Regular Paper

Challenges in wide-area structure-from-motion

Marc Pollefeys (Institute of Visual Computing, ETH Zurich)
(Dept. of Computer Science, University of North Carolina at Chapel Hill)

Jan-Michael Frahm
(Dept. of Computer Science, University of North Carolina at Chapel Hill)

Friedrich Fraundorfer
(Institute of Visual Computing, ETH Zurich)

Christopher Zach (Institute of Visual Computing, ETH Zurich)

Changchang Wu
(Dept. of Computer Science, University of North Carolina at Chapel Hill)

Brian Clipp
(Dept. of Computer Science, University of North Carolina at Chapel Hill)

David Gallup
(Dept. of Computer Science, University of North Carolina at Chapel Hill)

Abstract

The topic of this paper is wide area structure from motion. We first describe recent progress in obtaining large-scale 3D visual models from images. Our approach consists of a multi-stage processing pipeline, which can process a recorded video stream in real-time on standard PC hardware by leveraging the computational power of the graphics processor. The output of this pipeline is a detailed textured 3D model of the recorded area. The approach is demonstrated on video data recorded in Chapel Hill containing more than a million frames. While for these results GPS and inertial sensor data was used, we further explore the possibility to extract the necessary information for consistent 3D mapping over larger areas from images only. In particular, we discuss our recent work focusing on estimating the absolute scale of motion from images as well as finding intersections where the camera path crosses itself to effectively close loops in the mapping process. For this purpose we introduce viewpoint-invariant patches (VIP) as a new 3D feature that we extract from 3D models locally computed from the video sequence. These 3D features have important advantages with respect to traditional 2D SIFT features such as much stronger viewpoint-invariance, a relative pose hypothesis from a single match and a hierarchical matching scheme naturally robust to repetitive structures. In addition, we also briefly discuss some additional work related to absolute scale estimation and multi-camera calibration.

1. Introduction

In recent years there has been a growing interest in obtaining realistic visual representations of urban environments. This has mainly been driven by the need to provide a visual and spatial context for information on the internet. While currently most existing commercial products are limited to aerial views, such as Google Earth and Virtual Earth/Bing Maps, or only provide 2D visualization, such as Google Street View, the most effective and flexible representation would be a photo-realistic ground-level 3D model. Besides virtual exploration, this would also support many more applications such as for example autonomous navigation, visual localization and mobile augmented reality.

There are two main types of approaches being explored for capturing realistic 3D models of large-scale urban environments. One type uses LIDAR to capture the 3D geometry and images to capture the appearance, while the other type of approach uses solely images to capture both 3D geometry and appearance simultaneously. An early example of a LIDAR-based approach is the work by Früh and Zakhor[16]. The earliest examples for image-based 3D modeling of urban scenes probably dates back about a hundred years to the origin of photogrammetry. However, only in the last decade or two has automation made it feasible to approach large scale ground-based modeling of urban scenes from images. The early approaches for automated 3D modeling from images such as the ones proposed by Tomasi and Kanade[57] and Pollefeys et al.[40] were limited to modeling more or less what could be seen in a single image. Our more recent approaches could model larger scenes that could not fully be seen from a single view-point[41], but was much too slow to use for larger scale reconstructions as processing was in the order of a minute per frame. In this paper, we will focus on our most recent approach[42], which leverages the computational power of the graphics processing unit (GPU) to recover 3D models from urban imagery at video-rate on a standard PC. The GPU is particularly well-suited to achieve high performance for many image processing tasks such as tracking or matching features[52),63),69),71] and stereo matching[65),67]. Other approaches to perform 3D reconstruction of urban scenes from images have recently been proposed,

but they mostly generate simplified models without a lot of detail, e.g.[9),32),64)], or require human interaction[53)]. Another interesting approach for obtaining visual 3D models leverages the emergence of community photo-collections[1),30),54)]. Similarly spatio-temporal city models can be obtained from archives collected over time[49)]. These approaches, however, are mostly limited to landmark structures for which many photographs are available.

The simplest approach to obtain consistent maps over large scales is to use a Global Positioning System (GPS), but this can be problematic for some applications such as mapping of indoor spaces, dense urban neighborhoods or other areas where the GPS signals are weak or unavailable (GPS signals can be easily jammed). While structure-from-motion allows to obtain consistent local 3D maps, over long distances errors in position, orientation and scale accumulate. Therefore, an important challenge in large-scale reconstruction and mapping consists of obtaining self-consistent maps. One of the most important steps to achieve this is to close loops when the camera revisits the same location. Many approaches have been proposed based on SIFT[31)] and other invariant features. Specific approaches have been proposed to efficiently match novel images to large number of previously acquired images[10),15),24),38)]. However, these approaches all rely on the ability to generate enough potential correspondences in the first place. This can be a significant problem in scenarios where the viewing angle can be very different when a place is revisited. Our approach introduced in Wu et al.[61)] proposes to use the local geometric reconstruction to extract visual features on the 3D surface instead of in the images (i.e. we extract features in ortho-rectified patches). This provides viewpoint invariance and allows for direct estimation of a 3D similarity transformation from a single match, which enables robust matching even with very few correct correspondences.

The remainder of the paper is organized as follows. In Section 2 we introduce our video-rate urban 3D modeling pipeline. In Section 3 we present our approach to loop-closing under widely varying viewpoints. Section 4 discusses additional issues related to calibration of multi-camera systems and solutions to the problem of absolute scale estimation from video. A discussion of open issues and the conclusion are given in Section 5.

2. Real-time Urban 3D modeling from images

This section describes the different steps of our 3D modeling pipeline. The input to our system consists of a video stream combined with a GPS and an Inertial Navigation System (INS), although we are currently exploring how to perform drift free large scale reconstruction without these additional sensors (see Section 3 for more details). The output is a detailed dense textured 3D surface reconstruction of the recorded urban scene. An example is shown in **Fig. 1**.

The example was recorded in Victorville, CA, on the site of the DARPA Urban Challenge. As our goal in this case was to reconstruct 3D models of the facades (as opposed to robot navigation for example), our cameras were oriented to the side of the vehicle. For this example the camera recorded 170,000 video frames at 30Hz with a resolution of 1024×768. This means that at 50 km/h an image is recorded approximately every 50 cm along the path of the vehicle. The small baseline simplifies feature tracking for motion estimation. It also ensures very high overlap between the images. The high redundancy facilitates the use of simple multi-view stereo algorithms that can be implemented very efficiently on the GPU. In **Fig. 2** an overview of the different stages of our video processing pipeline is given. While it is important to perform processing at rates comparable to capture time, it is more effective for this processing to happen off-line than on-board the vehicle. For efficiency a separate input

Fig. 1 Urban 3D modeling from video: top view of 3D model of Chapel Hill, NC, and close-up view of the model in two areas

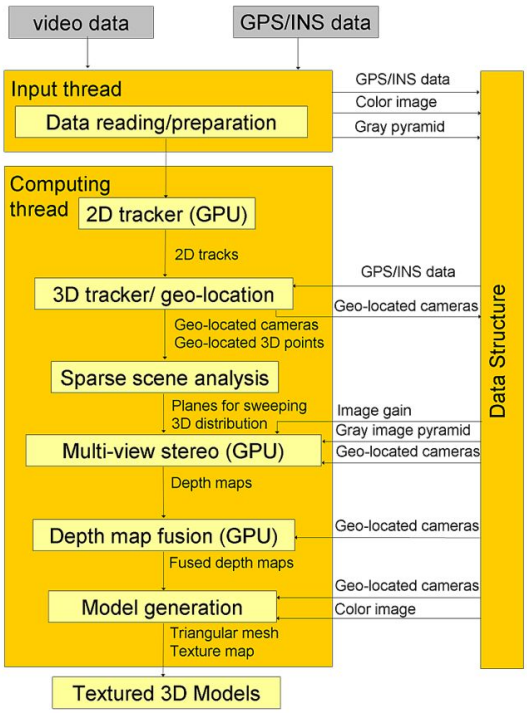

Fig. 2 Processing modules and data flow of our 3D reconstruction pipeline.

thread deals with reading the video data from disk. The first computing module extracts features that are then tracked in subsequent frames. To achieve a high performance, this step is performed on the GPU. Next, the feature tracks are used in complement with the INS/GPS system to determine the precise motion of the vehicle and to localize the cameras in world coordinates. At the same time, the 3D location of the tracked features is recovered. This is then used by the next module to obtain a range for the depth of the scene, as well as to extract the dominant orientations of the facades. This information is used to set up the dense multi-view stereo module. This step is followed by a robust depth-map fusion processing, which computes consensus depth-maps by making use of visibility constraints. Both of these steps are efficiently performed on the GPU. Finally, the fused depth-maps are triangulated to obtain a 3D surface mesh. Double representations are removed and a subset of the original images are used as textures for the 3D model. The overall processing rate of our system is 30Hz on a single PC for one video stream as described above. The largest fraction of time is spend in the stereo and fusion steps (about 30% each) which

are performed at half-resolution (512×384), while the other steps are performed on full-resolution images. The next sections are discussing the main processing steps in more detail.

2.1 2D feature tracking and motion estimation

The first step to determine the motion between consecutive video frames is to extract salient image features and track them from frame to frame. For this purpose we use a variant of the well-known Kanade-Lukas-Tomasi (KLT) tracker[50]. To deal with a mix of sunlit and shadow regions, it is important to vary the exposure during recording. In[27] we showed that a consistent global exposure change for the image can efficiently be recovered jointly with the feature displacement and that this performs better than brightness invariant approaches. Our current pipeline uses a very fast KLT implementation, which tracks 1000 features with more than 200Hz in 1024×768 images[69]. This implementation is available in open-source[71].

The next step consists of determining the motion of the camera. As feature tracks can drift and become erroneous, it is important to use robust techniques for motion estimation. For this purpose we use the Random Sampling Consensus (RANSAC)[11]. As this is an essential algorithm for many computer vision systems, many improvements have been proposed. We have for example proposed an approach to deal with quasi-degenerate cases such as scenes where most points lie on a single plane[12] (as these violate some assumptions of the basic algorithm and tend to confuse RANSAC into stopping too early). As hypotheses generated by RANSAC are dependent on a minimal sample, they can often strongly be affected by noise and not allow to directly identify all the inliers. We have recently also proposed a RANSAC algorithm, which properly takes the measurement uncertainties and the transformation uncertainties into account to early identify additional potential inliers and gain up to an order of magnitude in efficiency[44]. In the current system, we use a RANSAC approach, which combines the benefits of several previous approaches to minimize the amount of processing[43]. Knowing the internal calibration of the cameras used we deploy minimal solvers to estimate the relative motion from five points[36] and perform pose estimation from three points[22] as hypothesis generators for

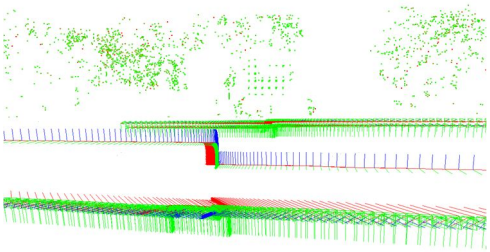

Fig. 3 Illustration of benefit of combining video and GPS/INS for motion estimation. The central track of coordinate frames, which exhibits 10cm vertical drift while the vehicle stopped at a red traffic light, corresponds to the GPS/INS only motion estimation. The coordinate frames in the front and back represent the results from combined GPS/INS and video tracking.

RANSAC. The initial triangulation of feature points uses the optimal approach proposed in[23]. Our approach is similar to the visual odometry approach described in[37]. However, this approach only provides satisfying results over relatively short distances. To obtain better results from video only, it is important to perform visual loop-closure as will be discussed in Section 3.

For large scale reconstructions our current system uses a Kalman filter on the 2D feature tracks and the GPS/INS data to estimate geo-located camera poses and 3D feature locations. In **Fig. 3** the benefit of fusing the GPS/INS data with the 2D feature tracks in a Kalman filter is illustrated. Even high-end GPS/INS systems suffer from drift which are in particular disturbing when the vehicle stops or moves slowly. In these cases vision allows to remove most of the error.

In addition, as a by-product of the motion estimation, we also obtain the 3D location of the tracked salient scene features. This is very useful as it provides us with a range of interest for the dense stereo matching. In addition, we extract the dominant orthogonal facade orientations from the salient 3D feature points to facilitate the generation of plane hypotheses aligned with building facades as this improves the result of the stereo algorithm[17]. The vertical facade direction is obtained from the INS system or by detecting the corresponding vanishing points. The feature points are projected on a horizontal plane. For each possible orientation in the plane the histograms of x and y coordinates are computed and

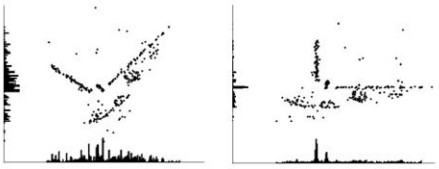

Fig. 4 Top-view of 3D feature locations for two different orientations together with histograms of x and y coordinates. The minimal histogram entropy configuration (shown on the right) is selected.

the orientation for which these histograms have the lowest entropy is selected. This process is illustrated in **Fig. 4**.

2.2 Fast multi-view stereo matching

To obtain a detailed reconstruction of the surface geometry the sparse set of points reconstructed previously is insufficient. For each video frame and its temporal neighbors we perform multi-view stereo matching to compute the depth for every pixel. Our approach is based on the GPU-friendly multi-view stereo approach proposed in[65),67)]. For a reference video frame its neighbors are identified and a collection of scene planes is hypothesized which samples the depth range sufficiently densely to avoid disparity aliasing. For a particular plane each neighboring image is now projected first on the plane and from there into the reference image where its photo-consistency with respect to the reference image is evaluated. This is done in a single image warping operation during which the exposure change is also compensated. For each pixel the sum of absolute differences is computed. In practice, this is done separately for the reference image and the five previous images and for the reference image and the five succeeding images to provide robustness to occlusions. The minimum of the previous images costs and the succeeding images cost is kept. For each pixel costs are aggregated over a correlation window and the plane with the lowest cost is selected for each pixel independently. From this plane labels it is simple to obtain the depth for each pixel. The same process is repeated for the next frame. A rolling buffer is used to store the eleven frames currently used on the GPU so that each time only one new frame needs to be transferred. More details on this algorithm are provided in[26),42)]. For urban scenes with dominant facade

orientations, better results are obtained by using plane hypotheses aligned with the facades. This approach is described in detail in[17]. In this case three sets of planes are hypothesized, two for orthogonal facade directions and one parallel with the ground-plane (not necessarily horizontal in our implementation). For each pixel one can now obtain a depth and a normal. This approach is illustrated in **Fig. 5**. It can be seen from the figure that the orientations obtained by this approach are largely correct. The depth results are also better as typically the lowest cost is now achieved for the whole aggregation window consistently for the correct depth and orientation. To help resolve the ambiguity in homogenous regions, we add a prior, which is derived from the distribution of the sparse feature points along the different axes as illustrated in Fig. 4. If needed, this stereo algorithm can also be accelerated significantly by only considering the planes with the highest prior likelihoods. Very recently, we have also explored the possibility to explicitly detect extended planes across multiple images and use these to improve the quality of the model[19].

One important issue with stereo is that the accuracy degrades quadratically with depth. In many application though the goal is to recover a reconstruction of the scene up to a pre-determined accuracy. In these cases fixed-baseline stereo often has trouble reaching the required depth resolution in the far range of the working volume, this often implies a prohibitive amount of computations are performed in the near range. In[18] we proposed an approach to vary both baseline and resolution used throughout the working volume. The discretization of space for both the standard algorithm and our algorithm are shown in **Fig. 6**. The amount of computations and accuracy are proportional to the density and shape of the volume elements respectively.

Once a depth map has been computed for each video frame, it is important

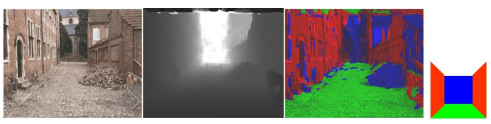

Fig. 5 Multi-view stereo with multiple surface orientation hypotheses: original video frame (left), depth map (middle), orientation map (right).

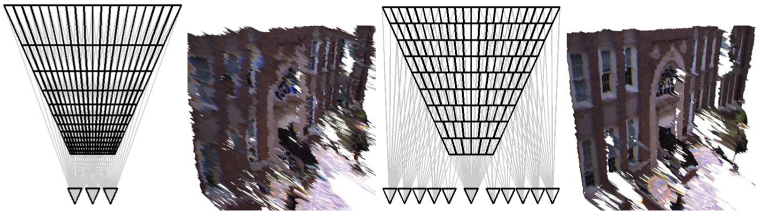

Fig. 6 Discretization of space and results for standard stereo (left) and variable baseline/resolution stereo (right).

to reconcile overlapping depth maps. This is performed with a robust visibility-based depth map fusion approach. Different types of visibility-based conflicts are shown in **Fig. 7**. On the left current hypothesis A conflicts with the point A' obtained from view i and we say the free-space of A' is violated. On the right current hypothesis C would be occluded in the reference view by C' obtained from view i. Our approach selects for each pixel the closest point along the viewing ray which has at least as many occlusions as free space violations. The approach can very efficiently be implemented on the GPU. More details on this approach can be found in[34]. Another very interesting depth-map fusion approach, which minimizes the $TV-L^1$ was recently proposed by Zach et al.[68]. While stereo depth-maps were computed for every frame and had a lot of overlap (every point on the surface is seen in 10-30 frames), fused depth-maps are only computed for a subset of these frames. The goal is to maintain a factor of 2-3 overlap so that regions of the scene that are occluded by foreground objects in one view can be filled in from neighboring fused depth-maps.

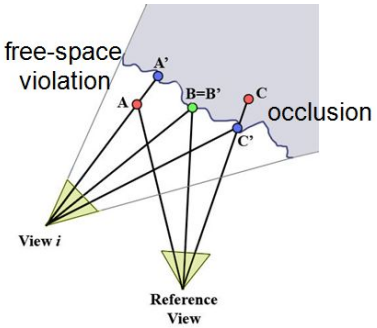

Fig. 7 Visibility-based constraints for depth-map fusion.

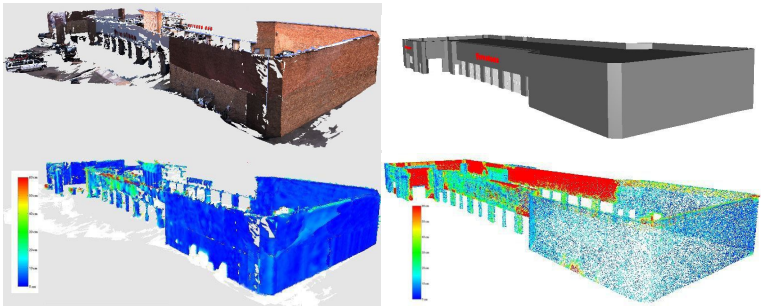

Fig. 8 Firestone evaluation: reconstructed 3D model (top-left), ground-truth 3D model (top-right), color-coded accuracy evaluation (bottom-left) and color-coded completeness evaluation (bottom-right). Blue indicates errors below 10cm while red indicates errors larger than 50cm.

2.3 3D urban modeling

Starting from the fused depth maps, a 3D mesh is obtained by overlaying a triangular mesh on the image and projecting it into space according to the depth. An efficient multi-resolution quad-tree algorithm is used to minimize the number of triangles in the mesh[39]. As consecutive fused depth-maps still have a significant amount of overlap, we remove double representations in a post-processing step by projecting previously reconstructed surfaces in the current view and verifying depth consistency. Only previously not reconstructed surfaces are then reconstructed. This strategy allows to fill in holes in the reconstruction initially left behind columns for example.

To evaluate the quality of our results, our 3D reconstruction results were compared to a ground-truth model obtained through a professional survey. The results of this comparison are shown in **Fig. 8**. The accuracy was evaluated by determining the closest point on the ground-truth model for each of the vertices in our model. Completeness is determined similarly by determining if for every vertex of a regular sampled ground-truth model there is a point on our model within some pre-determined distance ($30cm$ in our case). The median and mean error for our approach on this data set are $2-3cm$ and $5-6cm$ respectively, depending on the settings, and the completeness varies from $66\%-73\%$. The relatively low completeness is mostly due to unobserved surfaces and saturated white regions for which no surface was reconstructed.

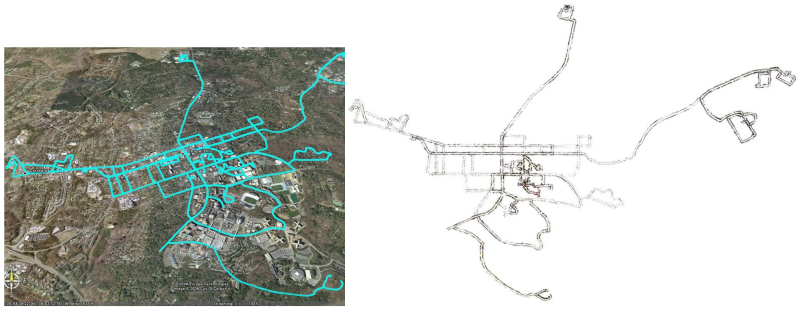

Fig. 9 Map of Chapel Hill, NC, with vehicle path overlaid (left) and top view of recovered reconstruction (right).

Fig. 10 Top view of segment of the Chapel Hill reconstruction with two views of facades.

The complete 3D urban modeling pipeline can process incoming 1024×768 video streams collected at 30 frames per second in real-time on a single PC (with stereo and depth-map fusion being run at half-resolution). This was for example done for a 1.3 million frame data set captured in Chapel Hill. In **Fig. 9** the path of the vehicle is shown on the map and a top view of the complete reconstruction is shown. An alternative simplified fusion and reconstruction approach has very recently been proposed in[20]. In **Fig. 10** a top view of a one reconstruction segment is shown, as well as facade views of two buildings from that segment.

3. Visual loop-closing

3D models that are created sequentially from input images are always subject to drift. Each small inaccuracy in motion estimation will propagate forward and the absolute positions and motions will be inaccurate. It is therefore necessary to do a global optimization step afterwards to remove the drift. This makes

constraints necessary that are capable to remove drift. Such constraints can come from global pose measurements like GPS (as currently used in the system as described in the previous section), but in case these are not available (e.g. indoors, GPS denied areas or urban canyons) internal consistency constraints like loops and intersections of the camera path can be used, e.g.[45]. In this section we will therefore discuss solutions to the challenging task of detecting loops and intersections and using them for global optimization.

3.1 Loop detection using visual words

To correct the encountered disturbance through the drift in the absence of GPS, we need to detect when the camera intersects its previous path. Registering the camera with respect to the previously estimated path provides an estimate of the accumulated drift error. For robustness the path self-intersection itself can only rely on the views itself and not on the estimated camera motion, which drifts unbounded. This visual loop detection and can also be phrased as a location recognition problem [46].

Our location recognition system determines the path intersection by using salient image features and evaluating their similarity to the previously observed salient features in all views. The system deploys the SIFT- [31] or if local geometry is available our view invariant VIP-features[61] (see next section). The local geometry is typically provided through the estimation processes described in Section 2.2 . For the fast computation of SIFT features, we make use of SIFTGPU[63], which can for example extract SIFT features at $\sim 10Hz$ from 1024×768 images.

To find corresponding previous views, we would need to test the current view for overlap to all previous views. Typically the overview would be determined through a matching of the salient features in the two views. Given that it is computationally prohibitive to compute the similarity of salient features in the current view to all features in all previously observed views we use the vocabulary tree [38] to find a small set of potentially corresponding views. The vocabulary tree provides a computationally efficient indexing for the set of previous views. It quantizes a high-dimensional feature vector (SIFT or VIP) by means of hierarchical k-means clustering. An alternative consist of using[60].

The quantization assigns a single integer value, called a visual word (VW), to the originally high-dimensional feature vector. This results in a very compact image representation, where each location is represented by a list of visual words, each only of integer size. The list of visual words from one location forms a document vector, which is a v-dimensional vector where v is the number of possible visual words (a typical choice would be $v = 10^6$). The document vector is a weighted histogram of visual words normalized to 1 (more precisely, the term frequencyinverse document frequency is used). To compute the similarity matrix the $L2$ distance between all document vectors is calculated. The document vectors are naturally very sparse and the organization of the database as an inverted file structure makes this very efficient.

Determining the visual words for the features extracted from the query image requires traversal of the vocabulary tree for each extracted feature in the current view including a number of comparisons for the query feature with the node descriptors. Hereby the features from the query image are handled independently, hence the tree traversal can be performed in parallel for each feature. To optimize performance we employ an in-house CUDA-based approach executed on the GPU for faster determination of the respective visual words. The speed-up induced by the GPU is about 20 on a GeForce GTX280 versus an CPU implementation executed on an Intel Pentium D 3.2Ghz. This allows to perform more descriptor comparisons than in [38], i.e. a deeper tree with a smaller branching factor can be replaced by a shallower tree with a significantly higher number of branches. As pointed out in [48], a broader tree yields to a more uniform, hence representative sampling of the high-dimensional descriptor space. An alternative for optimization consists of using *kd*-trees which are cheaper to compute at the cost of a more restricted partitioning of space (i.e. axis-aligned).

Since the vocabulary tree only delivers a list of previous views potentially overlapping with the current view we perform a geometric verification of the delivered result. First we exhaustively compare the features of the search frame with the features in the candidate frame to find highly correlating pairs of features. For efficiency reasons we implemented this as a dense matrix multiplication on the GPU. Afterwards the potential correspondences are tested for

a valid two-view relationship between the views through our efficient RANSAC approach[43]. In case scenes contain repeated structures, it might be necessary to use a more advanced verification scheme which verifies global consistency[70].

To increase the performance of the above described location search we use our recently proposed index compression method [24]. This allows us to perform the search with approximately 10 Hz and typically success rates of more than 70%. The approach uses the initial redundancy of the feature representation in the standard vocabulary approach that stores a representation for each feature in each image. Instead we use the information from the structure from motion (Section 2.1), which links the salient features belonging to the same 3D scene point. We perform a mean-shift clustering [8] on all observed SIFT or VIP descriptors of the 3D scene point and store only the cluster centers in the vocabulary tree. This effectively increases the signal to noise ratio of the search. Additionally we introduce summarization views for the scene in stead of indexing all original views, which reduces the size of the index.

Fig. 11 shows the effect of loop closing on a 400m long trajectory of a vehicle equipped with a camera. The path in blue is from the initial camera poses from structure-from-motion. Loop closing is performed using bundle adjustment and the result is the red trajectory in Fig. 11, which shows that the loop is nicely closed. For this experiment bundle adjustment[58] was used to optimize the camera positions and the 3D features. The detected loops were added as constraints to the bundle adjustment optimization. We provide open source code for sparse bundle adjustment[71].

Our approach can also be used to perform efficient large-scale reconstruction from community photo collection[30]. An example of such a reconstruction is shown in **Fig. 12**.

3.2 Viewpoint invariant patches (VIP) for loop closing and 3D model registration

Almost all existing approaches attempt to close loops by matching 2D image features only[10),46]. However, in most applications scenarios for loop closure and localization images are not recorded in isolation. Indeed, a robot navigating through an environment typically collects videos. This imagery is often suffi-

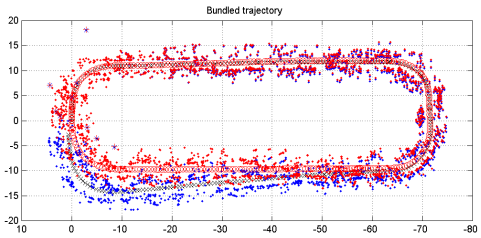

Fig. 11 Camera positions and triangulated features after loop closing (red). Initial estimates are shown in blue and black. The loop is nicely closed.

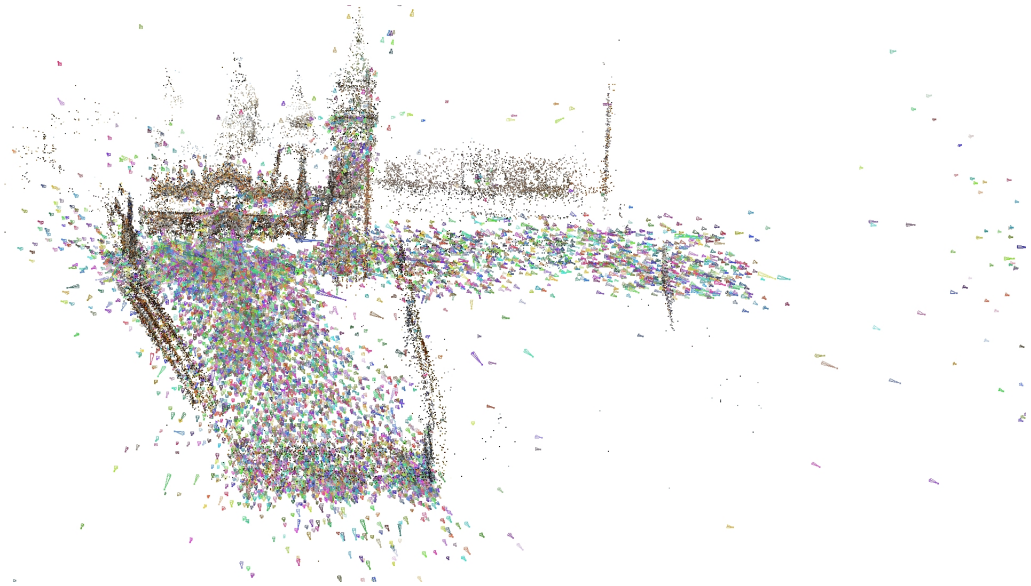

Fig. 12 Reconstruction of part of San Marco square in Venice from a community photo collection with 10338 cameras.

cient to build a local 3D model at each pass using structure from motion and dense stereo matching techniques as explained earlier in this paper. Therefore, we propose to leverage local 3D scene geometry to achieve viewpoint invariance. For this purpose we have introduced Viewpoint Invariant Patches (VIP) in[61]. VIP's are textured 3D patches extracted from the local textured 3D model in a viewpoint invariant way. Conceptually, our goal is to extract features directly on the surface, in stead of in the images. In urban areas with many planar regions, this can be done efficiently by generating an ortho-texture for each planar

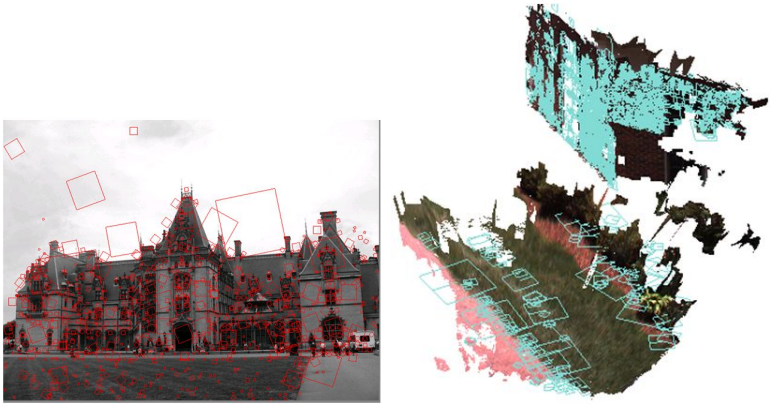

Fig. 13 While SIFT features are extracted from 2D images and are not fully invariant to viewpoint (left), VIP features are extracted on the 3D surface which provide viewpoint invariance and enables single match hypotheses (right).

surface and then extract features from those texture maps. As the ambiguity for a 3D model extracted from images is a 3D similarity transformation (i.e. scale, orientation and location), for features on a 2D plane embedded in the 3D world, 2D similarity invariance is sufficient to deal with viewpoint changes. Therefore, extracting SIFT features (which are designed to be invariant to 2D similarities) from the ortho-textures provides us full viewpoint invariance (up to view-dependent effects due to non-Lambertian surface reflectance and non-planar details). In addition, a single VIP correspondence is sufficient to uniquely determine a full 3D similarity (scale is obtained from the scale of the feature, orientation from the patch normal and the dominant texture gradient in the patch, location from the keypoint at the center of the patch). The VIP concept is illustrate in **Fig. 13**

The viewpoint invariance of the VIP features makes them a perfect choice to be used for 3D model registration or loop closing. In the case of 3D model registration we seek a similarity transformation between two overlapping 3D models. For this, VIP features are extracted from each model and subsequently matched. It should be noticed that the relative scale between all matching features extracted from two separate models should be the same. Similarly, the relative rotation between models should also be constant for all patches.

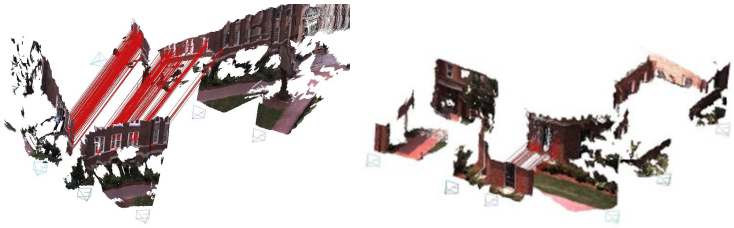

Fig. 14 3D registration of 2 3D models with 45° viewing direction change using VIP features.

Therefore, these can be verified independently. This allows a very effective Hierarchical Efficient Hypothesis Testing (HEHT) scheme. We first verify relative scale by finding the dominant scale and remove all potential feature matches with inconsistent scales. Next, we find the dominant rotation and eliminate outliers and finally we verify inliers for the dominant translation. It turns out that is approach is particularly effective on urban scenes with many repeating structures and only few good matches supporting the correct hypothesis. The reason is that repeated structures generally support the right scale and –for structures repeating on the same or parallel planes– also the right orientation. For the example shown in **Fig. 14** (left), there were 2085 potential VIP matches, 1224 scale inliers, 654 rotation and scale inliers and 214 true inliers. For the example shown on the right of Fig. 14 there were 494 potential VIP matches, 141 scale inliers, 42 rotation and scale inliers and 38 true inliers. For this last example alternative 2D registration approaches failed to return any valid solution. The viewpoint invariance allows the detection of loops with much more significant viewpoint changes. The hierarchical matching enables robustness to repetitive structures and very high levels of outliers ($> 90\%$). It was recently shown that this scheme can also be very effective for location recognition from stereo images in the context of robotics[14].

4. Additional calibration and motion estimation issues

To be able to fuse the models of the different cameras of our capture system into one coherent 3D model we need to determine a common coordinate system for the reconstructions of each individual camera. In the case of known GPS

tracking this can be solved by calibrating all cameras internally and registering them into a common coordinate system for which relative scale to the world coordinate system is known, as well as translation and orientation difference to the world coordinate system. In Section 4.1 we provide more detail about the method for internal calibration and external calibration of all cameras into a single common coordinate system. Even with a calibrated (multi-)camera system it is often not straight-forward to determine the scale of the vehicle motion. In Section 4.2 we discuss several approaches to obtain the absolute scale of motion from cameras mounted on a vehicles.

4.1 Mirror-based calibration of non-overlapping cameras

For many mapping and robotics applications a wide field of view (FOV) is required of the cameras system. This can be achieved using omnidirectional sensors such as cata-dioptric or fish-eye lenses, or by using camera clusters. In both cases calibration poses specific challenges. In[55),56)] we have proposed a simple self-calibration approach for rotating omnidirectional cameras based on multi-view geometry. When high resolution and high frame rates are desired, camera clusters can be a better choice. To obtain the external calibration for the different cameras of the capture system traditional calibration pattern based methods[59),72)] can not be used due to the fact that the fields of view of the cameras have no overlap. To establish an external calibration of all camera into a single coordinate system we deploy our recently proposed technique for the calibration of non-overlapping cameras using a mirror and a standard calibration pattern[29)]. Our technique places one calibration pattern in a fixed position to define the reference coordinate frame for the external calibration of the system. This pattern is typically not seen by any of the cameras or only a very few cameras. Then we use a planar front surface mirror to enable each camera to observe the calibration pattern under multiple mirror positions. Since for the internal calibration of the cameras the mirroring of the cameras does not have any influence we can use any standard technique[59),72)] for pattern based internal calibration directly.

A byproduct of these standard calibration methods is the camera position for each frame captured that shows the pattern reflected by the mirror. Given that

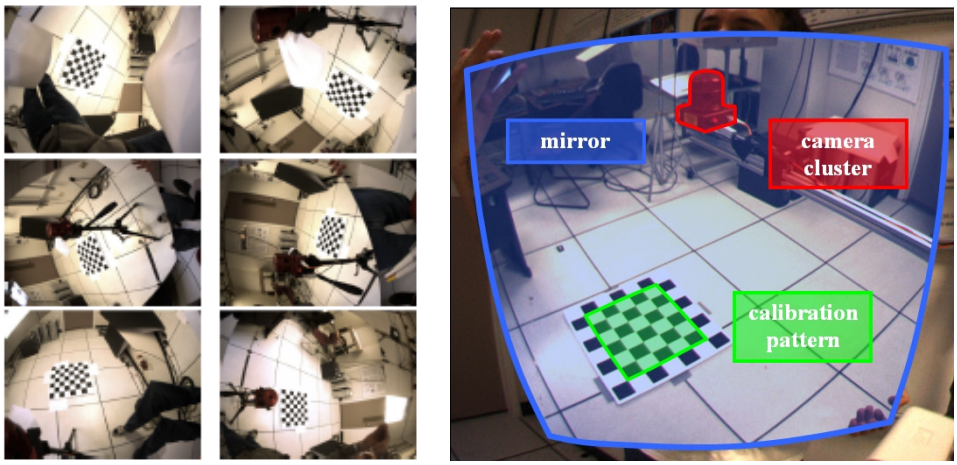

Fig. 15 The calibration setup up for a six camera head is shown on the left. On the right a set of example calibration frames as observed by the cameras are provided.

the camera frame shows the reflected pattern the reconstructed camera pose for each frame is also the reflected camera pose. This set of reflected camera poses describes a three-dimensional family of camera poses corresponding to the three degrees of freedom for the mirror position, which are the two off-mirror plane rotations and the translation along the mirror normal. Since the mirror positions are unknown from the frames captured for the calibration the camera position in the pattern coordinate system can not be computed through inverse reflection. We showed that in fact the observed three dimensional family of mirrored camera poses determines the real cameras position and orientation uniquely without requiring known mirror positions[29] from as few as two images (five when using only linear equations). The calibration accuracy obtained through this method are comparable to the precision obtained from standard calibration methods like[59),72)]. One could also imagine a self-calibration approach based on the shared motion (up to the relative pose) of the different cameras in the cluster. This idea was explored for rotated cameras[3] and for orthographic cameras[2], but more work is needed for an approach that works well in the general case. In the next section we discuss how to obtain the true world scale even in the absence of GPS measurements.

4.2 Absolute scale estimation of motion

The standard way to get the absolute scale in motion estimation is the use of a stereo setup with a known baseline, e.g.[7),37)]. The fields of views of the two cameras need sufficient overlap and motion estimation is done by triangulating feature points, tracking them, and estimating new poses from them. In[4),25)] we developed algorithms that could compute the absolute scale of motion even without overlap between the two cameras. From independently tracked features in both cameras and with known baseline, full 6DOF motion can be estimated. Another approach[6)] makes use of a minimally overlapping stereo pair, which maximizes the field of view of the combined system but leaves some minimal overlap to help compute the absolute scale.

For the case of interest in this paper, where the camera is mounted on a wheeled vehicle, we demonstrated recently that it is even possible to compute the absolute scale of the motion from a single camera only[47)] (for the planar motion case). This is possible due to the non-holonomicity of a wheeled vehicle. A wheeled vehicle (e.g. car, bike, etc.) that is constructed to follow the Ackermann steering principle will undergo locally circular motion[21)]. In particular, any point on the vertical plane containing the fixed rear axle performs a circular motion, the others will not. A camera, that is not located at the rear axle, will undergo a planar motion, different than that of the circular motion of the car coordinate center. From this camera motion and a measurement of the offset from the rear axle (in meters) the absolute scale of the camera motion can be computed. This makes it possible to upgrade an up-to-scale motion estimate for the camera to an absolutely scaled motion. The absolute scale can be estimated at multiple location throughout the vehicle's path. From these points, the absolute scale can be propagated through structure from motion. **Fig. 16** shows results of the absolute scale estimation on a 3 km long camera path. To achieve accurate measurements the absolute scale is only estimated at specific spots where circular vehicle motion is ensured and the turning angle is sufficiently high. It is also important to take measures to successfully propagate scale information in bundle adjustment[14)].

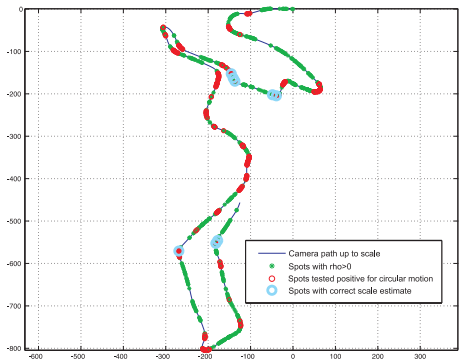

Fig. 16 Absolute scale results on a 3km camera path. The blue circles show spots where the absolute scale was computed. To achieve accurate measurements the absolute scale is only estimated at specific spots where circular vehicle motion is ensured and the turning angle is high.

5. Discussion and conclusion

In this paper we have discussed a video-rate processing pipeline for large-scale scene reconstruction from ground reconnaissance video. To obtain the high performance reconstructions the system deploys the graphics processing unit throughout the different computation steps of the reconstruction pipeline. The current system relies on GPS for consist modeling of large areas, but we discussed progress on loop-closing and other techniques to enable consistent large scale reconstruction. In particular, the VIP features presented in this paper were shown to be very effective for closing loops in challenging circumstances. Also, depending on the camera configuration, different methods exist to recover and maintain the correct absolute scale of the motion. However, while many of the subproblems now have solutions, many challenges remain to develop an automatic system for wide area reconstruction that does not rely on GPS. Solving this will be important for example to allow the deployment of robots that can operate fully autonomously in large buildings with vision as their main sensor. The techniques discussed here are also important for other applications such as image-based localization. The possibility to determine the location from an image for example is very important to enable advanced location-based services for mobile phones. Although the hardware for our current real-time

system is only a single PC, future applications would greatly benefit from the possibility to perform localization and mapping functions on smaller and more energy efficient embedded platforms. We are currently investigating the possibility of performing visual SLAM on very small embedded platforms to support autonomous navigation of micro-aerial vehicles. Other related projects we are currently pursuing are image-based localization for mobile phones and visual SLAM for humanoid robots. Notice that for autonomous robot navigation, besides performing traditional visual SLAM with sparse features, we would also aim to recover a dense surface model and free space from images.

References

1) S. Agarwal, N. Snavely, I. Simon, S. Seitz and R. Szeliski "Building Rome in a Day", *Proc. Int. Conf. on Computer Vision*, 2009.

2) R. Angst, M. Pollefeys, "Static Multi-Camera Factorization Using Rigid Motion", *Int. Conf. on Computer Vision*, 2009.

3) Y. Caspi, M. Irani, Aligning Non-Overlapping Sequences, *Int. J. of Computer Vision*, Vol. 48, Issue 1, pp. 39–51, 2002.

4) B. Clipp, J.-M. Frahm, M. Pollefeys, J.-H. Kim, R. Hartley, "Robust 6DOF Motion Estimation for Non-Overlapping Multi-Camera Systems", *Proc. IEEE Workshop on Applications of Computer Vision (WACV'08)*, 8 pages, 2008.

5) B. Clipp, R. Raguram, J.-M. Frahm, G. Welch, M. Pollefeys "A Mobile 3D City Reconstruction System", *Workshop on Virtual Cityscapes*, IEEE Virtual Reality, 2008.

6) B. Clipp, C. Zach, J.-M. Frahm, M. Pollefeys, "A New Minimal Solution to the Relative Pose of a Calibrated Stereo Camera with Small Field of View Overlap", *Proc. Int. Conf. on Computer Vision*, 8 pages, 2009.

7) B. Clipp, C. Zach, J. Lim, J.-M. Frahm and M. Pollefeys, "Adaptive, Real-Time Visual Simultaneous Localization and Mapping", *Proc. IEEE Workshop on Applications of Computer Vision (WACV'09)* , 2009.

8) D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis." *Proc. IEEE Trans. Pattern Anal. Mach. Intell*, 24(5), 2002

9) N. Cornelis, K. Cornelis, L. Van Gool, "Fast Compact City Modeling for Navigation Pre-Visualization". *Proc. CVPR06 (IEEE Conf. on Computer Vision and Pattern Recognition)*, vol.2, pp.1339–1344, 2006.

10) M. Cummins, P. Newman, "FAB-MAP: Probabilistic Localisation and Mapping in the Space of Appearance", *Int. J. of Robotics Research*. June 2008.

11) M. Fischler, R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Comm. of the ACM* 24:381-395, June 1981.

12) J.-M. Frahm, M. Pollefeys, "RANSAC for (Quasi-)Degenereate data (QDEGSAC)", *Proc. CVPR'06 (IEEE Conf. on Computer Vision and Pattern Recognition)*, 2006.

13) F. Fraundorfer, C. Wu, M. Pollefeys, "Combining monocular and stereo cues for mobile robot localization using visual words", *Proc. ICPR'10 (IEEE Int. Conf. on Pattern Recognition)*.

14) F. Fraundorfer, D. Scaramuzza, M. Pollefeys, "A Constricted Bundle Adjustment Parameterization for Relative Scale Estimation in Visual Odometry", *Proc. ICRA'10 (IEEE Int. Conf. on Robotics and Automation)*.

15) F. Fraundorfer, J.-M. Frahm, M. Pollefeys, "Visual Word based Location Recognition in 3D models using Distance Augmented Weighting", *Proc. 3DPVT'08 (Int. Symp. on 3D Data Processing, Visualization and Transmission)*.

16) C. Früh and A. Zakhor, "An Automated Method for Large-Scale, Ground-Based City Model Acquisition", *Int. J. of Computer Vision*, 60(1), pp. 5 - 24, 2004.

17) D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, M. Pollefeys, "Real-Time Plane-sweeping Stereo with Multiple Sweeping Directions", *Proc. CVPR'07 (IEEE Conf. on Computer Vision and Pattern Recognition)*.

18) D. Gallup, J.-M. Frahm, P. Mordohai, M. Pollefeys, "Variable Baseline/Resolution Stereo", *Proc. CVPR'08 (IEEE Conf. on Computer Vision and Pattern Recognition)*.

19) D. Gallup, J.-M. Frahm, M. Pollefeys, "Piecewise Planar and Non-Planar

Stereo for Urban Scene Reconstruction", *Proc. CVPR'10 (IEEE Conf. on Computer Vision and Pattern Recognition).*

20) D. Gallup, J.-M. Frahm, M. Pollefeys, "A Heightmap Model for Efficient 3D Reconstruction from Street-Level Video", *Proc. 3DPVT'10 (Symposium on 3D Data Processing, Visualization and Transmission).*

21) T. Gillespie, *Fundamentals of Vehicle Dynamics*, Warrendale: SAE, Inc., 1992.

22) R. Haralick, C.-N. Lee, K. Ottenberg, M. Nolle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *Int. J. of Computer Vision*, 13(3), pp.331–356, 1994.

23) R. Hartley, P. Sturm, "Triangulation", *Computer Vision and Image Understanding (CVIU)*, 68(2):146157, 1997.

24) A. Irschara, C. Zach, J.-M. Frahm, H. Bischof, "3D Scene Summarization for Efficient View Registration", *Proc. CVPR'09 (IEEE Conf. on Computer Vision and Pattern Recognition)*, 2009

25) J.-H. Kim, R. Hartley, J.-M. Frahm and M. Pollefeys, "Visual Odometry for Non-Overlapping Views Using Second-Order Cone Programming", *Proc. ACCV'07 (Asian Conf. on Computer Vision).*

26) S. J. Kim, D. Gallup, J.-M. Frahm, A. Akbarzadeh, Q. Yang, R. Yang, D. Nister, M. Pollefeys, "Gain Adaptive Real-Time Stereo Streaming", *Proc. Int. Conf. on Computer Vision Systems*, 2007.

27) S.-J. Kim, J.-M. Frahm, M. Pollefeys, "Joint Feature Tracking and Radiometric Calibration from Auto-Exposure Video", *Proc. ICCV'07 (Int. Conf. on Computer Vision).*

28) S.J. Kim, M. Pollefeys, "Robust Radiometric Calibration and Vignetting Correction", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 562-576, Apr. 2008.

29) R. K. Kumar, A. Ilie, J.-M. Frahm, M. Pollefeys, "Simple calibration of non-overlapping cameras with a mirror", *Proc. CVPR'08 (IEEE Int. Conf. on Computer Vision and Pattern Recognition).*

30) X. Li, C. Wu, C. Zach, S. Lazebnik and J.-M. Frahm, "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs",

Proc. ECCV'08 (European Conf. on Computer Vision), 2008.

31) D. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. of Computer Vision*, 60, 2 (2004), pp. 91-110.

32) R. Matsuhisa, H. Kawasaki, S. Ono, A. Banno and K. Ikeuchi, "Extensive Urban City Model Construction using Multiple Omnidirectional Image Sequences taken by Vehicle Camera", MIRU 2009 Meeting on Image Recognition and Understanding, July 2009.

33) P. Merrell, P. Mordohai, J.M. Frahm, M. Pollefeys, "Evaluation of Large Scale Scene Reconstruction", *Proc. Workshop on Virtual Representations and Modeling of Large-scale environments (VRML'07).*

34) P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, M. Pollefeys, "Fast Visibility-Based Fusion of Depth Maps", *Proc. ICCV'07 (Int. Conf. on Computer Vision).*

35) K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, A comparison of affine region detectors. *Int. J. of Computer Vision* 65(1/2):43-72, 2005

36) D. Nister. "An efficient solution to the five-point relative pose problem". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):756777, 2004.

37) D. Nister, O. Naroditsky, J. Bergen, "Visual odometry for ground vehicle applications", *J. of Field Robotics*, 23(1), 2006.

38) D. Nister and H. Stewenius. "Scalable recognition with a vocabulary tree", *Proc. CVPR'06 (IEEE Conf. on Computer Vision and Pattern Recognition)*, vol. 2, pp. 2161–2168, 2006.

39) R. Pajarola. "Overview of quadtree-based terrain triangulation and visualization". Tech. Report UCI-ICS-02-01, Information & Computer Science, University of California Irvine, 2002.

40) M. Pollefeys, R. Koch and L. Van Gool. "Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters", *Int. J. of Computer Vision*, 32(1), 7-25, 1999.

41) M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, "Visual modeling with a hand-held camera", *Int. J. of*

Computer Vision 59(3), 207-232, 2004.

42) M. Pollefeys, D. Nister, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, "Detailed Real-Time Urban 3D Reconstruction From Video", *Int. J. of Computer Vision*, 78(2), pp.143–167, 2008.

43) R. Raguram, J.-M. Frahm, M. Pollefeys, "A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus", *Proc. ECCV'08 (European Conf. on Computer Vision).*

44) R. Raguram, J.-M. Frahm, M. Pollefeys, "Exploiting Uncertainty in Random Sample Consensus", *Proc. ICCV'09 (Int. Conf. on Computer Vision).*

45) O. Saurer, F. Fraundorfer, M. Pollefeys, "OmniTour: Semi-automatic generation of interactive virtual tours from omnidirectional video", Proc. 3DPVT2010 (Int. Symp. on 3D Data Processing, Visualization and Transmission).

46) D. Scaramuzza, F. Fraundorfer, M. Pollefeys, R. Siegwart. "Closing the Loop in Appearance-Guided Structure-from-Motion for Omnidirectional Cameras." *Proc. Eight Workshop on Omnidirectional Vision*, ECCV 2008.

47) D. Scaramuzza, F. Fraundorfer, M. Pollefeys, R. Siegwart, "Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints", *Proc. ICCV'09 (Int. Conf. on Computer Vision).*

48) G. Schindler, M. Brown, R. Szeliski, "City-Scale Location Recognition", *Proc. CVPR'07 (IEEE Conf. on Computer Vision and Pattern Recognition).*

49) G. Schindler, F. Dellaert and S.B. Kang, "Inferring Temporal Order of Images From 3D Structure", *Proc. CVPR'07 (IEEE Conf. on Computer Vision and Pattern Recognition).*

50) J. Shi and C. Tomasi, "Good Features to Track", *CVPR94 (IEEE Conf. on Computer Vision and Pattern Recognition)*, pages 593-600, 1994.

51) S. Sinha, P. Mordohai, M. Pollefeys, "Multi-View Stereo via Graph Cuts on the Dual of an Adaptive Tetrahedral Mesh", *Proc. ICCV'07 (Int. Conf.*

on Computer Vision).

52) S. Sinha, J.-M. Frahm, M. Pollefeys, Y. Genc, Feature Tracking and Matching in Video Using Programmable Graphics Hardware, *Machine Vision and Application*, online Nov. 2009.

53) S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, M. Pollefeys, "Interactive 3D Architectural Modeling from Unordered Photo Collections", *ACM Trans. on Graphics (SIGGRAPH ASIA 2008)*, 27(5), December 2008, pp. 159:1–10.

54) N. Snavely, S. Seitz, R. Szeliski, "Photo Tourism: Exploring image collections in 3D", *Proc. of SIGGRAPH 2006.*

55) S. Thirthala, M. Pollefeys, "The Radial Trifocal Tensor: A Tool for Calibrating Radial Disortion of Wide-Angle Cameras", *Proc. CVPR'05 (IEEE Conf. on Computer Vision and Pattern Recognition)*, Vol. 1, pp. 321–328, 2005.

56) S. Thirthala, M. Pollefeys, "Multi-view geometry of 1D radial cameras and its application to omnidirectional camera calibration.", *Proc. ICCV'05 (Int. Conf. on Computer Vision)*, Vol. 2, pp. 1539–1546, 2005.

57) C. Tomasi and T. Kanade. "Shape and motion from image streams under orthography—a factorization method". *Int. J. of Computer Vision*, 9(2):137–154, 1992.

58) B. Triggs, P. McLauchlan and R. Hartley and A. Fitzgibbon (1999). "Bundle Adjustment A Modern Synthesis". *ICCV '99: Proc. of the Int. Workshop on Vision Algorithms*. Springer-Verlag. pp. 298–372.

59) R.Y.Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses" *IEEE J. for Robotics and Automation*, 3, pages 323–344, 1987.

60) T. Tuytelaars and C. Schmid, "Vector Quantizing Feature Space with a Regular Lattice", *Proc. ICCV'07 (Int. Conf. on Computer Vision).*

61) C. Wu, B. Clipp, X. Li, J.-M. Frahm, M. Pollefeys, "3D Model Matching with Viewpoint Invariant Patches (VIPs)", *Proc. CVPR'08 (IEEE Conf. on Computer Vision and Pattern Recognition).*

62) C. Wu, F. Fraundorfer, J.-M. Frahm and M. Pollefeys, "3D Model Search

and Pose Estimation from Single Images using VIP Features", *Proc. S3D workshop*, CVPR'08.

63) C. Wu, open source SIFTGPU, `http://www.cs.unc.edu/~ccwu/siftgpu/`

64) J. Xiao, T. Fang, P. Zhao, M. Lhuillier, L. Quan, "Image-Based Street-Side City Modeling", *ACM Trans. on Graphics (SIGGRAPH ASIA)*, 2009.

65) R. Yang and M. Pollefeys, "Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware", *Proc. CVPR'03 (IEEE Conf. on Computer Vision and Pattern Recognition)*, pp. 211-218, 2003.

66) R. Yang, M. Pollefeys, and G. Welch. "Dealing with Textureless Regions and Specular Highlight: A Progressive Space Carving Scheme Using a Novel Photo-consistency Measure", *Proc. ICCV'03 (Int. Conf. on Computer Vision)*, pp. 576-584, 2003.

67) R. Yang, M. Pollefeys, "A Versatile Stereo Implementation on Commodity Graphics Hardware", *J. of Real-Time Imaging*, Volume 11, Issue 1, February 2005, Pages 7-18.

68) C. Zach, T. Pock, and H. Bischof. "A globally optimal algorithm for robust TV-L1 range image integration", *Proc. ICCV'07 (IEEE Int. Conf. on Computer Vision)*, 2007.

69) C. Zach, D. Gallup, and J.-M. Frahm, "Fast Gain-Adaptive KLT Tracking on the GPU", *CVPR Workshop on Visual Computer Vision on GPU's (CVGPU)*, 2008.

70) C. Zach, M. Klopschitz, M. Pollefeys, "Disambiguating Visual Relations Using Loop Constraints", *Proc. CVPR'10 (IEEE Int. Conf. on Comp. Vision and Pattern Recognition)*.

71) C. Zach, open source code for GPU-based KLT feature tracker and sparse bundle adjustment: `http://www.cs.unc.edu/~cmzach/opensource.html`

72) Z. Zhang, "A flexible new technique for camera calibration", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp.1330–1334, 2000.