Unsupervised Learning for Improving Efficiency of Dense Three-Dimensional Scene Recovery in Corridor Mapping

Thomas Warsop and Sameer Singh

Research School of Informatics, Holywell Park, Loughborough University, Leicestershire, UK, LE11 3TU
T.E.Warsop@lboro.ac.uk, S.Singh@lboro.ac.uk

Abstract. In this work, we perform three-dimensional scene recovery from image data capturing railway transportation corridors. Typical three-dimensional scene recovery methods initialise recovered feature positions by searching for correspondences between image frames. We intend to take advantage of a relationship between image data and recovered scene data to reduce the search space traversed when performing such correspondence matching. We build multi-dimensional Gaussian models of recurrent visual features associated with distributions representing recovery results from our own dense planar recovery method. Results show that such a scheme decreases the number of checks made per feature to 6% of a comparable exhaustive method, whilst unaffecting accuracy. Further, the proposed method performs competitively when compared with other methods presented in literature.

1 Introduction

The term *corridor* has been used to describe a linear, directional flowing, geographic band connecting two points of a transportation service ([20]). *Corridor mapping* is the process by which data is collected regarding such a transportation corridor for the creation of a virtual representation. The work presented in this paper is part of a larger project which is concerned with corridor mapping from a train mounted, forward-facing High-Definition video camera. The ultimate goal of this project is to perform line-of-sight analysis regarding railway assets, using recovered 3D scene data as input to a geometric analysis process. The work presented in this paper is only concerned with the 3D scene recovery from monocular video aspect of this project.

As pointed out by Favaro et al. [7], the majority of 3D scene recovery methods are based on the same principle of matching features between image frames and recovering 3D position using camera geometry. This can be achieved, for example, by tracking image features across image frames ([25, 16]). Typical features used in such scenarios include Harris corners ([13, 15, 6]), SIFT features ([27]) and more recently, SURF features ([1]). Detected feature points are then matched across images. For example, using template matching within a window of possible positions as described by Kanbard et al. [13].

A. Heyden and F. Kahl (Eds.): SCIA 2011, LNCS 6688, pp. 393–402, 2011.

However, these methods only considered 2D information present in the images processed when calculating feature correspondences. It is possible to integrate 3D information into this problem. Using stereo cameras, as can be seen in the work of Ogale et al. [22], Yun et al. [26] and Zhang et al. [29] (to name a few) this can be achieved by searching epipolar scanlines across left and right-hand images for matching feature correspondence, typically with a template matching scheme. It is possible to integrate these epipolar searching concepts into monocular camera configurations. For example, Klein et al. [14] presented a method named *Parallel Tracking and Mapping (PTAM)*. In which features are initialised with their 3D positions by searching along epipolar lines defined by depth between key frames of the image sequence. Davison et al. [4, 5] presented a similar idea. Along the depth-defined epipolar lines, regular intervals were considered and matched in subsequent frames by projecting them into the current frame, using normalised sum-of-square differences template matching.

The previous methods are only concerned with recovery of 3D points. However, it is possible to compute higher-order structures such as planes. In fact, doing so has the advantage that an infinite density of points can be described in only a few parameters. Whereas, with the previously discussed methods, increasing the number of feature points increases computation quadratically ([19]). Further, storing planes collapses state space reducing computation and improving scalability as well as giving a higher-level scene description ([11]). Also, memory requirements are reduced as many points are represented by a few parameters ([18]). There are different ways in which these planes can be computed. For example, Chekhlov et al. [2] and Gee et al. [11, 10] recover 3D points first and then fit planes to this information. Any new points that are subsequently recovered can be added to these created planes. Fraundorfer et al. [8] proposed a method in which initial planar *seed regions* were chosen from which the rest of the planar region could be grown. A different approach taken by Pollefeys et al. [23] tested a reduced set of planes, projecting image pixels onto the chosen planes, using image pixel value differences to select the best. Yet another type of method is presented by both Furukawa et al. [9] and Sinha et al. [24]. Sinha et al. [24] use sparse reconstructed point and line clouds to provide evidence for a set of candidate planes. Planar depth was then recovered for each image by assigning each pixel to one of the candidate planes.

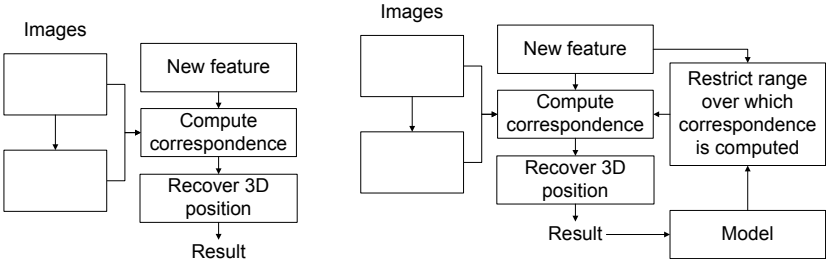
In our application, the 3D scene recovery will be performed in an offline capacity. Thus allowing us to process image sequences in reverse chronological order - presenting two interesting properties. First, new scene elements appear at the image edge, allowing redundant information to be easily ignored. Secondly, image areas recovered in subsequent image frames exhibit similar 3D scene properties when they process similar image properties to those processed previously. This concept is highlighted in Figure 1. It may therefore be possible to exploit this information, using relationships between image features and recovered 3D scenes to reduce the size of the search spaces traversed when computing feature correspondences. Such a concept has not been proposed by previous methods

Fig. 1. When processing image sequences in reverse order, *new* scene elements entering at image edges exhibit similar image and 3D scene properties

and forms the main contribution of this work (named Temporal Search space Reduction, or TSR).

To further clarify the novelty of our proposal, most 3D scene recovery initialise new features by computing correspondences between image frames. Computing these correspondences requires searching a range of values in some capacity (for example, searching along epipolar scan lines for matches). This structure is shown in Figure 2(a). We propose the use of relationships learnt from previously processed image features to reduce the range traversed for correspondence computation. This is shown in Figure 2(b).

The structure of the remainder of this paper is as follows. Section 2 describes a simple, dense 3D scene recovery method and our novel method for learning recurring structures to reduce correspondence range traversal. Section 3 presents experimental results regarding our method. Finally, section 4 concludes this paper.



(a) The general structure of 3D scene recovery methods with regard to the recovery of individual features.

(b) Proposed extension to general 3D scene recovery methods, using previously processed results to reduce search space traversal when looking for correspondence, hence improving efficiency.

Fig. 2. Comparison between general 3D scene recovery methods and that proposed by this work

2 Dense and Efficient 3D Scene Recovery

2.1 Recovering Planar Structures

Figure 1 shows typical images to be processed by our method. These images are typical in that they contain many planar structures to be recovered. For example, the flat ground and almost vertical dense vegetation. Therefore, we employ a method which searches for these planar structures directly.

Image Division into Quadrilaterals. First, each image considered for 3D scene recovery is divided into a set of quadrilaterals. Currently this is achieved by dividing the image into a regular grid and using the grid cells as quadrilaterals (more sophisticated methods could be employed in the future). Through experimentation we found the cell size of 64 × 64 provided the best trade off between execution time and accuracy.

Quadrilateral Recovery. Each quadrilateral is recovered by determining a plane (defined by a normal vector and an offset value) which gives the minimum difference between the original image area and that defined by the area of the plane reprojected into an adjacent image. The intersection of rays between between focal point, image quadrilateral and plane of best fit then provide the 3D coordinates of the recovered plane. This process is summarized in Figure 3.

If the image coordinates of a quadrilateral are denoted iq_0, iq_1, iq_2 and iq_3, the corresponding projected plane coordinates for a plane with normal n and focal point offset value V are computed as:

$$pq_i = F + \left(\frac{n \cdot ((F + nV) - F)}{n \cdot (iq_i - F)} \right) (iq_i + F) + C_x, \ \forall i \in \{0, 1, 2, 3\} \tag{1}$$

where, F is the cartesian coordinates of the focal point and C_x is a vector storing the central x-coordinate of the recovered scene space. Using the ego-motion between images, $pq_{0..3}$ are updated with respect to a new image in which they are likely to appear:

$$pq'_i = R \times pq_i + T, \ \forall i \in \{0, 1, 2, 3\} \tag{2}$$

where, R and T are rotation and translation matrices describing the ego-motion between frames. Note, ego-motion was detected using a similar method to Goecke et al. [12]. Projecting each pq'_i into the image space of this second image then provides coordinates of the updated quadrilateral with respect to the plane described by n and V.

Even though the initial quadrilateral dividing strategy produces squares, the previous will work for any shape quadrilateral (even if $iq_{0..3}$ represents a square, $iq'_{0..3}$ may not). Since we wish to compare the image information within these two quadrilaterals using a sum-of-absolute differences measure, for ease they are transformed into squares using a texture mapping procedure. If the two square

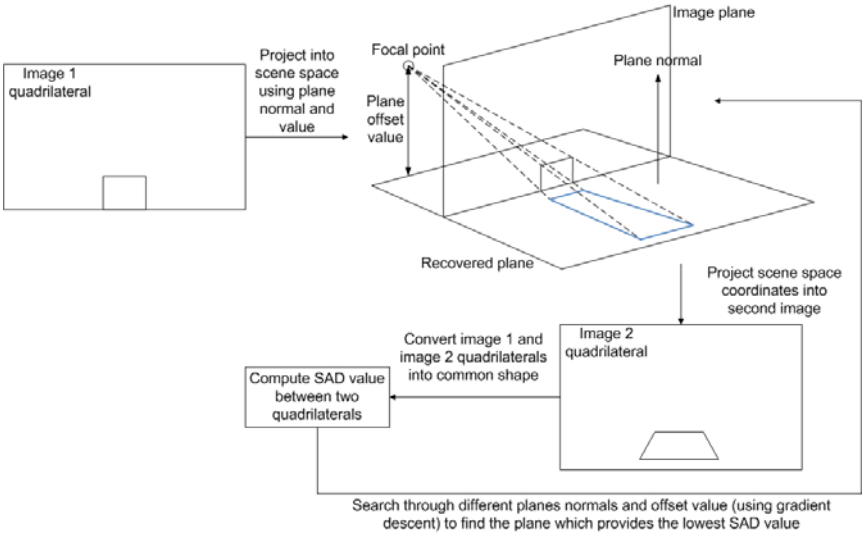


Fig. 3. 3D sequence recovery pipeline of the proposed dense, planar image quadrilateral recovery method

areas are denoted S_1 and S_2, they are then compared with a sum-of-absolute differences measure:

$$sad = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{3}|S_1(i,j,k) - S_2(i,j,k)|}{N^2} \tag{3}$$

where, N is the length of one side of the squares and k iterates over the red, green and blue colour channels of the images.

Given the previously described method, the task of recovering the 3D coordinates of an image quadrilateral is now finding the plane normal vector and focal point offset which correspond to the lowest sum-of-absolute difference value for projection into a second image. For a single normal vector, we applied gradient descent for searching the possible values.

2.2 Unsupervised Learning for Temporal Search Space Reduction

As previously mentioned, the intention of TSR is to link image features with 3D scene recovery results. This is achieved by storing multi-dimensional Gaussians representing similar, recurring features. Each of these *feature* Gaussians is associated with one or more one-dimensional Gaussian distributions representing different focal offset values computed as part of the previous planar recovery method, when recovering quadrilaterals with corresponding image features. Further, each of these *value* distributions has an associated plane normal (as per the previous method).

In terms of image features for the image area within a quadrilateral, separate red, green and blue channel histograms are computed. From each histogram the mean, standard deviation, skewness, kurtosis and energy are computed, providing 15 image features in total.

For any new quadrilateral processed, each of these features are computed and the probability these 15 features (f) belong to any of the feature distributions currently stored in the model is computed:

$$p_{FD_i} = \frac{1}{\sqrt{2\pi\sigma^2_{FD_i}}} e^{\frac{-(f-\mu_{FD_i})^2}{2\sigma^2_{FD_i}}}, \ \forall i \in FD \tag{4}$$

where, FD is the set of feature distributions in the model, μ_{FD_i} and σ_{FD_i} are the mean and standard deviation respectively of feature distribution i. If no p_{FD_i} is greater than a chosen threshold (in experiments we used 0.6), f represents a new image structure. The corresponding quadrilateral is recovered by traversing all plane normals and offset values as previously described. A new feature distribution (FD_{new}) is created such the $\mu_{FD_{new}} = f$ and $\sigma_{FD_{new}}$ is set in each dimension to 20% of the possible range for the corresponding feature values. FD_{new} is associated with a new value distribution (VD_{new}) representing the results of the recovery. Specifically, $\mu_{VD_{new}}$ is the offset value of the best fitting plane. This new feature, value distribution are then added to the model.

However, if any p_{FD_i} are greater than the threshold, each associated value distribution is considered in turn and recovery proceeds using the plane normal associated with the value distribution and the value range defined by:

$$min_{VD_{i,j}} = \mu_{VD_{i,j}} - (D\sigma_{VD_{i,j}} \times (1 - p_{FD_i})) \tag{5}$$

$$max_{VD_{i,j}} = \mu_{VD_{i,j}} + (D\sigma_{VD_{i,j}} \times (1 - p_{FD_i})) \tag{6}$$

where, $VD_{i,j}$ represents the value distribution associated with FD_i currently considered, $\mu_{VD_{i,j}}$ and $\sigma_{VD_{i,j}}$ are the mean and standard deviation of the 3D values associated with each value distribution ($VD_{i,j}$) associated with FD_i and D is a scalar value (chosen in experimentation to be 3). If the sum-of-absolute difference value corresponding to the best fitting plane computed from these ranges is greater than a threshold, the values chosen are assumed to of been inappropriate and the quadrilateral is reprocessed using all possible plane normals and offset ranges (this has been done to prevent convergence). The results of which are used to create a new value distribution (as before) which is associated with the feature distribution with the highest p_{FD_i} value. Further, the mean and standard deviation of this best fitting feature distribution are updated using f.

Finally, if the sum-of-absolute difference value corresponding to the best fitting plane is less than the chosen threshold, the feature and value distribution corresponding to the best match are updated accordingly.

3 Experimental Results

The data used for experimentation consists of High-Definition (i.e. 1920 × 1080 pixels) image frames, captured from a front-forward facing camera mounted on a train. In total, 5 sequences totalling 520 image frames were used. Due to the restrictions of the railway environment, each image frame had to be ground truthed by hand - matching features between image pairs and using these correspondences to reconstruct the true 3D position of the feature manually. Approximately, 850 features were ground truthed in this manner in each image. Even though this was done as accurately as possible, this ground truthing will not be completely accurate and so will provide a source of error when comparing with the recovered scene. However, the difference between recovered and ground truth scene can still be used as an indicator of the performance of each method relative to each other.

Tables 1 and 2 provide the results for number of plane offset values checked (i.e. size of the search space traversed) and accuracy (which is measured in metres difference from the ground truth) respectively for an exhaustive method not using TSR and the described TSR method. Note, the table column heading SX refers to the sequence number corresponding to the results. These results show that the incorporation of the image, scene relationship used for reducing feature correspondence search ranges greatly reduces the number of checks made (and hence computation) whilst unaffecting accuracy.

Table 1. Average number of focal offset values checked per quadrilateral

Method	**S1**	**S2**	**S3**	**S4**	**S5**	**Average**
Exhaustive	335.16	288.62	345.46	314.04	340.39	324.73
TSR	20.32	16.97	22.64	17.00	21.18	19.62

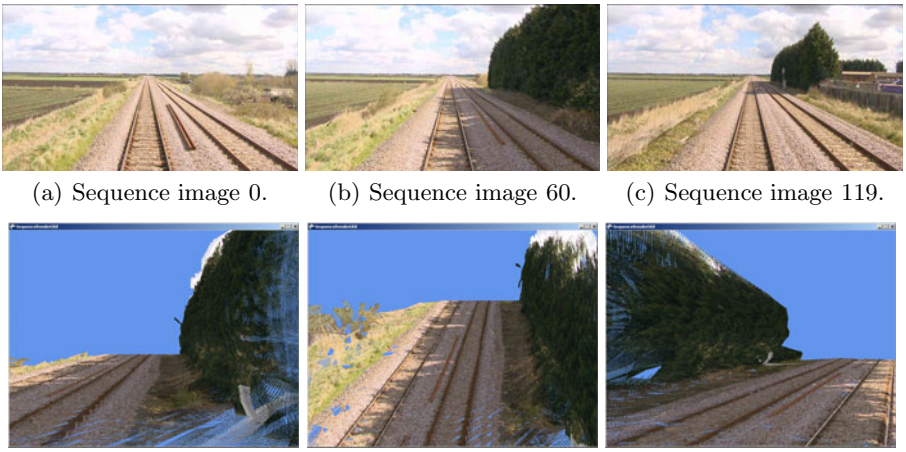
Table 2. Average accuracy of recovered scenes (per image)

Method	**S1**	**S2**	**S3**	**S4**	**S5**	**Average**
Exhaustive	0.88	1.40	1.80	0.77	1.62	1.13
TSR	0.55	0.88	1.53	0.64	1.36	0.99

Table 3 compares this TSR method with others presented in literature. Where possible, authors implementations have been used. For fairness of comparison, all other methods for which results are presented, recovered the same image areas as the TSR method. These results show that with the data used in our application, the presented method provides the most accurate results in the least amount of time. For completeness, Figure 4 shows typical reconstruction results using the TSR method from parts of image sequences considered.

Table 3. Comparison results of the proposed method with others from literature

Method	Time (seconds)	Difference (m)
TSR	1.29	0.99
SIFT features ([17, 28])	2.6	4.09
ENFT ([27])	8.91	3.92
PTAM ([14])	9.77	1.43
MonoSLAM ([5])	13.03	1.19
Locally planar patches ([21])	11.41	1.08
Calway features ([3])	15.19	1.03



(a) Sequence image 0. (b) Sequence image 60. (c) Sequence image 119.

(d) Example recovered scene screen shot 1. (e) Example recovered scene screen shot 2. (f) Example recovered scene screen shot 3.

Fig. 4. Example 3D recovered sequence 1

4 Conclusions

In this work, we have presented a method for 3D scene recovery which explicitly stores relationships between recurring image features and recovered 3D information to reduce search spaces traversed when computing feature correspondences. Scenes were recovered in a railway corridor mapping context and results showed that storing and using such relationships can dramatically decrease the computation required to recover a scene. Further, the proposed planar recovery method used in conjunction with this TSR method performs competitively with other methods presented in literature. For future work, we intend to investigate the benefits of TSR-based methods as applied to data sets other than the specific one described in this work. That is, we believe there are other data sets to which the previously described TSR framework could be applied, and investigation of this would prove the ability of the presented approach to generlise to other cases.

References

[1] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)

[2] Chekhlov, D., Gee, A.P., Calway, A., Mayol-Cuevas, W.: Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In: International Symposium on Mixed and Augmented Reality, ISMAR (November 2007)

[3] Chekhlov, D., Mayol-Cuevas, W.: Appearance based indexing for relocalisation in real-time visual slam. In: 19th British Machine Vision Conference, pp. 363–372 (2008)

[4] Davison, A.J.: Real-time simulataneous localization and mapping with a single camera. In: Proc. International Conference on Computer Vision, pp. 1403–1411 (October 2003)

[5] Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on PAtterns Analysis and Machine Intelligence 29, 1–15 (2007)

[6] Engels, C., Fraundorfer, F., Nistér, D.: Integration of tracked and recognized features for locally and globally robust structure from motion. In: VISAPP International Workshop on Robotic Perception, VISAPP RoboPerc (January 2008)

[7] Favaro, P., Jin, H., Soatto, S.: A semi-direct approach to structure from motion. The Visual Computer 19, 377–384 (2003)

[8] Fraundorfer, F., Schindler, K., Bischof, H.: Piecewise planar scene reconstruction from sparse correspondences. In: Image and Vision Computing, vol. 24, pp. 395–406 (2006)

[9] Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR, pp. 1422–1429. IEEE, Los Alamitos (2009)

[10] Gee, A.P., Chekhlov, D., Calway, A., Mayol-Cuevas, W.: Discovering higher level structure in visual slam. IEEE Transactions on Robotics 24, 980–990 (2008)

[11] Gee, A.P., Chekhlov, D., Mayol, W., Calway, A.: Discovering planes and collapsing the state space in visual slam. In: 18th British Machine Vision Conference (September 2007)

[12] Goecke, R., Asthana, A., Pettersson, N., Petersson, L.: Visual vehicle egomotion esitmation using the fourier-mellin transform. In: IEEE Intelligent Vehicles Symposium, pp. 450–455 (June 2007)

[13] Kanbara, M., Ukita, N., Kidode, M., Yokoya, N.: 3d scene reconstruction from reflection images in a sphereical mirror. In: The 18th International Conference on Pattern Recognition (ICPR 2006), pp. 874–879 (2006)

[14] Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Proceedings and the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reailty, ISMAR 2007 (2007)

[15] Li, P., Farin, D., Gunnewiek, R.K., de With, P.H.N.: On creating depth maps from monoscopic video using structure frm motion. In: 27th Symposium on Information Theory, pp. 508–515 (2006)

[16] Liu, G.-H., Feng, Q.-Y.: Recovering 3d shape and motion from image sequences using affine approximation. In: Second International Conference on Information and Computing Science, pp. 349–352 (2009)

[17] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)

[18] Martinez-Carranza, J., Calway, A.: Unifying planar and point mapping in monocular slam. In: British Machine Vision Conference, pp. 1–11 (September 2010)
[19] Martinez-Carranza, J., Calway, A.: Efficiently increasing map density in visual slam using planar features with adaptive measurements. In: British Machine Vision Conference, pp. 1–11 (September 2009)
[20] Metro Solutions (2006), http://metrosolutions.org/go/doc/1068/116948 (last accessed: July 25, 2010)
[21] Molton, N., Davidson, A., Reid, I.: Locally planar patch features for real-time structure from motion. In: Proc. British Machine Vision Conference, BMVC (September 2004)
[22] Ogale, A.S., Aloimonos, Y.: Shape and the stereo correspondence problem. International Journal of Computer Vision 65, 147–162 (December 2005)
[23] Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S.N., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. International Journal of Computer Vision, 143–167 (2008)
[24] Sinha, S.N., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. In: Twelth IEEE International Conference on Computer Vision, ICCV 2009 (2009)
[25] Tomasi, C., Kanade, T.: Shape and motion from image streams: a factorization method. Technical Report TR 92-1270, Carnegie Mellon (March 1992)
[26] Yun, S.U., Min, D., Sohn, K.: 3d scene reconstruction system with hand-held stereo cameras. In: 3DTV Conference, pp. 1–4 (2007)
[27] Zhang, G., Dong, Z., Jia, J., Wong, T.-T., Bao, H.: Efficient non-consecutive feature tracking for structure-from-motion. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 422–435. Springer, Heidelberg (2010)
[28] Zhang, G., Hua, W., Qin, X., Shao, Y., Bao, H.: Video stabilization based on a 3d perspective camera model. The Visual Computer 25, 997–1008 (2009)
[29] Zhang, G., Jia, J., Wong, T.-T., Bao, H.: Consistent depth maps recovery from a video sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 974–988 (2009)