Merging the Unmatchable: Stitching Visually Disconnected SfM Models

Andrea Cohen, Torsten Sattler, Marc Pollefeys
Department of Computer Science, ETH Zurich, Switzerland
{andrea.cohen,torsten.sattler,marc.pollefeys}@inf.ethz.ch

Abstract

Recent advances in Structure-from-Motion not only enable the reconstruction of large scale scenes, but are also able to detect ambiguous structures caused by repeating elements that might result in incorrect reconstructions. Yet, it is not always possible to fully reconstruct a scene. The images required to merge different sub-models might be missing or it might be impossible to acquire such images in the first place due to occlusions or the structure of the scene. The problem of aligning multiple reconstructions that do not have visual overlap is impossible to solve in general. An important variant of this problem is the case in which individual sides of a building can be reconstructed but not joined due to the missing visual overlap. In this paper, we present a combinatorial approach for solving this variant by automatically stitching multiple sides of a building together. Our approach exploits symmetries and semantic information to reason about the possible geometric relations between the individual models. We show that our approach is able to reconstruct complete building models where traditional SfM ends up with disconnected building sides.

1. Introduction

Despite recent advances that make Structure-from-Motion (SfM) methods more scalable [1, 15] and robust, *e.g.*, to challenges posed by repetitive structures [3,4,6,23], the fact that there might not always be enough images to obtain a single reconstruction remains a fundamental problem. For humans, determining the spatial arrangement of the different sub-models is often rather easy, indicating that even visually disconnected components contain information about their spatial relations. Yet, few approaches exist that attempt to automatically merge these components. There are multiple reasons why a SfM model can disconnect into multiple individual components: e.g. images obtained from photo-collection communities such as Flickr tend to mostly concentrate on a small set of iconic viewpoints [22]. In general, for popular landmarks, many pictures are taken of the front, few photos depict the sides and even fewer the back, and the changes of viewpoint are usually too large between different sides. Consequently, there is often not enough visual overlap between the different models to connect them through feature matching. Even in a controlled setting, where an expert familiar with SfM takes photos with the explicit goal of reconstructing the scene, it might not always be possible to obtain enough data for a complete reconstruction (*c.f*. Fig. 1). Trees or vegetation might block the view on different parts of a structure, preventing feature matching from finding enough correspondences to connect the parts. Convex building corners also represent a difficult case, especially in narrow environments where it is hard to take images with enough visual overlap while guaranteeing a wide enough baseline to enable a stable reconstruction. This problem is magnified in crowd-sourced reconstruction efforts [17], where non-expert users often find it hard to take enough pictures to enable the reconstructions of corners [16]. In addition, certain areas might not be accessible or might be hidden behind trees or vegetation.

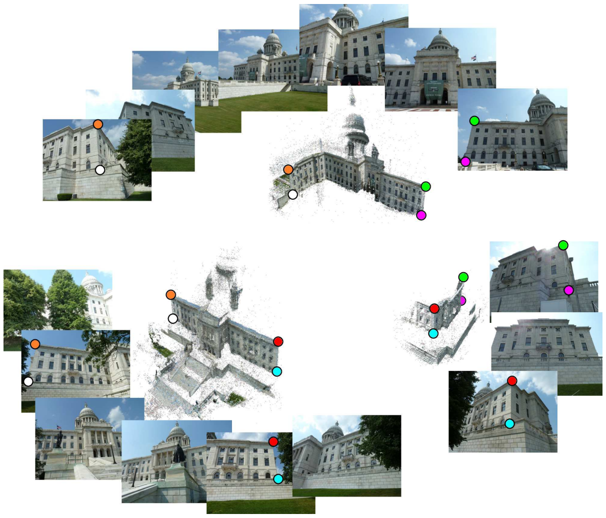


Figure 1. The Providence dataset from [5] disconnects into multiple sub-models that cannot be connected by feature matching due to trees blocking the view and missing images. Our method is able to automatically merge the sub-models by recovering their relative scales and joining them at the colored points.

While a reconstruction can disconnect into multiple sub-models due to missing visual overlap, we notice that it is often possible to obtain (nearly) complete reconstructions for individual sides of a building. In this paper, we consider the problem of automatically stitching 3D models corresponding to building sides into a single reconstruction. As our main contribution, we present a novel combinatorial approach for automatically solving this loop-closure problem that does not require temporal information. We use the symmetries and repetitive structures usually found on different sides of a building to recover the relative scales of the sub-models by exploiting feature matches discarded during the reconstruction process. We propose a method that detects *connection points* at which individual components can potentially be connected. We use semantic information to formulate a novel free-space compatibility measure in image space, enabling us to detect whether two sub-models can potentially be connected. A symmetry prior enables our system to help retain the more likely of multiple possible loop-closures. We show that our approach is robust enough to generate plausible results even if the individual sub-models are not perfectly reconstructed. To our knowledge, we are the first to tackle this model merging problem.

The rest of this paper is organized as follows. Section 1 presents a review of the related work. Sec. 2 analyzes the problem of merging visually disconnected components in more detail. Sec. 3 describes our novel model merging approach, which we evaluate in Sec. 4.

Related Work The ambiguities induced by repetitive and symmetric scene elements cause significant problems for SfM systems. To resolve these ambiguities, [23, 25] reason about missing correspondences. Exact duplicate structures are handled by enforcing a consistent epipolar geometry [26] or by jointly optimizing the geometry and inferring wrong matches [7, 12]. [6] uses conflicting observations between apparently related images to disconnect wrongly attached parts in a SfM reconstruction. The resulting sub-models are then connected using feature matching. Larger differences in viewpoints can be handled through viewpoint normalization prior to feature extraction [28] or by rendering the point clouds from novel views [14].

[4] exploits symmetries and repetitions to improve the quality of 3D models by explicitly incorporating these relations into bundle adjustment. Similarly, [3] incorporates symmetry detection into the SfM process.

In the context of efficient large-scale SfM, [15, 17] automatically register multiple models onto building outlines, using GPS priors for initialization. Approaches that align 3D points to 2D lines, such as [8], fail if no corners are present since registering 3D points on a plane against a 2D line is an ill-posed problem. [21] uses a digital surface model to align multiple individual reconstructions that do not share any overlap. [20] shows that even if feature matching between images of different components is not possible, potential overlap of the corresponding dense models can be used to compute an alignment. In contrast, we explicitly consider the case where no such overlap is available. [18] orders a set of single-view façade reconstructions based on visual overlap between the images, accelerating SfM and dense multi-view stereo by exploiting the resulting geometric information. [19] uses existing 3D models and additional street view imagery to accurately geo-register 3D models.

[11] registers disconnected reconstructions of an indoor environment onto a floor plan by reasoning about the temporal consistency of the movement of people between rooms. [2] uses semantic labelings to reconstruct a piecewise planar floor plan for indoor scenes. As in this paper, their goal is to recover the overall appearance of the scene instead of accurately reconstructing the scene in detail.

Similar to our approach, [13] identifies the contours of an object by detecting sky pixels. These contours are used to prevent filling the sky during Poisson reconstruction from multiple depth maps. In contrast, we use semantic labels to formulate free-space compatibility directly in image space.

2. The Model Merging Problem

Given a set of visually disconnected SfM reconstructions corresponding to different sides of a building, we define the *model merging problem* as the problem of stitching together all components to obtain a faithful representation of the true scene geometry. Since 3D reconstruction is usually only possible up to an unknown scale factor, solving the model merging problem includes determining the relative scales between the components. In general, resolving the scale ambiguity between sub-models is impossible as they do not share any visual overlap. However, man made structures such as buildings often exhibit repeating and symmetric structures. SfM approaches try to eliminate ambiguous structures [6, 7, 12, 25, 26] as they often result in collapsing spatially unrelated façades with similar appearance into a single part of the model [6] or hallucinating structures not contained in the scene [23]. However, these ambiguous structures also contain valuable information as they enable us to establish point correspondences between visually disconnected sub-models. In turn, these point correspondences can be used to both recover the relative scales between the sub-models and to align them along the vertical direction. In this paper, we thus make the assumption that the building that we want to reconstruct contains such repeating and symmetric structures.

As mentioned above, the motivation for this paper is to provide an automatic loop-closure mechanism that generates a single, consistent reconstruction of a building from visually disconnected sub-models. Assuming that each sub-model is perfectly reconstructed and that there are no gaps, we should be able to detect a clear boundary or corner for each of them. Merging sub-models thus corresponds to the problem of determining which sub-models have to be connected at their boundaries or corners and under which angle. As we are interested in reconstructing buildings, we make a Manhattan world assumption to limit the set of possible angles to multiples of 90°, resulting in a combinatorial optimization problem. Notice that there might be multiple combinations of sub-models that close the loop. We found that opposite sides of a building often contain aligned symmetric structures. If we detect such symmetry planes, we can thus use them to determine the more plausible between multiple solutions.

In theory, a valid combination of sub-models should not contain any two sub-models that intersect each other. Thus, free-space constraints could be used to distinguish between valid and invalid combinations. In practice, it can happen that the sub-models actually have a slight overlap at their boundaries (*c.f.* Fig. 1) that would violate free-space constraints based on the 3D scene geometry. In order to avoid this problem, we assume that some of the images showing the boundaries of the sub-model observe a silhouette of the building against the sky. Thus, we can formulate free-space compatibility by determining whether the points from one sub-model project into the sky.

3. Automatic Model Merging

Based on the assumptions detailed in the last section, we derive our approach to merging multiple sub-models into a single, consistent model. The input to our method is a set of SfM reconstructions [26, 27] of the same building that do not share enough visual overlap to merge them based on feature matching and epipolar geometry estimation, as well as the images that were used to compute the reconstructions. These non-overlapping reconstructions will be referred to as sub-models. Our approach consists on the following stages:

- Estimate the scale and relative height for all sub-models. The goal of this step is to have all models on the same axis-aligned reference frame. The 3D alignment of models then becomes a 2D problem. This stage is described in section 3.1
- Find the best connection points for each sub-model. This stage is described in section 3.2 and it aims at finding the boundaries or corners of each sub-model as candidates for stitching two sub-models.
- Generate all possible pairwise transformations based on the estimated connection points and a Manhattan world assumption. These hypotheses are then evaluated using semantic labels in order to filter out improbable transformations (see section 3.3)
- Exhaustively generate all possible fully connected reconstructions by combining the surviving pairwise transformations. Choose the best ones according to a loop-closure and a symmetry-alignment criterion. This is detailed in section 3.4

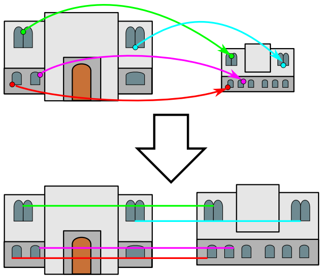


Figure 2. Relative scale and height estimation.

3.1. Resolving Relative Scales

We exploit symmetries and repetitions detected in each of the sub-models to define a natural coordinate frame for each component. This coordinate system is aligned with the vertical axis, *i.e.*, façades represented by this model stand on the xy-plane and are mostly aligned with the x direction. We compute this alignment using feature matches that were discarded during the SfM process as geometrically inconsistent. This procedure was proposed and fully explained in [4]. We use this coordinate system instead of the arbitrary coordinate frame of the original reconstruction as it simplifies the problem of aligning different components. This reduces the degrees of freedom (DOF) for the similarity transformation that aligns two models from 7 DOF to 4 DOF plus a discrete set of orientations (under a Manhattan world assumption), as we only need to find a 3D translation, a rotation around the vertical direction, and a scaling factor.

Relative scale and height estimation We want to further reduce the degrees of freedom of the similarity transformation such that we only need to estimate a single rotation angle and a 2D translation. Therefore, we need to be able to estimate the relative scale between different sub-models and the z-translation (vertical translation) between them. In order to do so, we rely on feature matches between different sub-models. More specifically, we perform the following steps:

- For each pair of sub-models, we do a 3D-3D matching using the SIFT-features of the keypoints (and their mirrored counterpart) from which each 3D point was reconstructed. This results in a set of 3D-3D putative correspondences. The assumption is that these correspondences represent repetitive structures across sub-models that should have the same scale and height.
- Sequential RANSAC for 3D similarities is performed on the putative matches. A set of 3D similarities (3D translation, rotation and scale) is obtained. This is illustrated in figure 2.

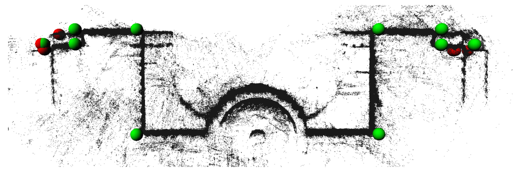


Figure 3. Connection points for sub-model 0 of the University dataset. Connection points resulting from the intersection of planes are colored in green, connection points created from the boundaries of planes are shown in red.

- We do a consistency check between all triples of sub-models that were successfully matched. This way, we get rid of similarities that are not consistent in a loop (a threshold t computed automatically from the point cloud is used).
- Similarities that survived the loop-consistency check are used to propagate a scale and a vertical translation between pairs of sub-models that didn't have enough 3D matches for similarity estimation, again using triples.

It is not always possible to find matches between all pairs of components. Still, in our experiments, we were always able to find a set of relative alignments that form a graph with only a single connected component. We were, thus, able to propagate the transformations and select an overall scale and height alignment that is consistent among all sub-models, finding a consistent coordinate frame.

3.2. Finding Connection Points

Without any knowledge of the size of the gaps in the reconstruction, it is impossible to tell how far different components should be placed apart. We therefore make the assumption that there are no gaps between sub-models, *i.e.*, that we get a good approximation of the overall shape of the building if we place different components directly next to each other. In this case, both sub-models intersect in a common connection point. Fixing the connection point thus fixes the translation component of the similarity transformation.

Naturally, connection points can be found at the boundaries of each sub-model. In the previous steps, a natural coordinate frame was chosen such that the vertical direction corresponds to the z-direction. We make the assumption that splits occur along the façade direction (*c.f.* Sec. 2). Making a Manhattan-world assumption, façades can be approximated by z-x and z-y planes. Therefore, connection points can be defined as the boundaries and intersections of these planes. Finding candidate connection points boils down to estimating z-y and z-x planes for each sub-model. This is done in two main steps: main plane estimation and plane division. Plane estimation is done on dense data to improve accuracy.

Algorithm 1 Plane division

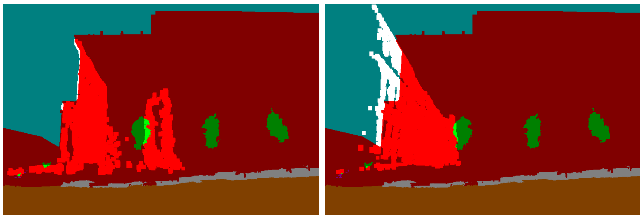
```
Input: Set of $x$-$y$ lines $L$
Output: Set of detected segments $S$
for all $l \in L = \{l_1, \dots, l_N\}$ do
    Divide $l$ into $K$ intervals $I$ of equal length $t$
    for all $I_k : k \in \{1, \dots, K\}$ do
        Drop $I_k$ if $|I_k| < |l|/K$
        Divide surviving $I_k$ along $z$ direction into $J$ intervals $I_{kj}$ of equal height $t$.
        $\forall I_{kj}$ : Drop $I_{kj}$ if $|I_{kj}| < |I_k|/J$
        The surviving adjacent $I_{kj}$ give the height of $I_k$
    end for
    Create segments $s$ from surviving adjacent $I_k$ with the same height.
    Add all $s$ to $S$
end for
```

Main plane estimation Given a set of points projected onto the ground plane, we detect lines that correspond to planes. Since we look for z-x and z-y planes, we only need to search for 2D lines with normals parallel to the x- and y-axis, respectively. If we first look for x-aligned lines and then for y-aligned ones, a single point fully defines a line hypothesis. A point $\mathbf{x} \in \mathbb{R}^2$ is an inlier to a line defined by another point $\mathbf{x}_0 \in \mathbb{R}^2$ and a normal direction $\mathbf{n} \in \mathbb{R}^2$ if the closest distance between $\mathbf{x}$ and the line is below some threshold δ, *i.e.*, if

$$|\mathbf{n}^T\mathbf{x} - \mathbf{n}^T\mathbf{x}_0| \leq \delta \ . \tag{1}$$

Since the normal is aligned with the axis, this is reduced to comparing only one of the coordinates of the points. We use a 1D Hough transform-inspired approach as follows: in order to be an inlier, the value $\mathbf{n}^T\mathbf{x}$ has to be in the interval $\Delta(\mathbf{x}_0, \mathbf{n}, \delta) = [\mathbf{n}^T\mathbf{x}_0 - \delta, \mathbf{n}^T\mathbf{x}_0 + \delta]$, which is equivalent to the case that the intervals $\Delta(\mathbf{x}_0, \mathbf{n}, \delta/2)$ and $\Delta(\mathbf{x}, \mathbf{n}, \delta/2)$ overlap. For each point $\mathbf{x}$, we pre-compute the interval boundaries for $\Delta(\mathbf{x}, \mathbf{n}, \delta/2)$. We then sort the boundaries in ascending order under the constraint that the starting value of an interval appears before the endpoint of another interval if they have the same value. Finding the line with the largest number of inliers then reduces to traversing the sorted list. This procedure is applied sequentially to find all lines.

Plane division Each detected line possibly consists of multiple connected segments, representing separate but aligned façades. This step aims to find each separate façade plane. Two aligned planes will be separated if they are not connected to each other or if they have different heights. The boundaries of each segment/plane, as well as the intersection of perpendicular planes with similar heights, correspond to possible connection points. The plane division procedure is described in alg. 1. The algorithm goes through each line, divides it into intervals of length t (with t being a



(a) Good alignment. (b) Bad alignment.

Figure 4. Evaluation of pairwise transformations using semantic labels. White pixels correspond to free space violations.

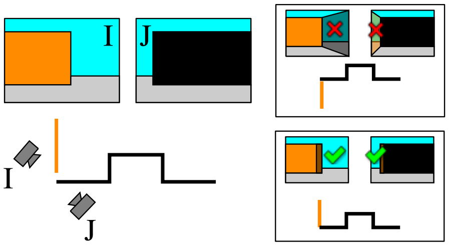


Figure 5. Example of consistent vs. inconsistent pairwise connections.

threshold computed automatically as 10% of the estimated point cloud height), and then rejects those intervals without enough points (the notation $|I|$ in alg. 1 denotes the point count of an interval). In a second stage (inner loop in alg. 1), each of these intervals is subdivided along the vertical direction and those subintervals without enough points are dropped. This way, only adjacent segments with roughly the same height (with a difference up to t) are assembled into a plane.

Finally, we obtain candidates for connection points by taking the endpoints of all planes as well as the intersections of multiple planes. An example of the resulting set of points is shown in Fig. 3.

3.3. Generating and filtering pairwise connections

Given a pair of sub-models and their connection points, and the assumptions of a Manhattan-world scene and an outside-looking-in arrangement, we generate all possible geometrically consistent connections in the following way:

- The fact that the z direction points upwards and the y direction points towards the cameras determines that the negative x direction is left and the positive x is right. We can eliminate hypotheses that stitch connection points on the same side since it would not be consistent with an outside-looking-in arrangement. Additionally, we only pair up connection points at similar heights. This is illustrated in figure 5.
- We connect models right to left which allows to choose only between a 0° and a 90° rotation angle.

In this manner, the total number of connections between two sub-models m_i and m_j, with $i \neq j$, is $2 \cdot n_i \cdot n_j$, where n_i is the number of connection points for m_i (and n_j, m_j respectively). Note that we attach sub-models on one side, therefore the set of connecting hypotheses between m_i and m_j is not the same as between m_j and m_i. In order to minimize the number of possible alignments, we want to select a few connections for each of the two rotation values. In order to do so, we evaluate the quality of each alignment: we extract semantic labels for each image in the two reconstructions using the classifier from [10], labeling each pixel as either ¨building¨, ¨vegetation¨, ¨ground¨, or ¨sky¨. Next, we select for each model the images containing pixels labeled as ¨sky¨ and use the rigid transformation defined by a connection point pair and a rotation angle to project the dense point cloud from one model into the images from the other model and vice versa. Points that are projected to pixels labeled ¨sky¨ constitute a conflicting observation as a good connection should preserve the silhouettes of the two sides of the building. We thus evaluate an alignment by counting the number of points projected onto the sky for each selected image. For each of the two rotation values, we then select the connection point pair with the lowest number of conflicting projections.

Fig. 4 shows an example for this procedure. Our inconsistency measure consists on counting the number of white pixels in these images. As can be seen, correct alignments produce a much lower inconsistency measure than wrong ones. Notice how our definition of the quality of an alignment naturally extends the definition of conflicting observations from [6]. While [6] uses superpixels to detect conflicting geometry, we use semantic information to detect misalignments of the underlying geometry since we do not have visual overlap between the images.

Obviously, not all splits occur at a corner, *i.e.*, there are sub-models for which the building outlines cannot be seen in their images. In this case, most alignments will have a low inconsistency measure. Consequently, we retain all possible connections. This case is detected when the difference between the lowest inconsistency measure and the second lowest inconsistency measure is not big enough (less than 10 times).

In order to be robust against imprecision in the alignment and noise in the dense point clouds, we first perform a dilation by 10 pixels (for this verification step we use subsampled images at a resolution of around 800x500) on the images containing the semantic labels and an opening using a 3x3 kernel on the projected points before counting conflicting observations.

3.4. Model Merging

After selecting the possible connections for each pair of sub-models, we proceed by connecting the models to a single reconstruction of the building. We fix the first model and iteratively add other models to the right. After the last

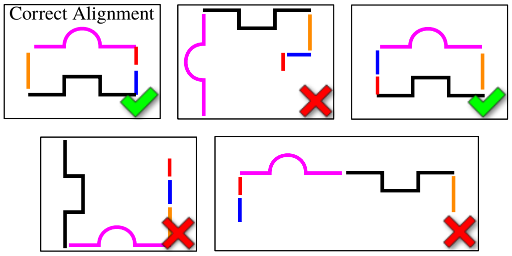


Figure 6. Loop-closure constraint for evaluating a reconstruction.

model is added, we measure the shortest distance between the left connection points of the first model and the right connection points of the last added model. Since we assume that we have only small gaps and should be able to reconstruct the building nearly completely, we prefer alignments where this distance is as small as possible. Consequently, we sort the generated models in ascending order of the Euclidean distance between their endpoints. This loop-closure criterion is illustrated in figure 6. Similar to [18], we currently generate all possible combinations exhaustively. We do this by setting all sub-models as the initial one.

Naturally, the model that best closes the building outlines is not necessarily the best approximation to the true geometry of the scene. However, we can again exploit symmetries to check whether we can identify amongst the top candidates the most likely ones based on consistent symmetry planes. If a sub-model has a symmetry plane, *i.e.*, if the corresponding part of the building contains mirror symmetries, detecting another symmetry plane for the opposite side of the building offers a strong cue that the two planes should actually be aligned. We use this cue to re-rank the top-10 reconstructions sorted based on the distance between their two free corners. For each such model, we search for pairs of parallel symmetry planes on opposite sides of the building and measure their distance in 3D. The final score for the reconstruction is then computed as the average distance between the closest opposite pairs of symmetry planes per x and y direction.

In the next section, we show that this second criterion helps us obtain better reconstructions. Notice that we do not perform a final optimization that tries to enforce an alignment of the symmetry planes. Such an optimization would need to respect free-space constraints, which are hard to optimize. We purposely do not want to enforce this as a hard constraint in order to allow for plausible solutions of a model that may not fully close. We also show that all top-ranked solutions are plausible in general, thanks to the geometrically consistent reasoning that inspired our method.

Algorithm 2 gives a summary of the pipeline described in the previous steps.

Algorithm 2 Complete pipeline

```
Input: $M = \{m_0, \ldots, m_N\}$ set of non-overlapping sub-models of a repetitive/symmetric building
for all $m \in M$ do
    Align $m$ with $x$-$y$-$z$ axes as in [4].
end for
for all $m_i, m_j \in M, i \neq j$ do
    Estimate scale and $z$-translation from SIFT matches if possible.
end for
for all $m_i, m_j, m_k \in M, i \neq j, j \neq k, i \neq k$ do
    Check consistency of scales and translations and propagate them for unmatched pairs.
end for
for all $m \in M$ do
    Estimate planes and compute connecting points.
end for
for all $m_i, m_j \in M, i \neq j$ do
    Measure conflicts between all consistent pairwise hypotheses.
    Filter out implausible hypotheses.
end for
Exhaustively generate the set $R$ of all possible reconstructions.
$\forall r \in R$ : Rank by the distance of their closest endpoints.
Take the best 10 reconstructions and re-rank them by the distance between their parallel symmetry planes.
Output the best reconstruction.
```

4. Experiments

We evaluate our approach on three challenging datasets showcasing different properties. The University model disconnects into 6 sub-models, reconstructed from a total of 338 photos, while the Museum and Capitol datasets contain 3 disconnected sub-models each. The Museum dataset has been reconstructed from 84 images while 469 photos were used for Capitol. This last dataset was taken from [5] where only the main façade was used. Fig. 9 illustrates, for each of the three datasets, the different components as well as the photos at the boundary of each sub-model. As can be seen, occluding objects such as trees and large differences in viewpoint make feature matching impossible. To further demonstrate that reconstructing each building is hard, in Fig. 8 we compare to the models obtained using VisualSfM [24]. For the University as well as the Museum datasets, the models get disconnected due to the lack of good matches on different corners. Furthermore, for University, one of the sides gets collapsed on to the other and is attached to the front of the main façade. For Capitol, symmetric and repetitive structures found on different façades cause the model to collapse on itself. In addition, we use a fourth dataset, Southbuilding taken from [5], for which a full reconstruction is available. By removing images from all four corners of the building, we obtain separate submodels that we then stitch back together using our method in order to provide quantitative results.

4.1. Experimental setup

Out method requires the use of a threshold t in order to estimate planes as mentioned in section 3.2. This threshold is computed automatically per dataset by taking 10% of the average building height, which proves to be accurate and restrictive enough. The classifiers used for the semantic labels are the ones presented in [10] and were trained on the eTrims dataset [9]. The importance of using the semantic labels to filter pairwise connections is key for models with many components, as is the case for the University dataset, where the number of candidate full reconstructions was reduced from 79.6M to 46.7K.

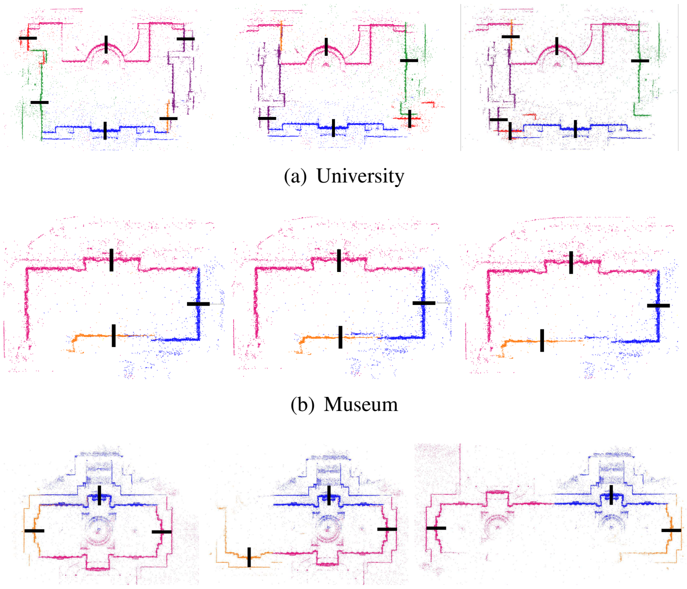
4.2. Results

Fig. 7 depicts the top-3 models generated by our method that have been obtained by re-ranking the 10 top-ranked models whose outlines are as closed as possible. At first glance, all three models look geometrically plausible for each dataset. A closer look at the second and third model generated for the University dataset reveals the importance of considering the alignment of symmetry planes. In both models, the left and right sides of the building have been exchanged, causing the left side of the building to move inwards and intersect the pink model. Our scoring function based on aligning symmetry axes enforces the correct placement of the red sub-model to the upper left corner while correctly placing the pink model at the right side. At the same time, the front and back sides of the building are properly aligned along the symmetry plane.

The alignment of symmetry planes becomes crucial when dealing with incomplete buildings, as is the case for the Museum dataset. Due to a lack of matches and good view-angle, a part of the back façade is missing. Using only the loop-closing criterion, the reconstruction shown in the third column of Fig. 7 is chosen. However, the correct solution is ranked in first place when checking for symmetry plane alignment (first column).

Notice that for the Capitol model, the first scoring function that minimizes the distance between connection points is already sufficient to generate the correct model.

Fig. 10 compares the top-ranked model generated for each dataset with an aerial view from Google Maps. Even though we can output a set of plausible reconstructions, the top-ranked one consistently matches the best to the real building outline for all three of our datasets.


(a) University

(b) Museum

(c) Capitol

Figure 7. The three top-ranked models generated by our method for each dataset. Black lines denote symmetry planes detected by our approach.

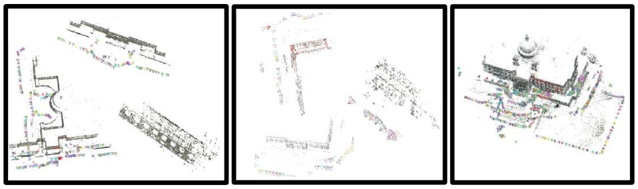


Figure 8. Reconstructions achieved by visualSfM. Left: University. Middle: Museum. Right: Capitol.

Fig. 11 shows the original reconstruction for Southbuilding as well as the results obtained after removing images from all corners. This example shows both the robustness of our method as well as a failure case. One of the sides of the buildings is mostly occluded by trees, therefore we cannot find a natural frame for the corresponding submodel nor an appropriate scale and z-translation. This submodel is thus ignored and not used in the stitching. However, our method still works very well even though there is a big gap, showing the robustness in presence of incomplete loops. We aligned the stitched model with the original reconstruction using ICP and found that the average position error between the original cameras and our result is about 50cm, with only 23cm corresponding to height differences. The average error for 3D points is about 40cm.

5. Conclusion

In this paper, we have tackled the challenging problem of creating a single 3D model from multiple SfM reconstructions of different parts of a building that do not share enough visual overlap. We have presented a method that is able to generate models that closely resemble the true structure of the scene without requiring any GPS measurements or other

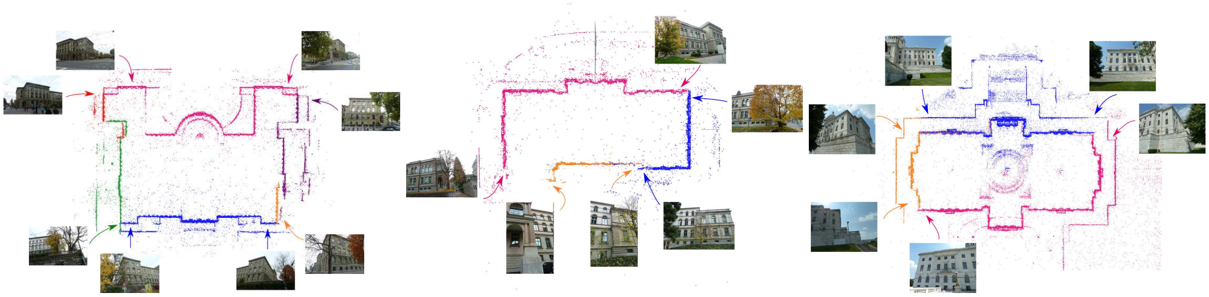

(a) University (b) Museum (c) Capitol

Figure 9. Aligned sub-models for each dataset. We show the photos corresponding to the connection points of each sub-model. The change of view-angle and occlusions make these pictures impossible to match.

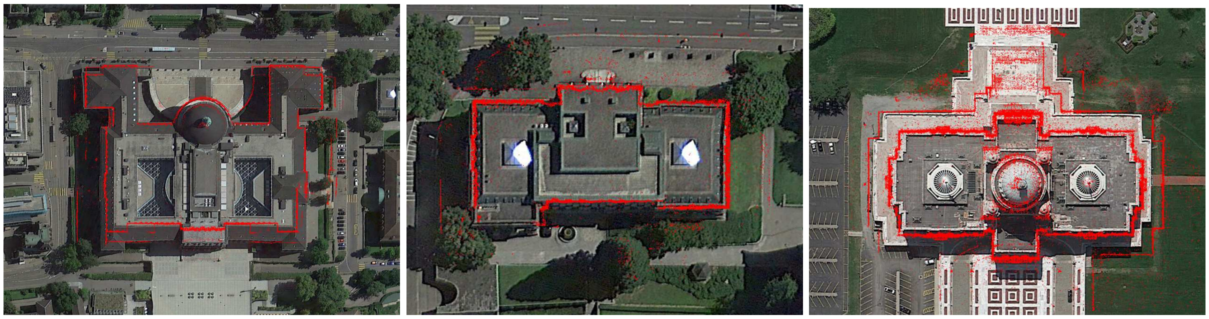


Figure 10. Overlaying the merged University, Museum, and Capitol models onto aerial views obtained from Google Maps. Small errors are present when there are gaps in the reconstruction. All reconstructions look plausible and similar to the actual scene.

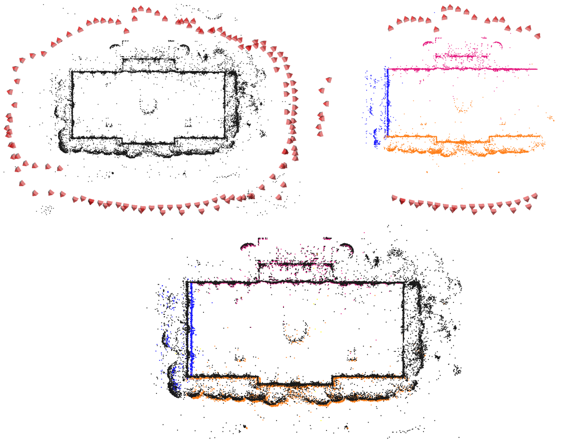


Figure 11. Southbuilding dataset: original complete reconstruction, stitched model, and overlap.

geographic information such as building outlines or geo-registered 3D models. Our method exploits symmetries to simplify the alignment problem and uses semantic reasoning, both on an image level and on the level of the geometry, to rate possible alignments. We have shown experimentally that our method succeeds in reconstructing buildings under challenging conditions.

In order to be able to solve the problem of aligning 3D models with no overlap, we had to make some assumptions about the nature of the problem, *e.g.*, that we have reconstructions covering all sides of the building and that there are symmetric structures that repeat across sub-models. While our assumptions hold for the largest part of urban scenes, in future work we would like to relax these assumptions. This implies finding additional sources of information, *e.g.*, from time stamps and illumination changes, that can be used to disambiguate the alignment problem. We would also like to explore the use of semantic labels to determine relative scale and vertical alignment instead of relying on symmetric structures.

Besides model merging, the proposed approach can also be used to simplify the image acquisition process. Instead of carefully taking pictures all around a building, it is sufficient to take a few images of each façade and combining the resulting components using our method. This can significantly accelerate the manual acquisition process, especially in more complicated scenes.

To our knowledge, we are the first to tackle the challenging problem of merging visually disconnected models. In order to inspire further research, we make all datasets and results available at `http://www.cvg.ethz.ch/research/model-merging/`.

Acknowledgements This project was funded by the CTI Switzerland grant #17136.1 Geometric and Semantic Structuring of 3D point clouds.

References

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

[2] R. Cabral and Y. Furukawa. Piecewise Planar and Compact Floorplan Reconstruction from Images. In *CVPR*, 2014.

[3] D. Ceylan, N. J. Mitra, Y. Zheng, and M. Pauly. Coupled Structure-from-Motion and 3D Symmetry Detection for Urban Facades. *ACM Trans. Graphics*, 2013.

[4] A. Cohen, C. Zach, S. Sinha, and M. Pollefeys. Discovering and Exploiting 3D Symmetries in Structure from Motion. In *CVPR*, 2012.

[5] C. Haene, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d scene reconstruction and class segmentation. In *CVPR*, 2013.

[6] J. Heinly, E. Dunn, and J.-M. Frahm. Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction. In *ECCV*, 2014.

[7] N. Jiang, P. Tan, and L.-F. Cheong. Seeing Double Without Confusion: Structure-from-Motion in Highly Ambiguous Scenes. In *CVPR*, 2012.

[8] R. Kaminsky, N. Snavely, S. Seitz, and R. Szeliski. Alignment of 3D Point Clouds to Overhead Images. In *CVPR Workshops*, 2009.

[9] F. Korč and W. Förstner. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn, April 2009.

[10] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative Hierarchical CRFs for Object Class Image Segmentation. In *ECCV*, 2009.

[11] R. Martin-Brualla, Y. He, B. C. Russell, and S. M. Seitz. The 3D Jigsaw Puzzle: Mapping Large Indoor Spaces. In *ECCV*, 2014.

[12] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. In *CVPR*, 2011.

[13] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. Occluding Contours for Multi-View Stereo. In *CVPR*, 2014.

[14] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. SIFT-Realistic Rendering. In *3DV*, 2013.

[15] C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and Scalable Large Scale Image Reconstruction. In *CVPR*, 2010.

[16] K. Tuite, N. Tabing, D.-Y. Hsiao, N. Snavely, and Z. Popović. PhotoCity: Training Experts at Large-scale Image Acquisition Through a Competitive Game. In *CHI*, 2011.

[17] O. Untzelmann, T. Sattler, S. Middelberg, and L. Kobbelt. A Scalable Collaborative Online System for City Reconstruction. In *ICCV Workshops*, 2013.

[18] G. Wan, N. Snavely, D. Cohen-Or, Q. Zheng, B. Chen, and S. Li. Sorting unorganized photo sets for urban reconstruction. *Graphical Models*, 74(1):14–28, 12012.

[19] C.-P. Wang, K. Wilson, and N. Snavely. Accurate Georegistration of Point Clouds using Geographic Data. In *3DV*, 2013.

[20] A. Wendel, C. Hoppe, H. Bischof, and F. Leberl. Automatic Fusion of Partial Reconstructions. *ISPRS*, I-3:81–86, 2012.

[21] A. Wendel, A. Irschara, and H. Bischof. Automatic Alignment of 3D Reconstructions using a Digital Surface Model. In *CVPR*, 2011.

[22] T. Weyand and B. Leibe. Discovering Favorite Views of Popular Places with Iconoid Shift. In *ICCV*, 2011.

[23] K. Wilson and N. Snavely. Network Principles for SfM: Disambiguating Repeated Structures with Local Context. In *ICCV*, 2013.

[24] C. Wu. Towards Linear-Time Incremental Structure from Motion. In *3DV*, 2013.

[25] C. Zach, A. Irschara, and H. Bischof. What Can Missing Correspondences tell Us About 3D Structure and Motion? In *CVPR*, 2008.

[26] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating Visual Relations using Loop Constraints. In *CVPR*, 2010.

[27] C. Zach and M. Pollefeys. Practical Methods For Convex Multi-View Reconstructions. In *ECCV*, 2010.

[28] B. Zeisl, K. Köser, and M. Pollefeys. Automatic Registration of RGB-D Scans via Salient Directions. In *ICCV*, 2013.