Generalized Detection and Merging of Loop Closures for Video Sequences

Manfred Klopschitz[1], Christopher Zach[2], Arnold Irschara[1], Dieter Schmalstieg[1]

[1]Graz University of Technology
{klopschitz,irschara,schmalstieg}@icg.tugraz.at
[2]University of North Carolina, Chapel Hill
cmzach@cs.unc.edu

Abstract

In this work we present a method to detect overlaps in image sequences, and use this information to integrate overlapping sparse 3D structure from video sequences. The additional temporal information of these images is used to increase robustness over single image pair matching. A scanline optimization problem formulation is used to compute the best sequence alignment using wide-baseline image matching techniques. Compared to a direct dynamic programming approach, the scanline optimization formulation increases the robustness of sequence alignment for general relative motions. The proposed alignment method is employed to integrate sparse 3D models reconstructed from separate video sequences. In addition loop closures are detected. Consequently, the 3D modeling process from sequential image data can be split into fast sequence processing and subsequent global integration steps.

1. Introduction

Current Structure from Motion (SfM) methods may be classified based on the image data they use. Two major types of image sources are still images and video sequences. This choice of input data usually has strong implications on the selection of algorithms used to solve the SfM problem. Unordered sets of still images relay on wide baseline image matching techniques to establish correspondence information and the SfM problem may be solved for all input images simultaneously or in an incremental way. Two examples of this type of systems are [14] and [26]. The correspondence problem can be reduced to a tracking problem when a video stream is used as input. The practical implication of this is that more difficultly textured scenes can be reconstructed, i.e. repetitive structures. Additionally can the SfM problem be solved efficiently in real time, as described for example in [16]. Using video based tracking to capture complex scenes is difficult in practice because one usually cannot capture the scene in just one take that is suitable for feature point tracking.

In this paper we examine how to extend the concept of image matching to the matching of consecutive video sequences and apply this matching technique in a 3D vision system. We want to combine the advantages of video with the advantages of global reconstruction optimization. The ultimate goal is to be able to provide at least partial reconstructions in near real time and integrate them into a larger database so that the user can be informed about the quality of the reconstruction and missing areas and add additional data selectively. We exploit the spatial-temporal relation of video sequence images to increase the flexibility of video 3D vision systems. Using the additional sequential information for the matching process also increases the robustness compared to single image matching.

Many reconstruction systems in computer vision are based on images from a moving video camera. These video based systems can use uncalibrated [20] or calibrated cameras [15] [16] [18] and are applied for example to cultural heritage modeling, odometry, robot navigation and city modeling. Multi-camera heads can be used to extend the field of view [1]. These methods are particularly appropriate to create large sparse reconstructions of continuous movements in real time.

Another source of images with additional sequential information are vision based Simultaneous Localization and Mapping (SLAM) methods. Incremental map building and continuous localization increase the robustness of SfM as demonstrated in [4] [3] [5]. The area that can be covered is limited by the number of landmarks that can be recognized and optimized efficiently. Furthermore, classical SLAM

[0]In part funded by FWF Y193, W1209-N15,FP6-2004-IST-4-27571, Vienna Science and Technology Fund (WWTF).

algorithms use every single image from the video stream for the tracking and mapping operations. To handle these amounts of data, bundle adjustment is replaced by simpler methods to integrate the mapped landmarks. An exception is [12], where tracking and mapping is split into separate tasks, but this SLAM approach is explicitly designed for small workspaces.

A logical extension of visual odometry based reconstructions is the integration of multiple sequences. We propose a method to integrate multiple sparse reconstructions from a visual odometry front end into a global coordinate system. Loops are a special case of overlaps of sequences and structure that provide a way to reduce drift from an odometry trajectory. An example of a system where loops are detected in a sparse reconstruction to reduce this drift can be found in [28]. In [9] the additional sequential information of image sequences is used for loop closing and therefore drift reduction in robot navigation.

In our work flow continuous, sequential sparse reconstructions from a scene are acquired using a visual odometry approach. The output of this step is a camera trajectory, sparse 3D points and the images used for visual odometry. These images have the property that they have been obtained consecutively and contain additional sequential information. The sequences are fed into a batch process where overlaps are detected, i.e. generalized loops. With this sequence overlaps, the 2D-3D correspondences are relinked. Bundle adjustment is then used to integrate sequences into one coordinate system and reduce drift simultaneously.

2. Image based Sequence Similarity

This section describes how a first matching of individual images of two sequences is done efficiently. The matching score of image sequences uses a similarity score of individual image pairs. The image pair similarity score is feature based. A vocabulary tree can be used to avoid the matching of all image pairs. The extracted features are further used to fuse the sparse reconstructions of the sequences. Currently our implementation uses SIFT features [13].

2.1. Image Sequences Extraction

The image sequences that we use are extracted with a monocular SfM framework similar to [16]. Our SfM approach takes a video from a single calibrated camera as input and computes the camera motion and sparse scene structure in an incremental way. Only a subset of key-frames from the video input is used in the SfM framework. This subset of images is chosen so that the baseline and feature track number between key-frames is optimized. Only this subset of key-frames is used for sequence matching, Figure 1 shows two example sequences.

2.2. Image Matching

The cost of matching all image pairs of two sequences would severely limit the possible sequence lengths. Retrieving similar images for a given one is currently a very active research topic e.g. [21, 19, 11]. To speed up the pairwise matching we employ a visual vocabulary tree approach similar to [19]. The vocabulary tree enables us to efficiently match a single image against all images in the sequence.

In our system the vocabulary tree is trained in an unsupervised manner with a subset of 2×10^6 SIFT feature vectors randomly taken from 2500 images. The descriptor vectors are then hierarchically quantized into clusters using a k-means algorithm. We set the branch factor to 10 and allow up to 7 tree levels. For each level the k-means algorithm is initialized with different seed clusters and the result with the lowest Euclidean distance error is retained. Once the vocabulary tree is trained, searching the visual vocabulary is very efficient and new images can be inserted on-the-fly.

In our current setting we rely on an entropy weighted scoring similar to the *tf-idf* "term frequency inverse document frequency" as described in [24]. Let D be an image in our database and t be the term in the vocabulary associated to feature f of the current query image Q, then our scoring function is,

$$\sum_{t \in Q \cap D} \log \left(\frac{N}{n(t)} \right) \tag{1}$$

where N is the total number of images in the collection and $n(t)$ is the number of images that contain term t. In order to guarantee fairness between database images with different number of features, the query results are normalized by the self-scoring result.

2.3. Visual Similarity Matrix

Given two image sequences s_1 and s_2, the image features of s_1 are inserted into an empty vocabulary tree to create the inverted file structure. For each image in s_2 a query with the the vocabulary tree and the scores of the k best matching images are returned. The obtained matching scores are used to construct a Visual Similarity Matrix (VSM). Figure 2 shows the VSM obtained from the two sequences of Figure 1. Each element $e_{i,j}$ of the VSM corresponds to an image similarity between image i of the first sequence s_1 and image j of the second image sequence s_2. Each row of the VSM has at most k non-zero entries.

Our experiments have shown that it is sufficient to use the image similarity scores from the vocabulary tree directly to construct a VSM. No further image to image fea-

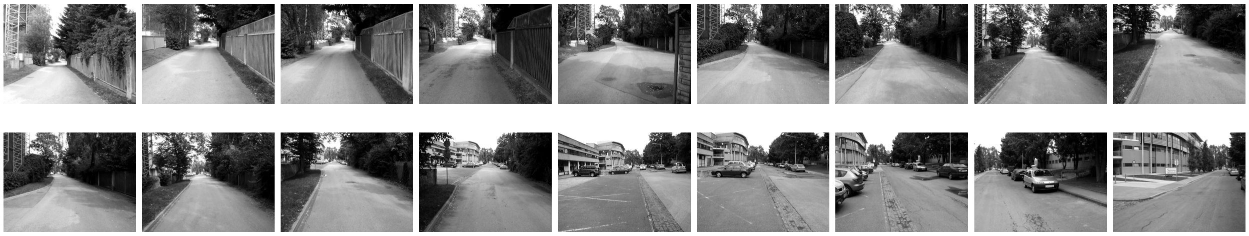

Figure 1. Each row shows an image sequence. Only a subset of key-frames obtained from the SfM input video is shown.

ture matching or geometric verification is done to enhance the scoring accuracy at this stage.

3. Sequence Alignment

After computing the VSM, a contiguous path of corresponding images in this matrix can be extracted to represent the video sequence overlap. This section describes our approach to solve this problem. An optimal local sequence alignment can be computed in principle using the well known Smith-Waterman [25] algorithm. This dynamic programming algorithm is used for example in bioinformatics to align protein or nucleotide sequences. *Local* sequence alignment includes the ability to detect and match only subsequences of the input and to ignore non-corresponding sections. *Global* sequence alignment can be achieved by a slightly simpler variant of the Smith-Waterman algorithm, the Needleman-Wunsch [17] method.

A limitation of this approach for image sequence matching is that only sequence overlaps in the forward direction of relative movement can be obtained. This is due to the ordering constraint inherent in all classical dynamic programming approaches. This means that image sequence matching has to be done two times for a sequence pair if the relative sequence movement is not known a priory. In more complex cases, where the relative movement direction of a sequence pair changes multiple times, the Smith-Waterman algorithm can only find sequence parts with consistent relative movement. Hence, the full overlapping sequence is unnecessarily split into several subsequences, which need to be merged in a post-processing step. We propose scanline optimization for local sequence alignment to find longer matching sequences, and therefore to avoid any later post-processing step.

We propose a variation of scanline optimization [22] to compute the best sequence alignment. In general, scanline optimization is a dynamic programming approach to determine the maximum a posteriori solution of 1-D Markov random fields. Most prominently, it is used in several methods for dense depth estimation from stereo images [22, 8]. In

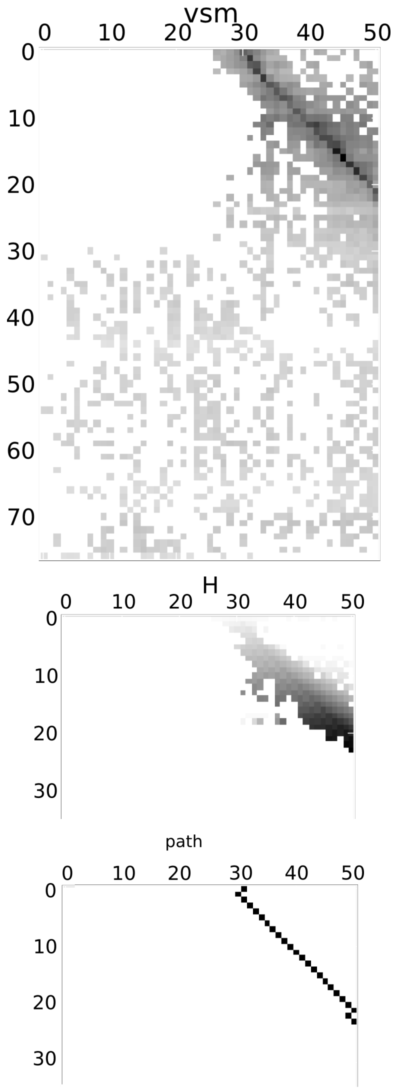

Figure 2. VSM matrix of the two sequences from Figure 1 and the resulting dynamic programming matrix H and the extracted correspondence path for the two sequences.

contrast to earlier Dynamic Programming (DP) approaches for stereo, scanline optimization does not enforce the ordering constraint. In our application, this feature enables sequence alignment for more general motions, which we consider the main advantage of our method compared with DP-based ones. A slight drawback of scanline optimization is the non-commutativity, i.e. the returned path for swapped inputs is not just the transpose of the original path.

3.1. Scanline Optimization Problem Formulation

Scanline optimization computes the optimal assignment of a sequence of images x_i to corresponding images of another sequence d_x. It finds the value of

$$\text{score}(x, d) = \arg\min_{d_x} \sum_{i=1}^{N} \Big(D(x_i, d_x) + \lambda V(d_x, d_{x-1}) \Big), \tag{2}$$

where $D(x, d) = -S(x, d)$ is the dissimilarity score of two images at positions x and $x + d$, and $V(d, d')$ is the regularisation cost and λ weights the relative influence of these two factors.

In order to obtain similar results in cases of pure forward/backward motion, we model the regularisation to approximate the moves favored by the Smith-Waterman algorithm. The smallest regularisation cost, zero, is assigned to diagonal moves, i.e. if $|d_{x-1} - d_x| = 1$. The cost of any occlusions in the image sequence (i.e. skipping images) is equal to the number of skipped frames. E.g., moving in one image, but not in the other ($d_{x-1} = d_x$) has cost one. More formally, the regularization cost for two successive image assignments d_{x-1} and d_x is given by:

$$V(d_x, d_{x-1}) = \begin{cases} i-1 & \text{if } d_x = d_{x-1} - i,\ i = 2, 3, \ldots \\ 0 & \text{if } d_x = d_{x-1} \pm 1, \\ 1 & \text{if } d_x = d_{x-1}, \\ i-1 & \text{if } d_x = d_{x-1} + i,\ i = 2, 3, \ldots \end{cases}$$

3.2. Efficient Minimization

Minimizing Eq. 2 and determining the corresponding optimal assignment d_x can be efficiently performed using a dynamic programming approach by maintaining the minimal accumulated costs $H(x, d)$ up to the current position in the first image sequence x:

$$\begin{aligned} H(x, d) &= D(x, d) + \\ &\quad \min_{d'} \left(H(x-1, d') + \lambda V(d, d') \right). \end{aligned}$$

We have the initial values $H(1, d) = D(1, d)$. Note, that our specific choice of $V(\cdot, \cdot)$ can be written as

$$V(d, d') = \min \left(|d + 1 - d'|, |d - 1 - d' \right), \tag{3}$$

i.e. it is the minimum of two linear discontinuity cost functions. Consequently,

$$\begin{aligned} &\min_{d'} \big(H(x-1, d') + \lambda V(d, d') \big) = \\ &\qquad \min \Big\{ \min_{d'} \left(H(x-1, d') + \lambda |d + 1 - d'| \right), \\ &\qquad\qquad \min_{d'} \left(H(x-1, d') + \lambda |d - 1 - d'| \right) \Big\}. \end{aligned}$$

Following [6], the simultaneous calculation of the sub expressions

$$\min_{d'} \left(H(x, d') + \lambda |d + 1 - d'| \right) \tag{4}$$

and

$$\min_{d'} \left(H(x, d') + \lambda |d - 1 - d'| \right) \tag{5}$$

for every d can be performed in linear time using a forward and a backward pass to compute the lower envelope. Hence, the proposed energy can be minimized in $O(n\, m)$ time, where n and m are the lengths of the two sequences, respectively. A direct approach would have $O(n\, m^2)$ time complexity.

The procedure to fill the entries of H is summarized in Algorithm 1. The necessary instructions to maintain the backtracking table for fast subsequent alignment extraction are omitted. This procedure is very similar to the scanline optimization method proposed for stereo, with two main distinctions: first, the discontinuity cost V has a different shape; second, clamping the accumulated cost to 0 indicates the potential termination of a locally aligned sequence. Note that in this application the accumulated costs are less or equal zero.

Figure 2 shows an example of the matrix H and the extracted sequence correspondence path.

Algorithm 1 Dynamic programming scanline optimization

```
Input: Dissimilarity scores $D_{n \times m} = -S_{n \times m}$
$H \leftarrow \mathbf{0}_{n \times m}$
$H[d, :] \leftarrow D[1, :]$
for $x = 2 : n$ do
  { $h$ can be computed in $O(m)$ time using [6]. }
  $\forall d : h[x, d] \leftarrow \min_{d'} \big( H[x-1, d'] + \lambda V(d, d') \big)$
  { Note: $H[x, d] = 0$ terminates the local alignment sequence. }
  $\forall d : H[x, d] \leftarrow \min \big( 0, D[x, d] + h[x, d] \big)$
end for
return $H$
```

3.3. Matching Multiple Sequences

We use an incremental approach to find the overlap of multiple image sequences. To compare multiple sequence matches, the optimal scanline assignment score, $\text{score}(x, d)$, is computed for all pairs.

A slow relative movement that covers only a small amount of structure overlap but contains many images produces a similar score as a larger movement with the same amount of wider placed images. We normalize the sequence matches to favour sequence matches that cover a wide range of structure over slow relative movements. This is done by scaling the matching score with the deviation from an ideal diagonal movement. This leads to the normalized score

$$\text{score}_n(x,d) = \text{score}(x,d)\frac{\text{width}(\text{path},x)\,\text{width}(\text{path},d)}{\|\text{path}\|^2},$$

where path is the sequence of image matches and $\text{width}(\text{path}, a)$ is the number of different images from the sequence a that is contained in the image correspondence path.

4. Sequence Merging

The image sequence correspondences are used to merge the sequences into one coordinate system. We assume that the sparse reconstructions (camera poses and 3D feature points) of the individual sequences are available. The following steps give an overview of the merging process:

Feature matching: For each image correspondence pair between two sequences from the scanline optimization, SIFT features are matched. A modified kd-tree [2] is used to speed the matching up.

Feature tracking: For each sequence, the SIFT features are merged to feature tracks. This is done by matching the features in neighboring images.

Geometric verification: The SIFT tracks are triangulated with the available camera poses of one sequence and the absolute pose [10] of each matching image in the other sequence is computed using RANSAC [7]. This is done to find the reliable correspondences between the two sequences.

Similarity transform: A similarity transform between two sequences is computed with the reliable feature matches and their corresponding triangulated 3D points. The sequences are brought into one coordinate system.

Bundle adjustment: Bundle adjustment [27] is used to refine the merged sequences. We use the feature tracks from the original sequences (corner points that are tracked [23]), the extracted SIFT tracks and their inter-sequence correspondences.

Figure 3 illustrates all used correspondences.

5. Results

We demonstrate our algorithms on two data sets. The first experiment demonstrates the added image matching robustness under weakly textured scenes and the concept of merging of different sparse odometry reconstructions. Three image sequences were captured by hand with a digital compact camera. The Motion JPEG videos with 640×480 pixels resolution were used to compute three separate sparse reconstructions. Figure 4 shows the result of the sequence overlap extraction and geometry merging process. The second data set demonstrates the extraction of image correspondence paths that change their relative movement direction and the loop closing capabilities of the merging process. One Motion JPEG video with 840×480 pixels resolution was used in this experiment as odometry input. A reversed copy of the video was added to the stream before the initial reconstruction was obtained so that the begin and end of the camera trajectory are the same. Figure 5 shows the results.

6. Summary and Conclusions

We presented a method and work flow to integrate multiple sparse reconstructions from video sequences into one coordinate system. To obtain the initial sparse reconstructions fast and robust feature point tracking and visual odometry can be used. The initial sequence overlaps are computed with the key frames from the sparse reconstruction. Matching the whole image sequences using a scanline optimization problem formulation increases the robustness compared to single image pair matching. No 3D structure is used at this point because structure can be severely distorted by drift and is difficult to match between different sequences. The image based sequence overlaps are then used to connect common 3D structure.

Results show that sequence relations can be obtained for data sets even where single images cannot be matched unambiguously. The presented image based techniques are particularly well suited for large scale reconstructions. Loop closing is just a special case of more general sequence overlaps. A limitation in our implementation at the moment is that the actual structure integration work is done using global bundle adjustment. Our sparse bundle adjustment implementation is only suitable for a few hundred images because it does not exploit the second order structure of the problem. Partitioning and hierarchical handling of global reconstructions will be addressed in future work.

References

[1] A. Akbarzadeh et. al. Towards urban 3d reconstruction from video. In *Proc. 3DPVT*, pages 1–8, 2006.

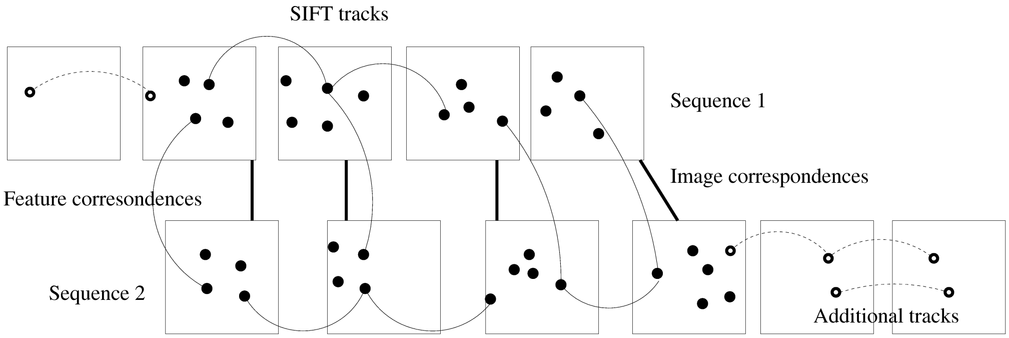

Figure 3. The different correspondences used for sequence merging. Image correspondences: These are the image matches from the scanline sequence matching. Feature correspondences: The matched inter-sequence correspondences that remain after the geometric verification. SIFT tracks: The extracted SIFT points are combined to tracks for each sequence. Additional tracks: The tracked corners from the original sequences.

[2] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. CVPR*, page 1000, 1997.

[3] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. ICCV*, page 1403, 2003.

[4] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *TPAMI*, 29(6):1052–1067, 2007.

[5] E. Eade and T. Drummond. Scalable monocular SLAM. In *Proc. CVPR*, pages 469–476, 2006.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *Proc. CVPR*, pages 261–268, 2004.

[7] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Association for Computing Machinery, 24(6):381–395.*, 1981.

[8] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proc. CVPR*, pages 807–814, 2005.

[9] K. L. Ho and P. Newman. Detecting loop closure with scene sequences. *IJCV*, 74:261–286, 2007.

[10] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5:1127–1135, 1988.

[11] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Proc. CVPR*, 2007.

[12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE International Symposium on Mixed and Augmented Reality(ISMAR)*, November 2007.

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

[15] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. Real time localization and 3d reconstruction. In *Proc. CVPR*, pages 363–370, 2006.

[16] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. 3d reconstruction of complex structures with bundle adjustment: an incremental approach. In *Proc. ICRA*, pages 3055–3061, 2006.

[17] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Bio.*, 48(3):443–453, 1970.

[18] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. CVPR*, pages 652–659, 2004.

[19] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, pages 2161–2168, 2006.

[20] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3):207–232, 2004.

[21] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, pages 414–431, 2002.

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47(1-3):7–42, 2002.

[23] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, Seattle, June 1994.

[24] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pages 1470–1477, 2003.

[25] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Bio.*, 147:195–197, 1981.

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

[27] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–375. 2000.

[28] F. Verbiest and L. V. Gool. Drift detection and removal for sequential structure from motion algorithms. *TPAMI*, 26(10):1249–1259, 2004. Member-Kurt Cornelis.

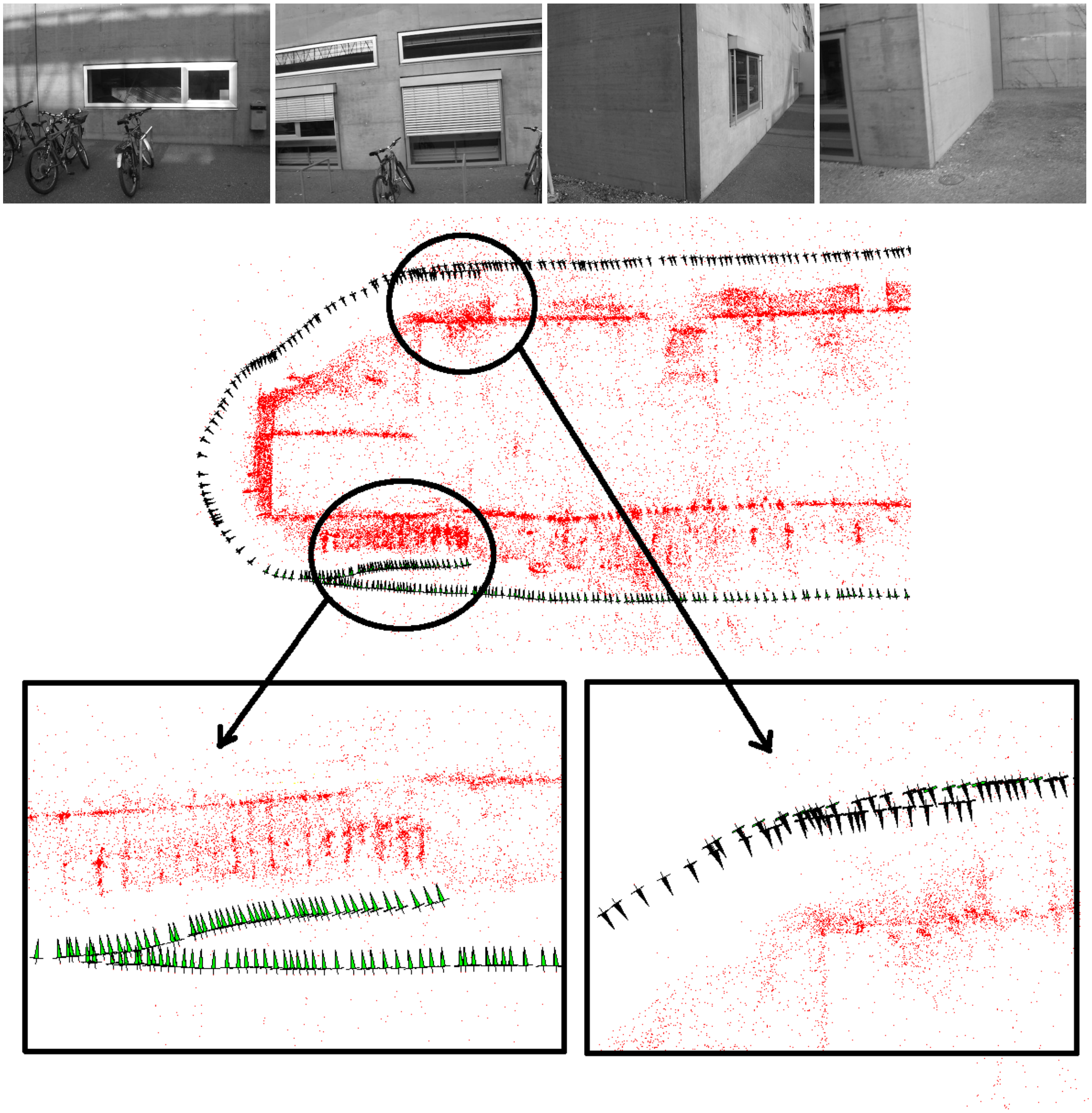

Figure 4. Merging of three odometry sequences in a difficult environment around a building. The pyramids represent camera positions, structure points are shown red. The three sequences overlap in two regions. These regions are highlighted in the zoomed in views. The merged reconstruction contains 449 cameras, the side length of the shown part of the building is about 20 meters.

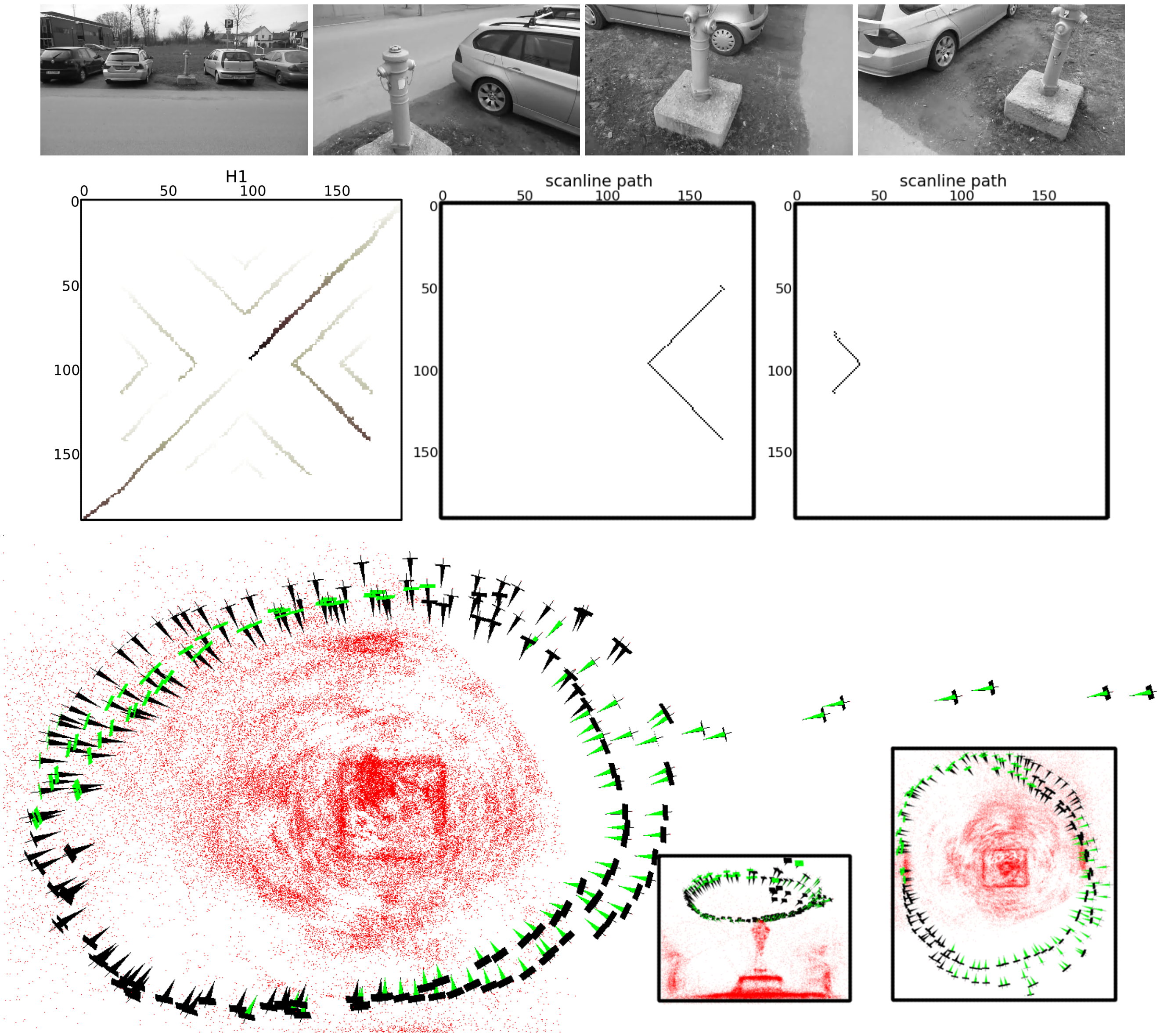

Figure 5. Sequence around a fire hydrant. The camera was moved towards the hydrant and then three loops were obtained. Before starting the initial visual odometry processing, a reversed copy of the video was added to the video stream, so that drift could be noticed easily. The second row shows the dynamic programming matrix H that was obtained by scoring the sequence with itself (the VSM diagonal scores have to be removed when a sequence is matched with itself) and two example correspondence paths. The last row shows the integrated reconstruction with successfully removed drift. Note that the reversed images do not have exactly the same position in space as the original because our odometry system selects images from the video stream dynamically.