# Region Graphs for Organizing Image Collections

Alexander Ladikos[1], Edmond Boyer[2], Nassir Navab[1], and Slobodan Ilic[1]

[1]Chair for Computer Aided Medical Procedures, Technische Universität München
[2]Perception Team, INRIA Grenoble Rhône-Alpes

**Abstract.** In this paper we consider large image collections and their organization into meaningful data structures upon which applications can be build (e.g. navigation or reconstruction). In contrast to structures that only reflect local relationships between pairs of images we propose to account for the information an image brings to a collection with respect to all other images. Our approach builds on abstracting from image domains and focusing on image regions, thereby reducing the influence of outliers and background clutter. We introduce a graph structure based on these regions which encodes the overlap between them. The contribution of an image to a collection is then related to the amount of overlap of its regions with the other images in the collection. We demonstrate our graph based structure with several applications: image set reduction, canonical view selection and image-based navigation. The data sets used in our experiments range from small examples to large image collections with thousands of images.

## 1 Introduction

Dealing with large image collections has recently become a subject of interest in the vision community. It includes such diverse topics as 3D reconstruction [1, 2], canonical view selection [3, 4], image-based navigation [5] and image retrieval [6, 7] among others. While applications in this domain can be very different, a key issue that all must address is how to efficiently organize and handle the available and often redundant data. In image retrieval, for instance, state-of-the-art approaches deal with image datasets containing up to one million images [6] and even in 3D reconstruction applications the sizes of the image sets grow rapidly, reaching up to 150,000 images [2]. Most approaches in this field organize images with graphs where edges relate images that share information and edge weights depend on the application. For instance [8] uses a graph where the edge weight is based on the covariance of the camera positions, while [4] weights the edges by the number of inlier matches between images. The resulting data structures reveal little on how informative an image is with respect to all other images. In this work, we take a different strategy and propose a data structure, region graphs, that encodes spatial relationships between an image and a collection of images. This provides a basis upon which various applications can be build, navigation or reconstruction for instance, where not all but only the most informative images are of interest.

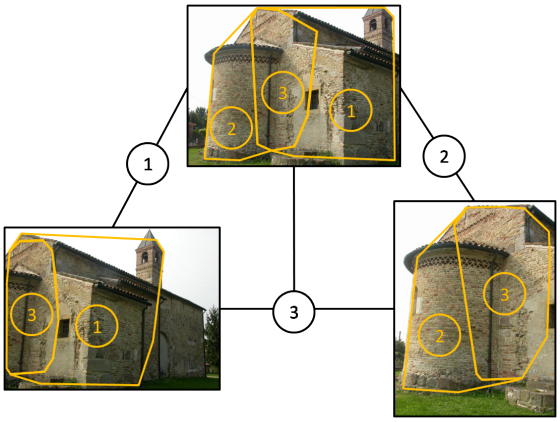



**Fig. 1.** Region graph for three images. Overlapping regions are denoted as 1, 2 and 3.

The central idea behind our approach is to use image regions and their redundancies over an image set to define a global hierarchy in the set. More precisely, we consider the overlap between images, where we consider the overlap to be the image regions which contain the same part of the scene. This is a natural criterion, based on objective evidence, that does not require any information about the 3D structure of the scene. It also adapts to the sampling of the scene given by the images. The overlapping regions are then used to build a graph relating all images spatially. The graph contains two kinds of nodes, one representing images and the other representing overlapping regions. Each region is connected with an edge to the images it is contained in. This means that region and image nodes are alternating on any given path through the graph. Using this graph we can efficiently represent the spatial relationship between the regions and images and identify redundancies over regions. This allows us to model the importance of regions shared by many images and to identify less important regions shared only by few or even no images. These less important regions are often small and not very essential to the scene. They typically contain background or other irrelevant information.

Figure 1 shows an exemplary region graph constructed from a three image data set. There are three image nodes and three region nodes in the graph, representing the images and the distinct overlapping regions, i.e. regions visible in a set of images. Using the region graphs as a basis we build applications for image set reduction, canonical view selection and image-based navigation. We tested our method on several real data sets ranging from a few dozen to thousands of images. The results obtained show that the data structure we propose reveals intrinsic properties of an image set that are useful for various applications.

In the remainder of the paper we first discuss the related work in section 2. We then proceed to describe the construction of the region graphs in section 3 and show some exemplary applications built on them in section 4. We present results in section 5 and conclude with section 6.

## 2 Related Work

In the last few years many papers dealing with the issue of large image collections have been published. Most of them focus on specific applications, for instance image retrieval [6, 7, 9] or 3D reconstruction [1, 4, 2]. In the 3D reconstruction literature one of the first major works on this topic was the Photo Tourism project [1]. In that paper a large set of images taken from Internet photo collections is used for performing a point-based 3D reconstruction of the scene. An exhaustive pairwise matching followed by an incremental bundle adjustment phase have been used both for the reduction of the image set and for 3D reconstruction. Follow-up work focused on navigating through large image collections [5], summarizing the scene by selecting canonical views [3] and speeding up the initial reconstruction process by building skeletal graphs over the image set [8]. While an image graph was used for instance in [8] it was designed for the goal of finding a better subset of images for the initial reconstruction. Li *et al.* [4] presented an application for performing reconstructing and recognition on large image sets. They construct a so called iconic scene graph which relates canonical views of the scene and use it for 3D reconstruction. The edge weights used are the number of inlier matches. Recently Farenzena *et al.* [10] proposed a hierarchical image organization method based on the overlap between images. The overlap is used as an image similarity measure used to assemble the images into a dendrogramm. The hierarchy given by the dendrogramm is then used for a hierarchical bundle adjustment phase. In this regard that work is interesting, because it also considers a global criterion. However, it is focused on Structure from Motion and not on defining global representations of image collections. Schaffalitzky [11] *et al.* also present some work dealing with handling large unordered data sets. They focus on the task of performing a 3D reconstruction from unordered image sets and only briefly mention image navigation, which they base on homographies.

*Contribution* Most existing work organizes images with respect to the application, which is often 3D reconstruction. We follow a different strategy and organize images with respect to the regions they share. This allows us to score images according to the information they bring and without 3D reconstruction. Subsequent applications can then easily build on the region graph structure, even navigation as shown later in section 4. We are not aware of any attempt to build such an intermediate structure based on 2D cues only. We think that these structures will become a key component when dealing with large and highly redundant image datasets.

## 3 Building Region Graphs

In this section we describe how to construct region graphs. The most important construction principle is to identify overlapping regions in the images. Overlapping regions are regions in different images showing the same part of the scene. Figure 2 gives an example. For instance region 1 is an overlapping region shared

by images A, B and E. To identify this overlapping region, the intersection of the overlap between image A and E and the overlap between image B and E has to be computed. Each overlapping region is represented as a *region node* in the graph. The images are represented in the graph as *image nodes*. Each region node is connected to the images in which it is detected. In the example of Figure 2 this means that node 1 representing region 1 is connected to the nodes of images A, B and E. In the following sections the graph construction process is described in more detail. The construction process is summarized in algorithm 1.

### 3.1 Identifying Overlap Between Images

The first step in the graph construction is to identify the overlap between the images. This is accomplished in a multi-step process. First we extract features using a scale-invariant interest-point detector on all input images [12]. We then match the features among the images. Since we are dealing with very large image sets, performing an exhaustive pairwise matching is computationally infeasible. Therefore we use vocabulary trees [13] to perform a preselection among the images (in our experiments we use the implementation provided by [14]). For every image we retrieve the $k$ (we use $k = 10$ in all our experiments) most similar images using the vocabulary tree.

This preprocessing step significantly reduces the size of the set of image pairs which have to be matched. The matching is performed using the standard SIFT distance ratio on the descriptors and the resulting putative matches are pruned using epipolar constraints in a RANSAC framework. Given the feature correspondences between two images we compute the convex hull spanned by the matched features in each image. This is illustrated in Figure 3. The area enclosed by the convex hull in each image is the overlap between the two images.

### 3.2 Identifying Overlapping Regions

After performing the matching, we generally obtain several different convex hulls per image, one per matched image. In general these convex hulls will overlap with

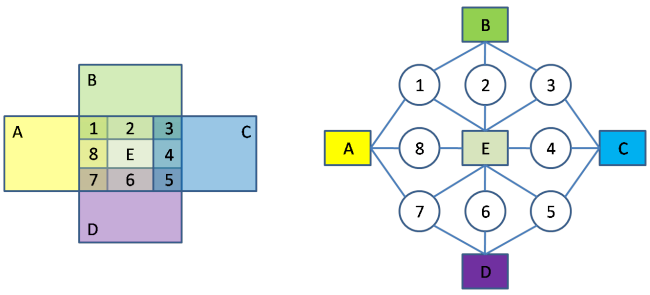



**Fig. 2.** Graph construction for a synthetic example containing five images (A to E) which create 8 different overlap regions.

each other. We want to identify each overlapping region created by the intersection of these convex hulls. In the following let $CH_i^j$ be the convex hull spanned in image $i$ by the features matching image $j$. To determine unique overlapping regions we assign each $CH_i^j$ a label $(i, j)$ to indicate that this region is shared by images $i$ and $j$. When two regions $CH_i^k$ and $CH_i^l$ overlap, the common region will receive the label $L = (i, k, l)$. After performing this labeling for all convex hulls every intersection will have an associated label $L$. The image is then subdivided into regions sharing the same label. While it is possible to perform these computations directly on the image by discretizing the convex hulls, we chose to perform the computations purely geometrically by representing the convex hulls as polygons and using CGAL to perform the intersection operations. This has the advantage of being image resolution independent and does not require to allocate a discretization space for every image, which would be very memory intensive for large image data sets. Finally every identified region is merged into a region list storing its label and the images in which it was detected.

### 3.3 Constructing the Region Graph

After all overlapping regions have been identified, the region list contains all the information needed to build the region graph. It is constructed by inserting one image node per image and one region node for every entry in the region list. The region nodes are subsequently connected to the image nodes specified in the region list. The weight of the edges connecting the region nodes to the image nodes is application specific. One generic choice is to assign the normalized size of the region, defined as the size of the overlapping region divided by the image size, as an edge weight. This is the edge weight which is used in most of our experiments.

## 4 Using Region Graphs

In this section we discuss several applications based on the proposed region graphs. The first application is image-based navigation which allows the user to

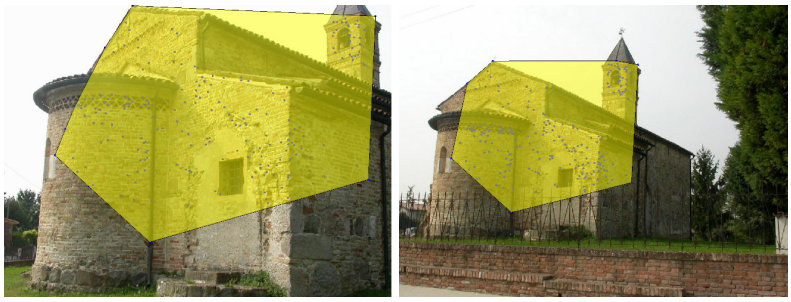


**Fig. 3.** The convex hull of the set of matched features between two images defines the regions considered during graph construction.

**Algorithm 1** Graph Construction

```
Extract feature points on all images I
Use a vocabulary tree to select the k most similar images for each image
Perform robust matching
for each image i in I do
    for each image j matched to i do
        Compute the convex hull CH_i^j and assign it the label (i, j)
    end for
    Intersect the convex hulls in image i to obtain the overlapping regions
    Add overlapping regions into region list
end for
for each image i in I do
    Create an image node in the graph
end for
for each region entry l in the region list do
    Create a region node and connect it to the image nodes of the images in which
    it was detected
    Set the weight of the outgoing edges according to the application criteria, e.g.
    the normalized size of the region
end for
```

traverse the image set in a spatially consistent way. The second application is image set reduction. Its goal is to reduce the size of the data set while retaining as much information as possible. The final application we are considering is canonical view selection. In this application we want to find a small orthogonal subset of images which summarizes the whole image set.

### 4.1 Image Set Reduction

The goal of image set reduction is to remove redundant and non-contributing images from the data set. In [8] for instance a subset of an image set is selected for performing a 3D reconstruction. However, the graph structure and edge-measure were application specific and based on the covariance of the camera positions. We would like to define a more general measure for the information content of an image. Intuitively an image which contains many regions shared with other images is more important for the data set than an image having little overlap with the other images in the data set. We therefore formalize an information criterion for an image $i$ and its associated image node $v_i$ in the region graph as

$$\rho(v_i) = \sum_{r \in N(v_i)} \sum_{e \in E(r)} w(e) \tag{1}$$

where $N(v_i)$ is the set of neighboring region nodes of image node $v_i$, $E(r)$ is the set of edges in the region graph connected to node $r$ and $w(e)$ is the weight of edge $e$. The intuition behind this information criterion is that an image which contains many regions which are also present in many other images is more

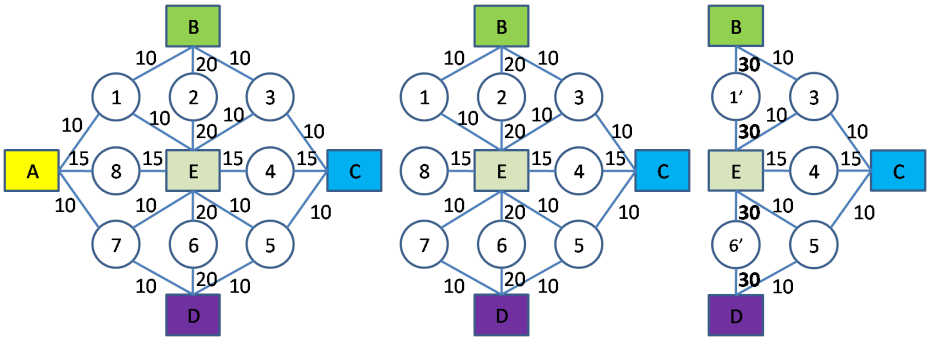



**Fig. 4.** Removal process on the synthetic example given in Figure 2. The image with the least score among all images is removed first (left). Leaf nodes created by the removal of the image are removed (middle). Newly created duplicate paths are joined. (right).

important than an image which only contains few regions shared with few other images. The choice of which images to remove is directly related to this criterion. At each step of the removal process the image with the smallest image score is removed. In Figure 4 we give an example of the image removal process in the graph. Once the image to be removed has been identified, its corresponding node and all incident edges are removed from the graph. The resulting graph might then contain leaf nodes (node 8 in the example) which are also removed. Due to the removal of an image it can also happen that two previously distinct regions collapse into one. This can be seen in the graph through the existence of several identical paths between two image nodes (paths $E \rightarrow 1 \rightarrow B$ and $E \rightarrow 2 \rightarrow B$ in Figure 4 (middle)). These paths are joined and their edge weights summed up to obtain a region node representing the new region. All these computations can purely be based on the graph. No recomputations are needed. This is due to the explicit representation of regions in the graph. If only images were represented in the graph it would have to be recomputed after every image removal.

### 4.2 Canonical Views

Canonical views are views which are of high importance in a given image set. They show parts of the scene which are captured in many images (e.g. because they are considered to be very important). We want to automatically find these important parts of the scene and select one representative view, i.e. the canonical view, for each of them. Some previous work on this subject was done in [3]. In that work the criterion for selecting a canonical view was based on the visibility of the points in the scene. A canonical view was defined to be an image which is very different from all other canonical views in terms of the scene points it observes. This criterion was optimized by a greedy approach. We have a similar definition of canonical views. However, we do not assume any explicit visibility

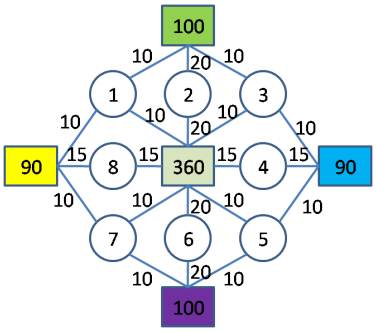


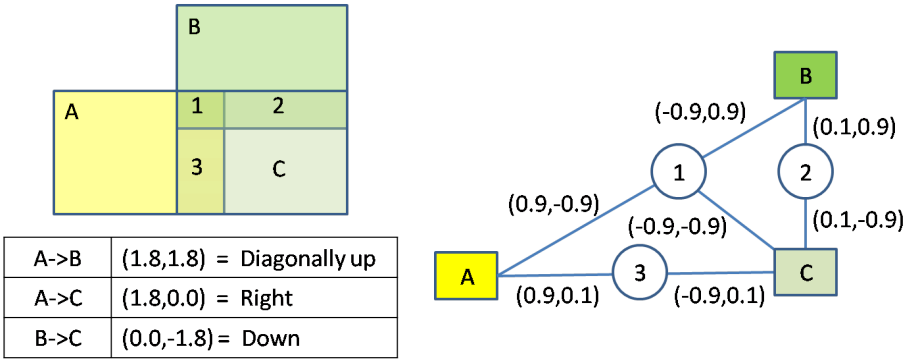

**Fig. 5.** Canonical view selection on the synthetic example given in Figure 2. The numbers inside the image nodes indicate the image score computed according to equation 1. The central image has a maximum score in its neighborhood and is therefore selected as the canonical view.

information to be available. We also do not perform a greedy optimization, but instead deduce the canonical views directly from the region graphs.

Intuitively the images having the highest amount of overlap with the image set should be selected as canonical views. However, we would like to avoid selecting multiple images of the same part of the scene. One natural way of including this constraint is to find maxima over the graph. Each image node $v_i$ is assigned a weight using the score function given in equation 1. Only the nodes which have a score bigger than all their neighboring image nodes are selected. These nodes then constitute the canonical views. The neighboring image nodes are defined to be all the image nodes which are only separated by a region node, i.e. two images are considered to be neighbors in the graph when they share a common region. Figure 5 gives an example.

### 4.3 Image-based Navigation

The goal of image-based navigation is to allow the user to traverse the image set in a spatially consistent order. For instance the user can choose to view the image to the right or to the left of the current image. In order to allow such a navigation the spatial relationship among the images has to be determined. While some prior work [5] assumes the availability of a 3D scene reconstruction we base the navigation purely on the images. This is achieved by considering the spatial positions of matching regions in the images. To represent this information in the graph we augment the edges with information about the spatial relationship of the associated nodes. In practice we assign each edge in the region graph a three-dimensional vector $(x\,y\,z)^\top$ which describes the relative position and scale of the region within the image. The position inside the image is specified with respect to the image center and normalized to the range $[-1;1] \times [-1;1]$. The first two components of the vector describe the horizontal and vertical position, while the third one represents the scale. They are computed by considering the




| | |
|---|---|
| A->B | (1.8,1.8) = Diagonally up |
| A->C | (1.8,0.0) = Right |
| B->C | (0.0,-1.8) = Down |

**Fig. 6.** Illustration of the image navigation. The region graph is augmented with the computed relative positions of the regions within the image. For determining the relative motion between two images all shared regions are considered and their relative motions are averaged. The resulting relative motions between the images are shown in the table. Note that for clarity scale is not considered in this example.

position of the center of gravity of the convex hull in the image. Let $\mathbf{g}_i = (g_i^x\, g_i^y)^\top$ represent the center of gravity of the region in image $i$ and let $w_i$ and $h_i$ be the size of the image in pixels. Then the relative position of the convex hull inside the image is given by

$$\mathbf{p}_i = \begin{pmatrix} \frac{2g_i^x - w_i}{w_i} \\ \frac{2g_i^y - h_i}{h_i} \end{pmatrix} \tag{2}$$

To represent the scale we consider the relative area of the region with respect to the image area. This makes us independent of the image resolution. Let $A_i$ be the number of pixels in the convex hull and $I_i$ the total number of pixels in image $i$. Then the scale is given by

$$s_i = \frac{A_i}{I_i} \tag{3}$$

The region movement (position and scale) for a region shared by images $i$ and $j$ is computed as

$$x_{i->j} = -(p_j^x - p_i^x) \tag{4}$$
$$y_{i->j} = p_j^y - p_i^y \tag{5}$$
$$z_{i->j} = s_j - s_i \tag{6}$$

To navigate the user specifies a spatial movement in the image plane (two dimensions) and a zoom-in/zoom-out movement (one dimension). This results in the desired movement vector. To find the next image to move to, the movement between the current and all neighboring images is computed. Given two images the relative movement is given by the average of the region movement of the regions shared by the images. The image whose region movement agrees most

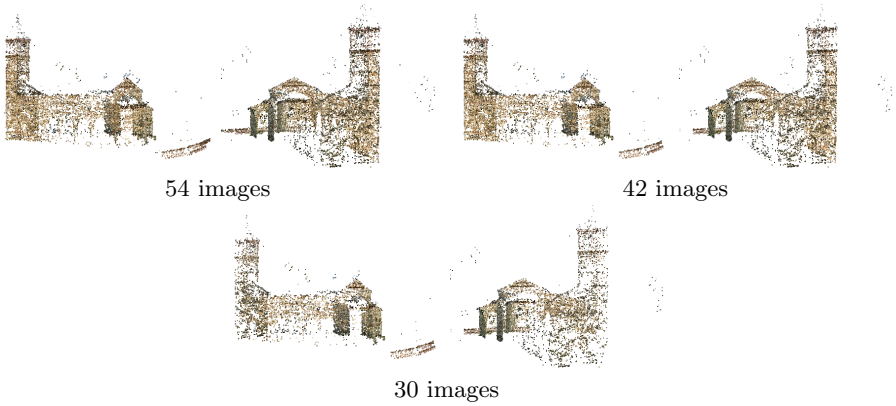



**Fig. 7.** Image set reduction for the *pozzoveggiani* data set. The first row shows the full reconstruction, while the second and third row show the results after removing 12 and 24 of the 54 images respectively.

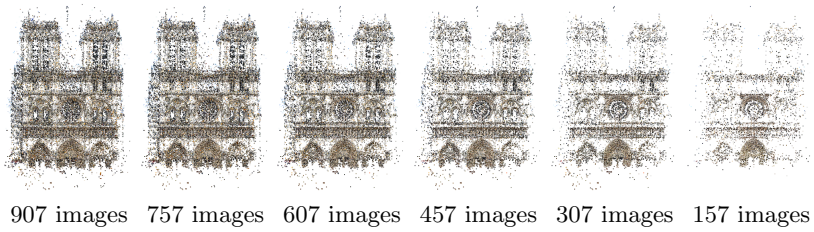



**Fig. 8.** Image set reduction for the *Notre Dame* data set. The first image shows the full reconstruction (907 images). Each following image shows the result after 150 images were removed from the previous reconstruction.

with the user motion (in the sense of the dot-product) is then displayed to the user. Figure 6 gives an example of how the relative movement between images is computed using the shared regions. Since we explicitly represent the regions in our graph it is also possible for the user to select a specific region of interest inside the image and to perform the navigation with respect to this region instead of the whole image.

## 5 Experimental Results

To validate our approach we performed experiments on several data sets of different sizes. In the following we will first briefly describe each data set used and then show results for the different applications we are proposing. The first two data sets we used were provided by [10]. The *pozzoveggiani* data set contains 54 images of a church and the *piazzaerbe* data set contains 259 images of a big town square. The other data set we used was the *Notre Dame* data set provided

**Fig. 9.** Canonical views for the *pozzoveggiani* data set. One image was selected for each side of the church.

**Fig. 10.** Canonical views for the *pozzoveggiani* data set as produced by [3]. The parameters for obtaining this result had to be manually adjusted until a reasonable result was obtained.

by [4]. It contains 6248 images of the Notre Dame cathedral in Paris collected from Flickr.

The first step common to all application is the construction of the region graph. The construction times (excluding feature extraction and matching) were 1 minute for *pozzoveggiani*, 3 minutes for *piazzaerbe* and 38 minutes for *Notre Dame* on a 2.66 GHz Intel QuadCore CPU (only one core was used). Most of the time was spent on intersecting the convex hulls.

### 5.1 Image Set Reduction

To show the validity of the reduction we first perform a 3D reconstruction with the full data set and then compare it to a reconstruction on the reduced data set. Figure 7 shows the results for the *pozzoveggiani* data set. The first row shows two views of the reconstruction obtained on the full data set, while the next two rows show the results obtained after removing 12 and 24 images respectively. While the point cloud does get sparser the whole structure is still present.

Figure 8 shows the results we obtained on the *Notre Dame* data set. We computed the connected components of the region graph and used the biggest one (907 images). The first image shows the full reconstruction. Each of the

**Fig. 11.** Canonical views for the *piazzaerbe* data set.

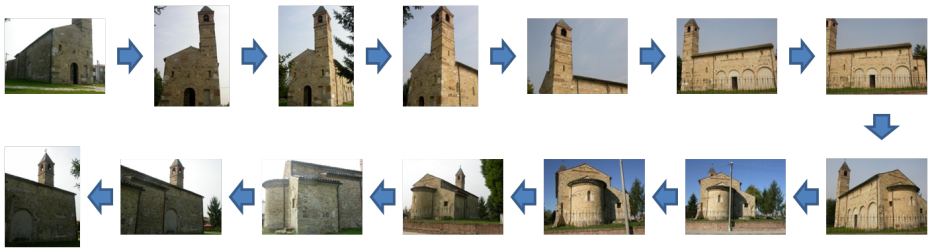


**Fig. 12.** Image-based navigation on the *pozzoveggiani* data set. Starting from the top left image the user always moves to the right, thereby circling the church once.

following reconstructions was obtained by removing 150 images from the previous one. Again the point cloud gets sparser, but the overall structure of the scene is retained.

### 5.2 Canonical Views

The results of the canonical view selection on the *pozzoveggiani* data set are shown in Figure 9. One view is selected for each side of the church. To compare to previous work we implemented the canonical view selection method described by Simon *et al.* [3]. The results of their method are shown in Figure 10. They are comparable to ours. The first four canonical views are virtually identical, while the last two are not very essential to the scene. Since Simon's method uses two tuning parameters, it was necessary to manually adjust them until a reasonable result was obtained. Their method also requires the availability of visibility information for each scene point, which is not always easy to obtain. Our method on the other hand is parameter free.

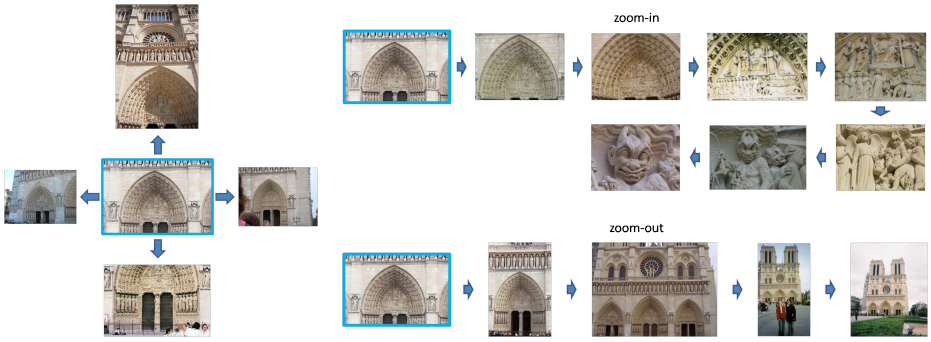



**Fig. 13.** Image-based navigation on the *Notre Dame* data set. The user starts with the highlighted image and then performs several navigation operations resulting in the shown images. The images on the left show the results of a spatial navigation (left, right, up and down) while the images on the right show the results of zooming in and out respectively.

The results of the canonical view selection on the *piazzaerbe* data set are shown in Figure 11. The selected images are very distinct from each other. Only the fountain and the pagoda are seen twice in the images. However, they are pictured from approximately opposite sides and have a completely different background.

Since we initially only use a sparse set of matches (i.e. we do not match every image to every other image), the region graph is also only sparsely connected. This means that similar images might not be connected in the region graph. The effect of this is that similar images might be selected as canonical views. Therefore we apply the canonical view selection twice. Once on the initial sparse graph and then on the obtained canonical views after performing an exhaustive pairwise matching on them. This is generally not very computationally expensive, since the number of canonical views is comparatively small compared to the size of the original data set. Optionally a vocabulary tree could be used to speed up the matching.

### 5.3 Image-based Navigation

Figure 12 shows the results for image-based navigation obtained on the *pozzoveggiani* data set. The user starts with the top left image and then continues to move to the right, circling the church once.

Figure 13 shows the results of an image-based navigation on the *Notre Dame* data set. On the left the user starts with the highlighted image and then navigates in the direction of the arrows (left, right, up and down). On the right the user performs a zoom-in and a zoom-out movement respectively. Note the number of scale levels traversed during the zoom-in and zoom-out operation.

## 6 Conclusion

We presented a novel framework for organizing large spatially related image collections. Our approach is based on the overlapping regions between multiple images. We represent these regions and the images in a graph and use this graph as a foundation for several different applications related to organizing large image collections, such as image-based navigation, image set reduction and canonical view selection. Using these applications we presented results on several image sets of different sizes, showing the validity of our image organization approach.

## References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. International Journal of Computer Vision **80** (2008) 189–210
2. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: International Conference on Computer Vision. (2009)
3. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: International Conference on Computer Vision. (2007)
4. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: European Conference on Computer Vision. (2008)
5. Snavely, N., Garg, R., Seitz, S.M., Szeliski, R.: Finding paths through the world's photos. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) **27** (2008) 11–21
6. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European Conference on Computer Vision. (2008)
7. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008)
8. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008)
9. Whyte, O., Sivic, J., Zisserman, A.: Get out of my picture! internet-based inpainting. In: British Machine Vision Conference. (2009)
10. Farenzena, M., Fusiello, A., Gherardi, R.: Structure-and-motion piepline on a hierarchical cluster tree. In: IEEE International Workshop on 3-D Digital Imaging and Modeling. (2009)
11. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "how do I organize my holiday snaps?". In: European Conference on Computer Vision. (2002)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
13. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition. (2006)
14. Fraundorfer, F., Wu, C., Frahm, J.M., Pollefeys, M.: Visual word based location recognition in 3D models using distance augmented weighting. In: Fourth International Symposium on 3D Data Processing, Visualization and Transmission. (2008)