hosang for his contribution to the idea of Iconoid Shift.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a Day. In *ICCV'09*, 2009.

[2] Y. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *in Proceedings of ACM Multimedia (MM 2010)*, 2010.

[3] T. Berg and A. B. Berg. Finding Iconic Images. In *CVPR'09 Internet Vision Workshop*, 2009.

[4] O. Chum and J. Matas. Large-scale discovery of spatially related images. In *PAMI*, 2010.

[5] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In *ICCV'07*, 2007.

[6] O. Chum, J. Philbin, J. Sivic, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *ICCV'07*, 2007.

[7] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *PAMI*, 24(5):603–619, 2002.

[8] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the World's Photos. In *WWW'09*, 2009.

[9] T. Denton, M. Demirci, J. Abrahamson, A. Shokoufandeh, and S. Dickinson. Selecting Canonical Views for View-based 3D Object Recognition. In *ICPR*, 2004.

[10] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a Cloudless Day. In *ECCV*, 2010.

[11] S. Gammeter, T. Quack, and L. Van Gool. I Know What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps. In *ICCV*, 2009.

[12] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. In *ECCV*, 2008.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *CVPR*, 2007.

[15] J. Philbin and A. Zisserman. Object Mining using a Matching Graph on Very Large Image Collections. In *ICVGIP'08*, 2008.

[16] T. Quack, B. Leibe, and L. Van Gool. World-Scale Mining of Objects and Events from Community Photo Collections. In *CIVR*, 2008.

[17] R. Raguram and S. Lazebnik. Computing Iconic Summaries for General Visual Concepts. In *CVPR'08 Internet Vision Workshop*, 2008.

[18] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter. In *ICCV*, 2009.

[19] Y. Sheikh, E. Khan, and T. Kanade. Mode-Seeking by Medoidshifts. In *CVPR*, 2007.

[20] I. Simon, N. Snavely, and S. Seitz. Scene Summarization for Online Image Collections. In *ICCV*, 2007.

[21] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*, 2003.

[22] A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *ECCV*, 2008.

[23] D. Weinshall, M. Werman, and Y. Gdalyahu. Canonical Views, or the Stability and Likelihood of Images of 3D Objects. In *ARPA Image Understanding Workshop*, 1994.

[24] T. Weyand, J. Hosang, and B. Leibe. An Evaluation of Two Landmark Building Discovery Algorithms. In *ECCV'10 RMLE Workshop*, 2010.

[25] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a Web-Scale Landmark Recognition Engine. In *CVPR*, 2009.