Dynamic and Scalable Large Scale Image Reconstruction

Christoph Strecha
CVLab EPFL
Christoph.Strecha@epfl.ch

Timo Pylvänäinen
Nokia Research Center
Timo.Pylvanainen@nokia.com

Pascal Fua
CVLab EPFL
Pascal.Fua@epfl.ch

Abstract

Recent approaches to reconstructing city-sized areas from large image collections usually process them all at once and only produce disconnected descriptions of image subsets, which typically correspond to major landmarks.

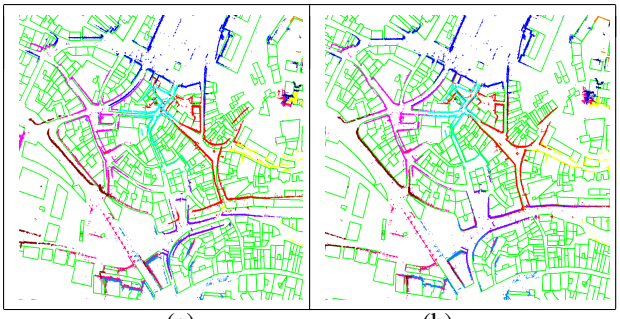
In contrast, we propose a framework that lets us take advantage of the available meta-data to build a single, consistent description from these potentially disconnected descriptions. Furthermore, this description can be incrementally updated and enriched as new images become available. We demonstrate the power of our approach by building large-scale reconstructions using images of Lausanne and Prague.

1. Introduction

As digital cameras and camera-equipped mobile devices become ever more prevalent, users naturally produce ever larger image databases [6]. Furthermore, since these devices tend to possess additional sensors such as GPS, inclinometers and compasses, additional meta-data is often created and stored as well.

State-of-the-art methods [18, 2] can now handle these large image collections to build any city in a single day, to paraphrase a recent paper [2]. However, because the spatial locations of the images often form compact clusters [5] that correspond to popular touristic sites, what is typically reconstructed is not the whole city but individual, unconnected 3D models corresponding to landmarks such as the Coliseum, the Trevi Fountain, or St. Peter's Basilica [18]. This follows from the fact that most camera calibration pipelines [20, 13, 10, 17, 2] tend to break typical image datasets, which include few if any pictures of the locations between landmarks, into separate clusters. Not only can images of *geographically* distant locations end up separated, but even those of the same 3D scene can end up in different clusters if they have been acquired under so different lighting conditions that point correspondences are hard to establish.

In this paper, we take advantage of the available meta-data, such as GPS, geo-tags, and models that are freely available in Geographic Information Systems (GIS), to overcome this fragmentation problem. Our framework is designed to withstand the fact that such data can be wildly inaccurate—GPS is not precise in urban environments, people make mistakes when tagging their photos and the information in a GIS database is outdated or imprecise—and could degrade the reconstructions instead of improving them, if such inaccuracies where not taken into account. To this end, we introduce an incremental approach in which image clusters are preserved but linked to each other in a flexible manner. Their relative positions can be updated as additional images and information become available.



(a) (b)

Figure 1. Aligning calibration clusters. (a) Rough alignment based solely on geo-tagged images. (b) Refined alignment. The points belonging to different clusters are shown in different colors overlaid on green building outlines obtained from a map. The points match the outlines far more accurately after refinement. This figure and most others in this paper are best viewed in color.

In other words, we introduce a framework that lets us produce increasingly refined representations as the size and richness of the image databases increases, without having to redo all the computations from scratch. We also propose an innovative integration scheme that can be invoked at any time to produce an integrated city model that brings together images, geo-tags, and GIS data into a consistent description, given the current state of the database.

This approach to integration has two key advantages. First, it is fast because it only requires estimating rigid trans-

formations to bring calibrated image clusters into the same coordinate system, as shown in Fig. 1. Second, the merger can be carried out even if the visual evidence for merging is weak or even absent.

This is in contrast to existing approaches that compute a single representation, or several unconnected representations, once and for all by using only image correspondences. As a result, the final bundle adjustment can take several hours and does not scale to models at city scale. Furthermore, there is no obvious way to refine it incrementally without redoing the whole computation when new images become available.

In the remainder of the paper, we first present the issues that must be addressed for effective reconstruction of large scale city models and discuss related work. We then introduce our approach to representing large image collections and the associated meta-data and to deriving an integrated model from this representation. Finally, we present our results using images of Prague and Lausanne.

2. Issues in Large Scale Reconstruction

Reconstructing a large scale area such as a city is often thought of as a batch process: One first acquires many images and then processes them all at once to produce a 3D reconstruction. This is something that 3D reconstruction pipelines [20, 13, 17, 18, 2] have become very good at by exploiting point correspondences between millions of images. However, they make no provision for updating the reconstruction as new images become available. In effect, one has to gather all images before creating a reconstruction instead of being able to incrementally refine an initial model using whatever new data becomes available. This raises two major issues that we discuss below.

2.1. Fragmentation of the Representation

Pipelines that rely solely on image correspondences often fail to produce a unified 3D reconstruction of the whole area. Instead, they tend to group the images into several disjunct clusters that correspond to different geographic locations or sometimes even to different parts of the same location. This usually happens because there are many more images of places of interest, which are taken from similar viewpoints and therefore easy to match, than images of locations between these places. Because the latter images are far sparser and taken from potentially very different viewpoints, they are much harder to match and only few, if any, correspondences can be found. If images of the same scene are acquired under very different conditions, by night or by day, in winter or in summer, appearances can be different enough so that matches cannot be established either, which is also likely to result in different clusters for the same location.

This problem can be overcome by acquiring the images very methodically and systematically as is done by car-mounted vision systems mounted on cars, such as those described in [1, 3] among many others. Such industrial-strength systems often couple differential GPS and sophisticated Inertial Measurement Units (IMU) to the cameras. This is effective and yields reliable and accurate camera positions over large areas but is hardly applicable to the images and associated meta-data produced by standard mobile devices such as those tourists usually carry. To exploit it effectively, one has to explicitly account for the fact that it is potentially not only inaccurate but also completely wrong. To the best of our knowledge, this is not a problem that has been addressed in the context of large scale reconstruction. The same applies to using GIS databases to improve reconstructions since they are rarely up-to-date. The only recent approach we are aware of aligns a single image-based reconstruction to an aerial image or a structural model [11] but makes no attempt at updating the one using the other.

2.2. Scalability

The computational requirements of bundle adjustment represent a major bottleneck in the production of large scale city models. This is especially true when one attempts to use all image correspondences to optimize globally the structure and all the camera parameters. Because this does not scale in terms of memory and speed, it has been proposed to use a skeleton graph model [18], from which redundant images have been removed. Nevertheless, for large, spatially extended scenes, redundancy is low and bundle adjustment quickly becomes infeasible.

Furthermore, a single large scale representation of all images also increases the risk of introducing drift, which is difficult both to detect and to correct. Once introduced, drift can prevent loop closing, even when new images that could possibly provide loop closing information become available. It is therefore advantageous to work with small, possibly overlapping representations, for which bundle adjustment is still feasible and within which drift is unlikely. The final reconstruction can then be efficiently obtained by keeping the representations fixed and optimizing only over their relative positions and orientations, once sufficient additional information has been obtained.

3. Cluster growing

The input to our system is a collection of calibrated sets of images, or *calibration clusters* that could be obtained by using anyone of the existing calibration pipelines [20, 13, 17, 16, 2]. We use our own, which initializes the internals from EXIF-data [16]. Like the others, it produces clusters that can be either large or consisting of as few as three images. Initially, these clusters do not overlap and are uncon-

nected.

At no point do we attempts to merge these clusters. Instead, as new images become available, whenever possible, we grow the clusters by adding the new images to them. When enough new images from a new and not yet discovered place are found, the reconstruction pipeline is restarted to build a new cluster.

To gather the images shown in this paper, we implemented a simple search for geo-tagged flickr images and location-tagged google images. These were continuously added to our database. Under those conditions, well calibrated clusters have a tendency to grow fast. By contrast, small ill-calibrated clusters have a much lower chance of doing so. In this way, which is easy to parallelize, clusters compete with each other so that high quality ones eventually become dominant.

To add a new image to the database we use a bag-of-words model. Each 3D point corresponds to a feature track in the images, where it is observed. The feature vectors of all tracks build the dictionary, which we use to assign each new image to the most similar clusters and then, within the clusters, to the most similar ones, as in [15].

The bag-of-words model is used to compute the term frequency - inverse document frequency (tf-idf) vector for all clusters and for all images. The tf-idf vector of an image is then matched to the clusters and - for the best clusters - to the images within those by using the normalized scalar product [15]. A detailed feature matching is then performed only on the best matching images. Depending on whether the internals are known, we use a robust version of the PnP algorithm [12, 4] which finds the external camera parameters, or reject the image when not enough correspondences can be established.

4. Aligning calibrated image clusters

In this section we describe the process of aligning calibration clusters w.r.t. each other and moving them into a common coordinate system. Sources of information include GPS, geo-tags, digital elevation models (DEM), 2D building models, up-vector estimates, as well as image correspondences between clusters. They all provide different constraints, which we discuss in detail later.

Formally, we wish to estimate, for each of the K calibration cluster $\mathcal{C}_k, k = 1 \dots K$, a similarity transformation $\mathcal{T}_k$ which minimizes alignment error. To perform this task, we are given measurements which, depending on the type of constraint, have different, unknown accuracy and which could even be wrong. We model this problem by formulating a generative model of inliers and outliers, similar in spirit to Fransens *et.al.*, *e.g.* [8, 7, 19].

We will now explain this model for one type of constraint for which we are given measurements y_i. These could be the difference of the GPS information with the position of camera as estimated by the cluster calibration (see 4.1.4). Later on, we combine all constraints, each of which follows the same inlier/outlier model. The inlier measurements are assumed to be generated by a normal distribution $\mathcal{N}(y_i; 0, \Sigma)$ with zero mean and unknown covariance Σ. All outliers are assumed to be generated from a uniform distribution g. The hidden variable x_i assigns each measurement y_i to the inlier $x_i = 1$ or outlier model $x_i = 0$ We denote by $\boldsymbol{\theta} = \{\mathcal{T}_k \mid k = 1 \dots K, \Sigma\}$ the set of transformation parameters $\mathcal{T}_k$ and the parameter of the inlier model Σ. Each similarity transformation $\mathcal{T}_k$ has seven parameters, which are the components for rotation, a vector in $\mathbb{R}^3$ for translation and a scalar value for scale.

The probability of a given measurement, conditioned on $\boldsymbol{\theta}$ and on the value of the hidden variable x_i, is taken to be:

$$p(y_i|\boldsymbol{\theta}) \quad = \quad \begin{cases} \mathcal{N}(y_i(\boldsymbol{\theta}); 0, \Sigma) & \text{if} \quad x_i = 1 \\ g & \text{if} \quad x_i = 0 \end{cases} \qquad (1)$$

The alignment error is defined as the log likelihood given by this model. Our solution is then the MAP estimate of this generative model.

The strength of this model lies in the robust treatment of outliers and in the appropriate relative contribution of the different cues to the final solution. For instance, the importance of GPS measurements varies quite substantially from place to place. Clusters in narrow streets are expected to be noisier than clusters in open space. Also, it is not known a priori how good the building outline model will be. These variations are automatically taken into account here, which is substantially different from a formulation that uses a fixed robust estimator, *i.e.* one for which the parameters (*e.g.*Σ, g) are not adjusted [8].

4.1. Alignment cues

4.1.1 Up-vector constraint

The "up-vector" constraint refers to the fact that buildings are mostly parallel to the Earth's gravity vector. We estimate the up-vector from each cluster to constrain the orientation of that cluster. Knowing the up-vector is - as we will see later on - also necessary for the building model constraint, which is purely defined in the 2-dimensional map space.

We estimate up-vectors based on the 3D structure of the cluster. More particulary, we assume that most image features are detected on vertical facade structures, which is a reasonable assumption for many urban scenes. First, for each 3D point we estimate its normal. They correspond to the smallest eigenvector of the covariance matrix build from the $n = 64$ nearest 3D point neighbors. We take the up-vector as being the one which is orthogonal to most 3D point normals.

Expectation Maximization (EM) is used to find the up-vector and to estimate the probability of each 3D point normal to be orthogonal to this up-vector. This yields the up-vector and a segmentation of the 3D points into upright and non-upright structures, which we will use to align clusters to a map as will be discussed in sec. 4.1.5).

For EM initialization, we use the smallest eigenvector computed from the covariance of all camera positions. This vector is already upright when all images are taken from the horizontal ground plane. We found that this approach leads to an efficient and precise up-vector estimation in an overwhelming majority of scenes. The direction of the up-vector (up or down) is then computed from images that constain this information in the EXIF-data.

The up-vector is then used to constrain the transformations $\mathcal{T}_k$. Let $\boldsymbol{R}_k$ be the rotation operator of $\mathcal{T}_k$ and $y_k = [\boldsymbol{R}_k \mathbf{u}_k]_3$ be the z-component of the transformed up-vector $\mathbf{u}_k$. Note that y_k is a scalar and $y_k = 1$ if the up-vector is parallel with the Earth's gravity vector. We write:

$$p_u(y_k \mid \boldsymbol{\theta}, x_k = 1) \propto \exp(-(y_k - 1)^T \Sigma_u^{-1} (y_k - 1)) \,. \quad (2)$$

The up-vector constraint only has influence on the rotational components of $\mathcal{T}$. We don't expect any outlier and set $g_u = 10^{-10}$. Each cluster provides one up-vector constraint, *i.e.* $k = 1 \ldots K$.

4.1.2 Camera constraint

During the growing process it is not only possible but also desirable that two clusters overlap and share one or more images. The position of these overlapping images should coincide after alignment. Let k and l be the index of two clusters, and $\boldsymbol{c}_m$ and $\boldsymbol{c}_n$ the 3D position of shared cameras in k and l, respectively. The probability of a measurement $\boldsymbol{y}_i = \mathcal{T}_k(\boldsymbol{c}_m) - \mathcal{T}_l(\boldsymbol{c}_n) \in \mathbb{R}^3$ is proportional to

$$p_c(\boldsymbol{y}_i \mid \boldsymbol{\theta}, x_i = 1) \propto \exp(-\boldsymbol{y}_i^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{y}_i) \,. \quad (3)$$

Since having an outlier camera in any of the given clusters is almost impossible to appear, we set the uniform outlier probability to $p_c(\boldsymbol{y}_i \mid \boldsymbol{\theta}, x_i = 0) = g_c = 10^{-10}$. The number of measurements $\boldsymbol{y}_i$ equals the number of identical images in all pairs of clusters.

4.1.3 Point constraint

As it was the case for the camera constraint, we can find 3D points that are shared between overlapping clusters. The transformation of $\boldsymbol{p}_m$ and $\boldsymbol{p}_n$, being the same 3D points in clusters k and l, to the common coordinate system $\boldsymbol{y}_i = \mathcal{T}_k(\boldsymbol{p}_m) - \mathcal{T}_l(\boldsymbol{p}_n) \in \mathbb{R}^3$, leading to

$$p_p(\boldsymbol{y}_i \mid \boldsymbol{\theta}, x_i = 1) \propto \exp(-\boldsymbol{y}_i^T \boldsymbol{\Sigma}_p^{-1} \boldsymbol{y}_i) \,. \quad (4)$$

The camera and point constraints are based on image correspondences, which are a very accurate source of information. Outliers on 3D points exist but are rare, since we only use 3D points that are visible in at least three cameras. An outlier is a point which accidentally matches correctly in all three images and we use $g_p = 10^{-4}$. The number of measurements $\boldsymbol{y}_i$ equals the number of identical 3D points in all pairs of clusters.

4.1.4 GPS and/or geo-tags

GPS, attached to a mobile image capture device, provides geodetic coordinates, but the calibration clusters are in Cartesian coordinate systems. Therefore, the GPS measurements are first transformed into the Earth Centered, Earth Fixed (ECEF) coordinate system, which is a Cartesian system capable of representing reconstructions at global scale. GPS measures latitude and longitude as well as the altitude of the captured image. Geo-tags are manually supplied indications of the image position on a map and provide only latitude and longitude. We fill this gap using the DEM, which is freely available on [9]. GPS data, DEM and geo-tags are the most unreliable information sources. Outliers are expected to appear, mostly because of users who geo-tag wrongly. GPS has a possibly noisy signal but does not generate real ourliers.

Let $\boldsymbol{p}_i^g$ be the position of a camera i in the ECEF coordinate system as given by GPS or geo-tags+DEM, $\boldsymbol{p}_i^c$ the position of the camera i given by the image based calibration of cluster k and $\boldsymbol{y}_i = \mathcal{T}_k(\boldsymbol{p}_i^c) - \boldsymbol{p}_i^g \in \mathbb{R}^3$ the difference of its transformation into the common coordinate system with the GPS position, than we can write the inlier distribution as:

$$p_g(\boldsymbol{y}_i \,|\, \boldsymbol{\theta}, x_i = 1) \propto \exp\left(-(\boldsymbol{y}_i)^T \boldsymbol{\Sigma}_g^{-1} (\boldsymbol{y}_i)\right) \,. \quad (5)$$

The number of GPS constraints equals the number of images where GPS/geo-tags are available.

4.1.5 Structure alignment with GIS building model

For many cities, the footprint model of all buildings is available, for instance see [14] for a free database. Real 3D building models are less common and usually only include important buildings and tourist attractions. For this reason, we will make the analysis based on 2D building models. We will assume here that the footprint of a building coincides with its facade. To align it with the footprint model, we first need to segment all 3D points in a calibrated cluster that lie on the facade. Our segmentation is based on the 3D point normals of sec.4.1.1. Points with their normal close to perpendicular w.r.t the gravity are assumed to be facade points.

Let $\boldsymbol{p}_i$ be a 3D point which is perpendicular to the up-vector. $\boldsymbol{p}_i$ is transformed by $\mathcal{T}_k$ to the common coordinate

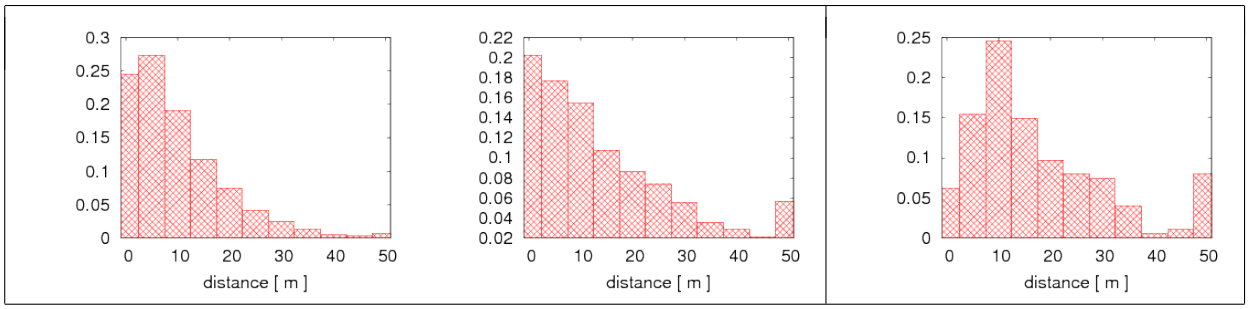


Figure 2. Error distributions extracted from the Lausanne and Prague dataset in meters. The error of mobile GPS devices attached to the camera (left: latitude/longitude, middle: altitude error). The right figure shows the error distribution for images from Flickr, where users have marked the image position manually. The last bin ($50m$) collects all larger errors.

system and, to relate it with the building model, further projected by $\mathbf{P}$ in the 2D map coordinate system. To measure alignment, we create a distance image $\mathbf{D}$ in the 2D map space which is zero on all facades and increases from there in all directions. The quality of a particular transformation is then measured by the value of this distance function

$$\begin{aligned} y_i &= \mathbf{D}\left(\mathcal{P}\mathcal{T}_k(\boldsymbol{p}_i)\right) , \\ p_b(y_i \,|\, \boldsymbol{\theta}, x_i{=}1) &\propto \exp\left(-\lambda_m y_i\right) . \end{aligned} \tag{6}$$

Note that, unlike for the other constraints that are normally distributed, we use here an exponentially distributed inlier model, which follows the error distribution, given by the values of the distance function $\mathbf{D} \in [0 \ldots]$.

For the building model we can assume high accuracy but many outliers, since the building map often contains only big buildings. Small upright structures which are present in our image based reconstructions, but not in the map, have to be assigned to the outlier model so as to not disturb the optimal solution. Another source of errors involves outdated building models. To account for this, we use in all experiments $g_b{=}0.2$ for the outlier model.

We use bi-linear interpolation on the distance image to calculate its value and derivative for $\mathcal{P}\mathcal{T}(\boldsymbol{p}_i)$. These building models constrain the orientation component of $\mathcal{T}$ and two components of the translation, but it provides no information on the absolute height of the aligned clusters. GPS, also provides only very noisy measurements on the height. To constrain this remaining degree of freedom we use a DEM model, which is defined next.

4.1.6 Altitude constraint

Consider the subset of all 3D points where normals are orthogonal to the up-vector. An even further subset is the collection of points that have minimal height w.r.t the up-direction. These points $\boldsymbol{p}_i$ are likely to represent the ground level, and their z-component should, after applying the transformation $\mathcal{T}_k(\boldsymbol{p}_i)$, be similar to the altitude $\mathbf{A}(\mathcal{P}\mathcal{T}_i(\boldsymbol{p}_i))$ provided by [9]. Thus:

$$\begin{aligned} y_i &= \mathbf{A}\left(\mathcal{P}\mathcal{T}_k(\boldsymbol{p}_i)\right) - [\mathcal{T}(\boldsymbol{p}_i)]_3 \\ p_a(y_i \,|\, \boldsymbol{\theta}, x_i{=}1) &\propto \exp\left(-y_i^T \Sigma^{-1} y_i\right) . \end{aligned} \tag{7}$$

Many outliers are expected here and we use $g_a{=}0.2$.

4.2. Optimization

In the previous section we described all the cues that we can use. We combine them by a non-linear optimization of the global objective function. First, the availability of the different cues is checked for each cluster and image. We select only those clusters for which at least 30 images are equipped with GPS or geo-tags. Than we use these to find the initial transformation $\mathcal{T}_i$ by a RANSAC procedure.

We alternate further between optimizing *(i)* over the parameters of all rigid transformations $\mathcal{T}_k$ and *(ii)* over the parameters of all inlier distributions Σ and the expected values of the hidden variables $E[x_i]$, which weight each individual constraint according to its inlier probability. The rigid transformations are optimized by Levenberg-Marquardt, for which the Jacobians can be computed analytically for all the log-likelihood terms in eq. 1. The inlier distributions and the expected values of x_i can be computed in closed form from the same log-likelihood [19, 7]. The process converges fast and after five iterations the parameters stabilize for all practical purposes.

5. Experiments

We present results obtained using 10735 images of Lausanne, which is a true three-dimensional city whose center is built on steep hills, and 17043 of Prague, which has a well photographed city center with many interesting sights, in addition to major landmarks.

In the remainder of this section, we first show that, once the full model is built, camera locations are known with much better accuracy than what the GPS and geo-tags initially provided. We then demonstrate that we can obtain

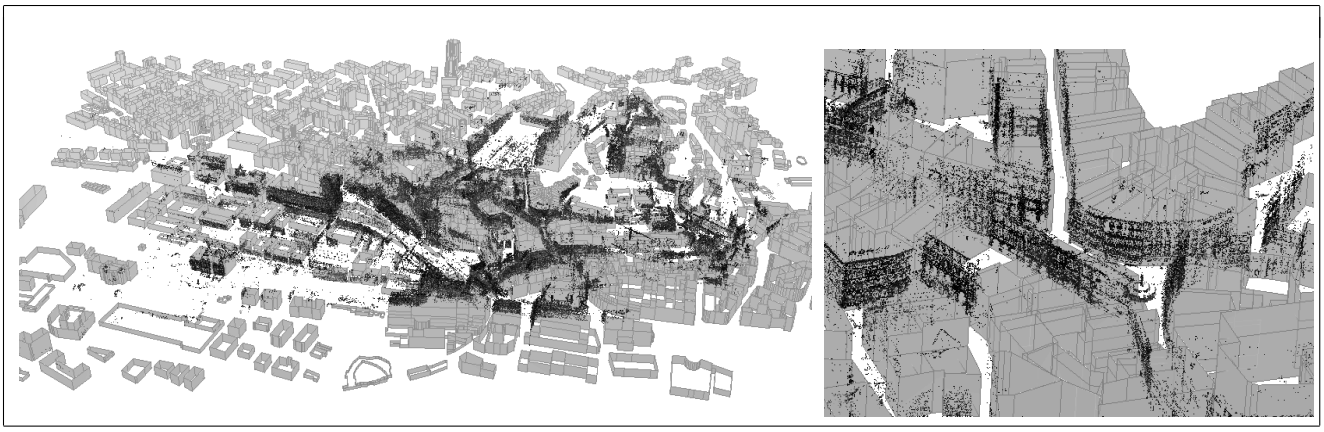


Figure 3. Three dimensional visualization of the Lausanne reconstruction of fig. 1 b.

more consistent representations than direct bundle adjustment over the whole dataset. Finally, we give integration examples of clusters which are far apart and which cannot be connected by image correspondences.

5.1. GPS and Geo-Tagging Accuracy

After alignment, the calibration clusters can be superposed on a preexisting map of the buildings, which has relatively high accuracy, as shown in Fig. 1. Fig. 3 shows the corresponding 3D rendering of this current reconstruction state for Lausanne.

Particularly amidst high buildings in narrow streets, GPS is notoriously inaccurate and it is fair to assume that the camera locations we obtain after alignment are of far greater accuracy than the GPS measurements.

It then becomes possible to actually estimate the accuracy of both the GPS data and the user-supplied annotations. By combining the results we obtained on the Lausanne and Prague data, we obtain the error distributions depicted by Fig. 2.

5.2. Algorithmic Behavior

It is well known that bundle adjustment of long "strings" of images suffers from accumulation of error. Not only that, but information propagation is limited by the availability of matches between images and so convergence in long sequences with only matches between neighboring images is often slow.

Therefore, by not attempting to bundle adjust all the images for which correspondences exist, but instead running our cluster alignment procedure, we can achieve better results in much less time. Fig. 4 illustrates the differences for four clusters from Prague which partially overlap. Our approach also makes it straightforward to use any number of additional constraints.

Furthermore, our method makes it possible to align clusters, even when they share absolutely no image matches. This is illustrated in Fig. 5. The cathedral tower has been photographed from different sides, but there are no image matches connecting the sides. GPS alignment alone does not result in visually pleasing result, but the proposed method successfully reconstructs the entire tower.

6. Conclusions

We have proposed an approach to handling image data sets that scales to city-sized models and can be dynamically updated as new images become available, without having to redo the whole computation from scratch. Furthermore, it results in higher accuracy when meta-data is available.

In practice, the ability to exploit meta-data is important because preexisting city maps can easily be found for most cities and it is by now a rare image database that does not include geo-tagged images.

The alignment of consistent smaller clusters, for which bundle adjustment is still feasible, has advantages over a global bundle adjustment for many practical settings and if additional information is taken into account. Such a formulation scales to more images, by orders of magnitude, and can solve reconstruction problems for which a global bundle adjustment does not fit to speed and memory requirements. In addition, problems of drift and weakly connected images are less pronounced.

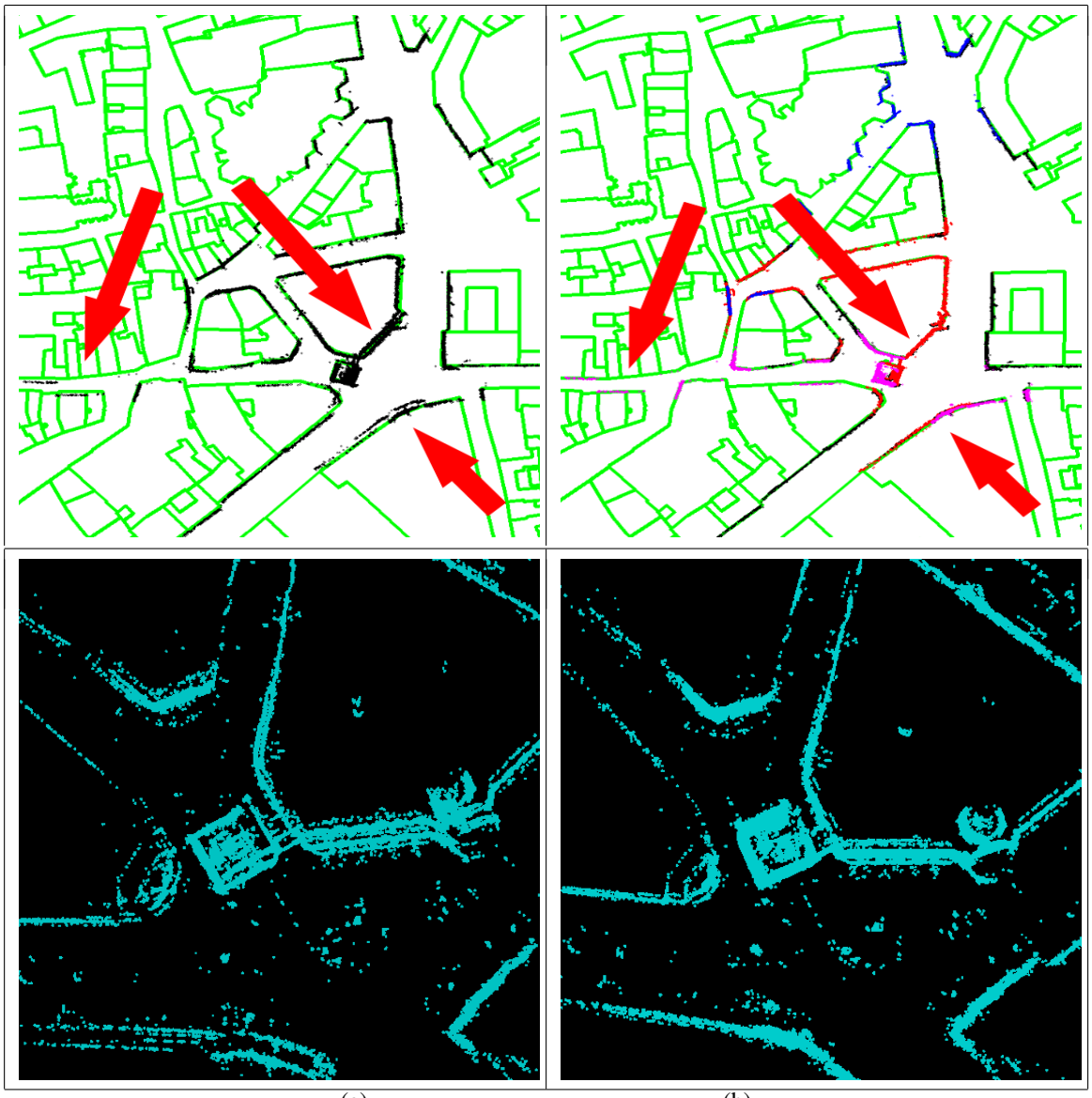
In future work, we will develop more semantic descriptions of the 3D models and detect changes with a view to developing automated methods for updating these models.

Acknowledgments

This work was supported in part by Nokia Research Center.

References

[1] Google street view. *http://maps.google.com/*.



(a) (b)

Figure 4. Global bundle adjustment vs. cluster alignment. (a) Alignment result obtained by merging all four clusters and bundle-adjusting them based on the images matches only. The reconstructed 3D points are then aligned with the preexisting city map (by using our optimization on this single cluster) and overlaid in black. (b) Alignment result obtained by not merging the clusters beforehand and applying our approach with all constraints. The reconstructed 3D points belonging to each cluster are shown in a different color. Note that they match the building outlines far more accurately than before. Two zoomed areas from the corresponding top images are shown in 3D underneath. Note the points spreading across some facades for the global bundle adjustment (a) but not in our reconstruction (b).

[2] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in one day. In *ICCV*, 2009.

[3] A. Akbarzadeh, J. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nistér, and M. Pollefeys. Towards urban 3D reconstruction from video. In *3DPVT*, pages 1–8, 2006.

[4] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. In *CVPR*, pages 1–8, 2008.

[5] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *WWW*, pages 761–770, 2009.

[6] Flickr. Community photo collection. *http://flickr.com/*.

[7] R. Fransens, C. Strecha, and L. V. Gool. A mean field em-algorithm for coherent occlusion handling in map-estimation prob. In *Conference on Computer Vision and Pattern Recognition*, 2006.

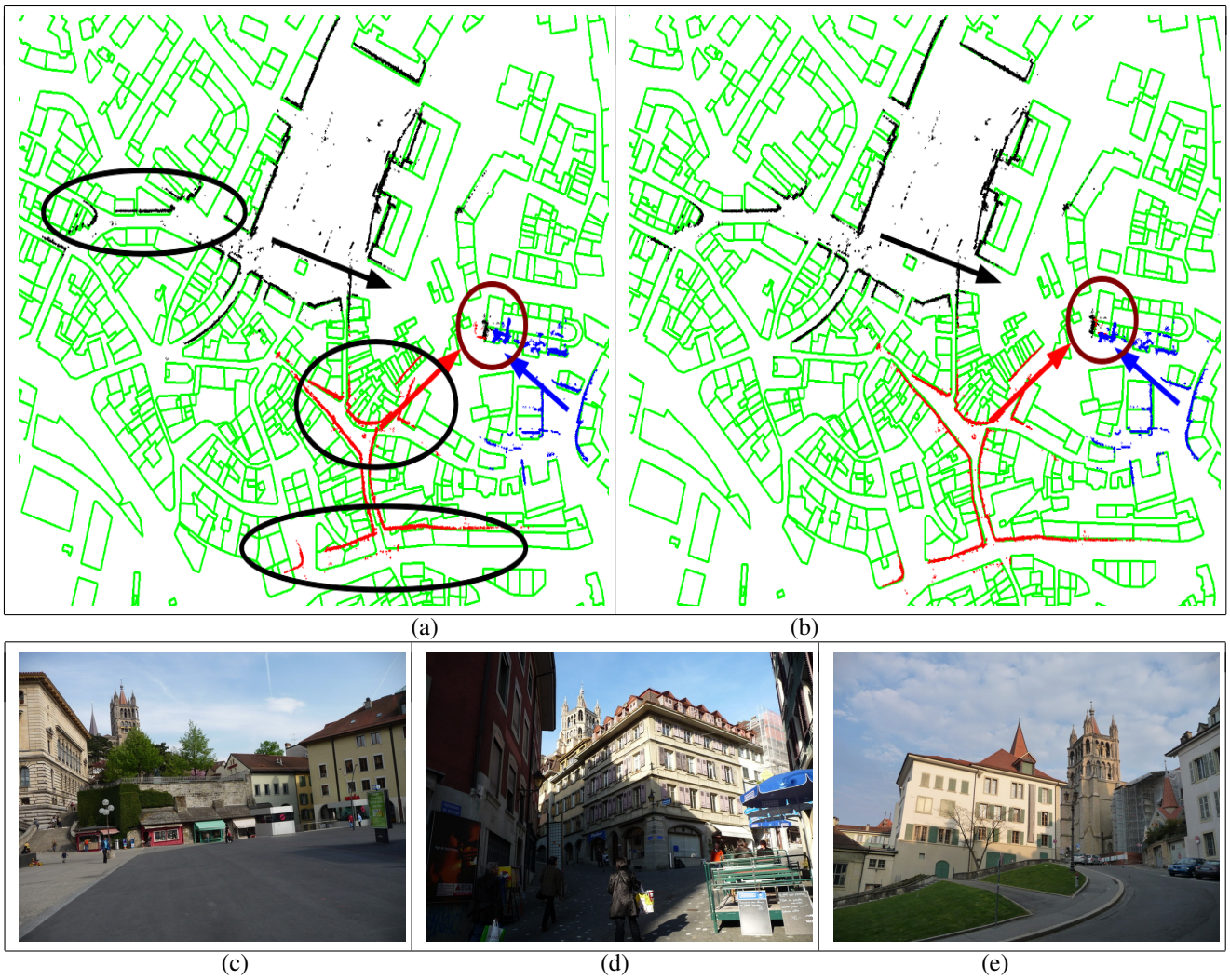


Figure 5. Bundle alignment without matches. The initialization (a) based on GPS/geo-tags and the final result (b). Observe how the points within the red circle that correspond to the cathedral tower become more consistent after the proposed alignment. Three images are shown, one for each cluster. The colored arrows indicate the approximate position and orientation: the black arrow corresponds to image (c), red (d) and blue (e). Note also the improved building alignment, most visible within the black ellipses.

[8] R. Fransens, C. Strecha, and L. Van Gool. Robust estimation in the presence of spatially coherent outliers. In *RANSAC workshop at CVPR*, 2006.

[9] Geographical database. *http://www.geonames.org/*.

[10] A. Irschara, C. Zach, and H. Bischof. Towards wiki-based dense city modeling. In *VRML07*, pages 1–8, 2007.

[11] R. Kaminsky, N. Snavely, S. Seitz, and R. Szeliski. Alignment of 3d point clouds to overhead images. In *InterNet09*, pages 63–70, 2009.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2), February 2009.

[13] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, 2007.

[14] The free wiki world map. *http://www.openstreetmap.org/*.

[15] J. Sivic and A. Zisserman. Efficient Visual Search for Objects in Video. *Special Issue of the Proceedings of the IEEE*, 96(4):548–566, 2008.

[16] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *ACM SIGGRAPH*, pages 835–846, Boston, MA, 2006.

[17] N. Snavely, S. Seitz, and R. Szeliski. Photosynth, 2008. *http://livelabs.com/photosynth/*.

[18] N. Snavely, S. Seitz, and R. Szeliski. Skeletal sets for efficient structure from motion. In *CVPR*, 2008.

[19] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *CVPR*, 2004.

[20] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Mach. Vision Appl.*, 17(6):411–426, 2006.