LexisNexis® Academic

Print Request: Current Document: 4
Time Of Request: Friday, February 09, 2018 01:18:02 EST
Send To:

MEGADEAL, ACADEMIC UNIVERSE
MASTER'S COLLEGE
LIBRARY
SAN JOSE, CA 00000

Terms: (lidar)

Source: IET Computer Vision
Project ID:

4 of 4 DOCUMENTS

IET Computer Vision

October 2013

Fusion of dense spatial features and sparse temporal features for three-dimensional structure estimation in urban scenes

BYLINE: Mohamad MotasemNawafmohamad.motasem.nawaf@univ-st-etienne.fr; AlainTrémeau

SECTION: Pg. 302 - 310 Vol. 7 No. 5 1751-9632

LENGTH: 5941 words

1

Introduction

Estimating the three-dimensional (3D) structure of a scene from two-dimensional (2D) image stream is one of the most popular problems within computer vision. It is referred to as structure from motion (SfM) or 3D reconstruction from video sequence [2]. SfM has been applied in several applications [2] such as robot navigation, obstacle avoidance, entertainments, driver assistance, reverse engineering and modelling etc.

In our work, we focus on the problem of estimating the 3D structure from a video taken by a camera installed on a moving vehicle in urban environments. This setup leads possibly to create 3D maps of our world. However, the dominant forward motion of the camera from one side, and the texture-less scenes that are present generally in urban environment produce an erroneous depth recovery. The forward camera motion could result degenerated configurations for a naturally ill-posed problem, or mathematically, a large number of local minima during the minimisation of the reprojection error [3], that results in inaccurate camera relative motion estimation. Moreover, the limited lifetime of tracked feature points prevents using general optimisation methods such as in traditional SfM. Additionally, forward motion restricts features matching because of non-homogeneous scale changes of image objects, especially those aligned parallel to camera movement.

In the proposed method, we suggest to benefit from the monocular cues (e.g. spatial depth information) to improve 3D depth estimation. We believe that such spatial depth information is complementary to temporal information. For instance, given a blue patch located at the top of an image, an SfM technique will probably fail to compute the depth because of the difficult matching problem, whereas the monocular depth estimation method (supervised learning) will assign it a large depth value as it will be considered as a sky with high probability.

Similar to other works [4, 5], we consider that our world is made up of small planar patches, and the relationship between each two patches is either connected, planar or occluded. Based on these considerations, the goal is to estimate the plane parameters where each patch lies. These patches are obtained from the image using over-segmentation method [6] or what is called superpixels segmentation. In order to fuse both temporal and monocular depth information, and also to handle the interactive relationship between superpixels, we proposed to use a Markov random field (MRF) model similar to the one used in [4]. However, we extend the model by adding new terms to include temporal depth information computed using a modified SfM technique. Moreover, we benefit from the limited degrees of freedom (DoF) of camera motion (which is such of the vehicle) to improve relative motion estimation, and in return, the depth estimation.

Spatial depth information is obtained using an improved version of the method proposed in [4], which estimates the depth from a single image. The method employs an MRF model that is composed of two terms; one integrates a broad set of local and global features, whereas the other handles the neighbouring relationship between superpixels based on occlusion boundaries. In our method, we compute occlusion boundaries from motion [7] to obtain more reliable results than using a single image as in the aforementioned method. Therefore it is expected to have better reconstruction, even before integrating the temporal depth information.

To perform SfM, which represents temporal depth information, we use optical flow based technique that allows forcing some constraints on camera motion (which has limited DoF). Moreover, it is proved to have better depth estimation for small baseline

distances and forward camera motion [6]. Here, we compute a sparse optical flow using an improved method of Lucas-Kanade with multiresolution and subpixel accuracy. Then, results are refined thanks to camera motion estimation. Based on the famous optical flow equation [6], we obtain the depth for a set of points in the image. Hence, we can add some constraints on the position of scene patches to whom these points belong.

The remainder of this paper is organised as follows. In Section 2, we give an overview of various methods for depth estimation using; video sequences, single image and then combined spatiotemporal methods. In Section 3, we introduce the MRF model that integrates SfM with the monocular depth estimation, and we explain its potential functions, parameters learning and inference. Section 4 presents our experiments and results of evaluating our method. Finally, in Section 5, we conclude our work and we discuss the advantages of the proposed method. This work is an expanded version of a study [1] published at the 2.5D Sensing Technologies in Motion workshop (QU3ST) in the European Conference on Computer Vision, Florence, 2012 (ECCV'12).

2

Related works

In computer vision, SfM has taken a great attention by researchers, it is considered as one of the well-studied problems. However, most of the efforts are focused on a certain number of aspects. For instance, improving feature points matching [8], formalise better constraints to improve relative camera pose estimation [9, 10], robust methods for outliers rejection [11], linear/non-linear reprojection error optimisation and bundle adjustment [10], formalise a set of constraints on more than two frames [9]. Most of these contributions do only consider temporal information that results from image stream variation with respect to time, without trying to analyse the monocular depth cues that are present in every single image.

From another side, several monocular cues that exist in a single image have been exploited by researchers, that includes: vanishing points and horizon line [9], shades, shadows, haze, patterns and structure [12]. Unfortunately, most of these cues are not present in all kinds of images, and they require specific settings. In contrast, we are looking to provide a general spatial depth estimation approach to be integrated with temporal depth estimation as mentioned earlier. Hence, we target a new generation (since past decade) of methods that perform 2D to 3D conversion using a single image. Generally, these methods have no constrains and are based on the use of exhaustive feature extraction and probabilistic models to learn depth. An early approach attempts to estimate general depth of an image is proposed in [13], which employs Fourier spectrum to compute a global spectral signature of a scene to estimate the average depth of the image scene. Later on, an innovative attempt to perform 3D reconstruction from one image is proposed in [14]. Where first the image is over-segmented into superpixels, then each superpixel is classified as sky, vertical (objects) or planar (ground). It employs a wide set of colour, texture, location, shape and edge features for training. Finally, the vertical region is 'cut and folded' in order to create a rough 3D model. Although this method has been improved later by considering some geometric subclasses (centre, left, right etc.) in [15], the 'ground-vertical' world assumption does not apply for wide range of images. A similar concept is proposed in [16], which is extended to motion classes such as 'right headed, left headed, oncoming and static background'. Such a medium-level representation of 3D scenes named 'stixel world' allows the extraction of multiple objects in complex inner city scenarios, including pedestrian recognition and detection of partially hidden moving objects. The best class assignment and the dependencies between neighbouring stixel labels are dynamically defined from prior knowledge about the current local 3D environment and temporal information using a conditional MRF. More general method is proposed in [5], which estimates the depth from a single image based on some predicted semantic labels (sky, tree, road etc.) using multi-class pixel-wise image labelling model. Then, the computed labels guide the 3D estimation by establishing a possible order and positioning of image objects. In [17], a convolutional neural network based method is proposed to learn features from noisy labels to recover the 3D scene layout of a road image. It combines colour planes to provide a statistical description of road or side-walk areas (i.e. horizontal ground) that exhibit maximal texture uniformity. In [18], 11 object classes (road, building, sky, tree, side-walk, car etc.) are used for labelling. Motion and structure features (height above the camera, distance to the camera path, projected surface orientation, feature track density and residual reconstruction error, inferred from 3D point clouds) and appearance features (textons, colour, location and histogram of Gaussian (HOG) descriptors) are combined, thanks to a conditional random fields model. Another general approach has been proposed in [4], which does not have initial assumption about scene's structure. It proceeds by over-segmenting the image similar to [14]. The absolute depth of each image patch is estimated based on learning an MRF model, where a variety of features that capture local and contextual information is employed. As an extension, the authors proposed a model to create 3D reconstruction from sparse views. We see later how a part of our work is inspired by this method. The contribution is that we adapted (and trained) our method to road scenes and forward motion. The added constraints improved relative motion. Also, the optical flow based SfM provides approximate but denser feature points as an alternative to points triangulation, since the later tends to fail near image plane axes. Another improvement in the current work is that we compute occlusion boundaries based on the motion between two frames, which is more robust and accurate than a multi-segmentation based approach using a single image.

In parallel with 3D structure estimation from road scenes, there is also an increased interest in trajectory estimation. For instance, the method proposed in [19] uses higher level structure to create structure driven temporal map to help in visual odometry computation. It employs planar features and plane fitting, however, the in-plane colour consistency is not been taken into account. So the computed planes are not meant to correspond to objects surfaces. In our method, we also perform plane fitting but implicitly. Whereas the in-

plane colour consistency is achieved through superpixel segmentation approach. In the same context, the authors in [20] proposed a monocular visual odometry scheme to recover camera relative motion. We proceed similarly in our method by tracking feature points over frames, then based on a shifting three frames window, we compute a frame-to-frame translation scale using the method proposed in [21].

In the context of combining both spatial and temporal depth information (as we aim), a method that combines SfM with a simultaneous segmentation and object recognition is proposed in [18], it targets road scene understanding. The task is achieved through a conditional random field model, which consists of pixel-wise potential functions that incorporate motion and appearance features. The author claims that it overcomes the effect of small baseline variations. In our method, we perform direct depth estimation rather than object recognition. However, similar to [18], our method is also supervised learning oriented, we benefit from computed features to capture contextual information and learn depth. In comparison with our approach, we use small planar patches to model the world rather than the pixel-wise approach used in [18], as we think they better describe the world around us. This idea is also supported by the experimental results in [22]. The method in [23] proposes to combine sparse reconstruction using SfM with a surface reconstruction using MRF optimisation. The main difference between this approach and ours is that in this paper they do not use the superpixels segmentation to model the depth of objects, but a 2D tetrahedral mesh segmentation to fit objects surfaces. In our study, we use superpixels as we think they preserve more neighbouring relationships between uniform 2D surfaces and temporal consistency. Another approach with the same context is a semantic SfM approach [24], which is based on a probabilistic model. The proposed model incorporates object recognition with 3D pose and location estimation tasks. Also it involves potential functions that represent the interaction between objects, points and regions. Another approach that combines both spatial and temporal information is proposed in [25]. A stereo-matching algorithm with ground plane and temporal smoothness constraints is proposed for vehicle control and surveillance applications. In this paper, the authors exploit the geometry of the scene (the road plane geometry) and a vertical damping scale in order to enforce temporal consistency. This method enables to relax smoothness of disparity maps along vertical axis and to prevent disparity resolution loss because of lack of texture or occlusions because of motion. Spatial and temporal information are aggregated via permeability filter and guided filter. Compared with other aggregation methods, this approach does not exploit contextual information nor the intrinsic complementarity of spatial and temporal information.

3

Spatiotemporal depth fusion framework

In this section, we first introduce some notations. Then, we explain how we compute spatial and temporal depth features. After that, we discuss estimating occlusion boundaries that play an important role in our model. Next, we introduce our proposed framework as an MRF model that incorporates several terms related to spatial and temporal depth features. Finally, we show how we estimate the parameters from a given dataset and perform the inference for a new input.

3.1

Image representation

As mentioned earlier, we assume that the world is composed of planar patches, and the obtained superpixels are their 'one-to-many' 2D projection. This assumption represents a good estimate if the number of computed superpixels is large enough. We obtain superpixels from an image by using an over-segmentation algorithm [6]. We represent the image as a set of superpixels $S^t = \{S_1^t, S_2^t, \ldots, S_n^t\}$, where S_i^t defines superpixel i at time t. We define $\alpha_i^t \in \mathbb{R}^3$, the plane parameters associated to S_i^t such that for a given point $x \in \mathbb{R}^3$ on the plane satisfies $\alpha_i^t x = 1$. Our aim is to find the plane parameters for all superpixels in the image stream. Fig. 2*b* shows an example of the obtained superpixels.

3.2

Spatial depth features

Spatial features for supervised depth estimation have not achieved much success compared with other computer vision domains, such as object recognition and classification. Although the problem of monocular vision had been well studied in human vision (even before computers appear) and many monocular depth cues that human uses have been identified; however, it was not possible to obtain explicit depth representative measurements such as in stereo vision. Recently, there were several attempts to infer image 3D structure using spatial features and supervised learning [4, 5, 18]. In our method, we proceed in the similar way, in order to capture texture information, the input image is filtered with a set of texture energies and gradient detectors ([#x223c]20 filters) [22]. Then, by using superpixel segmentation image as a mask, we compute the filter response for each superpixel by summing its pixels in the filtered image. We refer the reader to [22] for more details. In order to capture general information, the aforementioned step is repeated for multiple scales of the image. Also, to add contextual information, for example, texture variations, each superpixel feature vector includes the features of its neighbouring superpixels. Additionally, the formed feature vector includes colour, location and shape features as they provide representative depth source for fixed camera configuration and urban environment. For instance, recognising the sky and the ground. These features are computed as shown in Table 1 in [14]. We denote
$\boldsymbol{X}_i^t$
as the feature vector for superpixel
S_i^t
.

Table 1 Experimental results of spatial (SIE), temporal (SfM) and combined methods

	Error ratio	STD
SIE/SfM	1.82	0.43
SfM/combined	2.24	0.21
SIE/combined	2.81	0.38

3.3

Temporal depth features

In this subsection, we first describe some mathematical foundations and camera model. Then, we explain how to perform sparse depth estimation, which will be integrated in the probabilistic model given in Section 3.5.

We use a monocular camera mounted on a moving vehicle. We assume that the *Z*-axis of the camera coincides with the forward motion of the vehicle as shown in Fig. 1. Based on pin-hole camera model and camera coordinate system, a given 3D point *M*(*X*, *Y*, *Z*) is projected onto the 2D image as *m*(*x*, *y*) by a perspective projection (1)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}$$

When the vehicle moves, which is equivalent to fixed camera and moving world, the relationship between the velocity of a 3D point
$[\dot{X}\dot{Y}\dot{Z}]^{\mathrm{T}}$
and the velocity of its 2D projection
$[\dot{x}\dot{y}]^{\mathrm{T}}$
is given as the time derivative of (1). Then, based on the well known optical flow equation
$\dot{\boldsymbol{M}} = -\boldsymbol{T} - \Omega \times \boldsymbol{M}$
, and assuming a rigid scene, the 3D velocity is decomposed into translational *T* and rotational [#x3a9] velocities [2]. Hence, we obtain (2), which is the essence of most optical flow based SfM methods (2)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} xy/f & -f - (x^2/f) & -y \\ f + (y^2/f) & -xy/f & x \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$$

Based on this equation, we proceed in computing a sparse depth. We estimate the relative camera motion between two adjacent frames by first performing scale invariant feature transform (SIFT) feature points matching [8]. Next, we estimate the fundamental matrix using random sample consensus (RANSAC) [11]. Then, given camera intrinsic parameters, we can obtain the essential matrix that encodes the rotation and translation (which is up to scale) between the two scenes. This represents also the relative camera motion parameters [*T*[#x3a9]]. To reveal the scale ambiguity, we employ a reprojection-based method proposed in [21]. We track feature points over frames, then by using a shifting 3 frames window, we compute a frame-to-frame translation scale by projecting the trackable points on a reference frame after introducing a scale factor between two frames. The scale factor is then computed by

minimising a least square set of equations using singular value decomposition (SVD). Hence, we compute a correct frame-to-frame scale for the sequence of images. However, having first frame scale set to [*I*|0], we have an overall unknown scale. In our case, given that we are dealing with fixed configuration, we could set this scale using metric measures.

Fig. 1 Acquiring geometry: camera installed on a moving vehicle with Z-axis coincides with forward motion direction

The left hand side of (2) is basically the optical flow computed between two frames. In our implementation it is obtained using the well known Lucas-Kanade with multi resolution and subpixel accuracy. Moreover, we benefit from the estimated fundamental matrix to reject outliers in the optical flow. At this point, we could compute an approximate depth for the selected feature points. Specifically, we set a threshold for the difference between x and y disparities. In case of large difference (which means the pixel is close to image axes, but far from the centre), we compute the depth using only the larger component. We think this is an advantage over traditional 3D triangulation method, where both x and y are treated equally. However, this additional step is applied only when we spot dominant forward motion, in which our assumption is only true.

Besides, given the specific camera setup as shown in Fig. 1, the motion of the camera is not totally free in the 3D space (motion of a vehicle). Therefore we could add some constraints that express the feasible relative camera motion between two frames. For instance, limitation in T_y and [#x3a9]$_z$ velocities. However, because of the absence of essential physical quantities, precise constraints on camera (or vehicle) motion could not be established theoretically. Instead, we evaluate experimentally possible camera motion estimated from a set of video sequences acquired in different scenarios. As a result, we could establish some roles to spot outliers in the newly computed values for relative camera motion [*T*[#x3a9]]. This way we improve the relative camera motion estimation in our case as we regularly have degenerated configurations (because of small baseline variations and dominant forward motion as mentioned earlier).

3.4

Occlusion boundaries estimation

When the camera translates, close objects move faster than far objects, and hence this causes to change the visibility of some objects in the scene. Although this phenomenon is considered as a problem in computer vision, it provides an important source of information about 3D scene structure. In our approach, we benefit from motion to infer occlusion boundaries. We use the method proposed in [7] to generate a soft occlusion boundaries map from two consecutive image frames. The method is based on supervised training of an occlusion detector, thanks to a set of visual features selected by a random forest based model. Since occlusion boundaries lie close to surfaces edges, we use the classifier output as an indicator to the relationship between two superpixels, if they are connected or occluded. Hence, we add a penalty term in our MRF that forces the connectivity between superpixels. This term is inversely proportional to the obtained occlusion indicator. Fig. 2*c* shows occlusion surfaces, where pixels follow common motion, whereas Fig. 2*d* shows the estimated occlusion boundaries map.

Fig. 2 Scene representation and corresponding occlusion boundaries
a Original image
b Superpixels segmentation
c Occlusion surfaces
d Estimated occlusion boundaries map

3.5

MRF for depth fusion

MRF is becoming increasingly popular for modelling 3D world structure because of its flexibility in terms of adding appearance constraints and contextual information. In our problem, we formulate our depth fusion as an MRF model that incorporates certain constraints with variable weights so they are jointly respected. Furthermore, we preserve the convexity of our problem such as in [4] to allow solving it through a linear program rather than probabilistic approaches for less computation time. We have seen earlier how to obtain temporal depth information, monocular features and occlusion boundaries. Fig. 3 shows a simplified process flow for the proposed framework. We formulate our energy function, which includes all of these terms as (3)

$$E\left(\mathbf{a}^t|\boldsymbol{X}^t,\ \boldsymbol{O},\ \hat{\boldsymbol{D}},\ \mathbf{a}^{t-1};\quad \theta\right)=\underbrace{\sum_i \psi_i\left(\alpha_i^t\right)}_{\substack{\text{spatial depth}\\ \text{term}}}$$
$$+\underbrace{\sum_{ij}\psi_{ij}\left(\alpha_i^t,\ \alpha_j^t\right)}_{\substack{\text{connectivity}\\ \text{term}}}+\underbrace{\sum_{ik}\phi_{ik}\left(\alpha_i^t,\ \hat{d}_k^i\right)}_{\substack{\text{temporal depth}\\ \text{term}}}+\underbrace{\sum_i \phi_i\left(\alpha_i^t,\ \alpha_i^{t-1}\right)}_{\substack{\text{time}-\text{consistency}\\ \text{term}}}$$

where the superscripts *t* and *t* - 1 refer to current and previous frames. *X* is the set of superpixels feature vectors. *O* is a map of occlusion boundaries computed between the frames *t* and *t* - 1. The estimated sparse depth is $\hat{\boldsymbol{D}}$, whereas $\hat{d}_k^i$ is the estimated depth value for pixel *k* in superpixel *i*. *[#x3b1]*$_i$ is superpixel *i* plane's parameters and a is the set of parameters for all superpixels. *[#x3b8]* are the learned monocular depth parameters. We now proceed in describing each term in our model (in the first three terms, we will drop down the superscript of frame indicator *t* as they are the same).

Fig. 3 Graphical representation of our MRF
Occlusion boundaries and sparse SfM are estimated between two frames t and t + 1, whereas monocular depth features are extracted from the current frame t; the MRF model integrates this information in order to produce a joint result for 3D structure estimation

3.5.1

Spatial depth term

This term is responsible for penalising the difference between the computed plane parameters and the ones estimated from spatial depth features (based on the learned parameters *[#x3b8]*). It is given by the accumulated error for all pixels in the superpixel [22]. For simplification, let us define a function $\delta\left(d_k^i,\ \hat{d}_k^i\right)$ that represents one point fractional depth error between an estimated value $\hat{d}_j^i$ and actual value d_j^i given plane parameters *[#x3b1]*$_i$. This potential function is given as (4)

$$\psi_i\left(\alpha_i\right)=\beta_1\sum_j v_k^i\delta\left(d_k^i,\ \hat{d}_k^i\right)$$

where v_k^i is a learned parameter that indicates the reliability of a feature vector $\boldsymbol{X}_k^i$ in estimating the depth for a given point p_k^i. *[#x3b2]*$_1$ is a weighting constant.

3.5.2

Connectivity prior

This term is based on the map of occlusion boundaries explained earlier. For each two adjacent superpixels, we compute an occlusion boundary indicator by summing up all pixels located at the common border in the estimated map. The obtained occlusion indicators are

normalised, so that they are in the range [0-1]. We refer o_{ij} for the indicator between superpixels *i* and *j*. The potential function is computed for each two neighbouring superpixels by choosing two adjacent pixels from each. The function penalises the difference in distance between each of them to the camera. We have (5)

$$\psi_{ij}\left(\alpha_i,\ \alpha_j\right) = \beta_2 o_{ij} \sum_{k=l=1}^{2} \delta\left(d_k^i,\ d_l^j\right)$$

where *[#x3b2]*$_2$ is a weighting constant. This potential function forces neighbouring superpixels to be connected only if they are not occluded with the help of occlusion indicator o_{ij}. In comparison with the original method [4], we drop down the co-planarity constraint as we believe that the included temporal information and estimating occlusion boundaries indicator for motion provide an important source of depth information about plane orientation. Therefore we do not mislead the estimation procedure with such approximation.

3.5.3

Temporal depth term

This term enforces some constraints that are established from the set of points where the depth is known. It is evident that with three non-collinear points, we can obtain plane parameters *[#x3b1]*$_i$. However, to consider less or more number of points, we formulate this potential function to penalise the error between the estimated depth

$\hat{d}_k^i$

for a point

$p_k^i \in S_i$

and the computed depth given plane parameters *[#x3b1]*$_i$. Fig. 4 shows how this error is computed. Hence, we have (6)

$$\phi_{ik}\left(\alpha_i,\ \hat{d}_k^i\right) = \beta_3 |\hat{d}_k^i - 1/\alpha_i^{\mathrm{T}} \boldsymbol{r}_k^i|$$

where

$\boldsymbol{r}_k^i$

is a unit vector that points from camera centre to the point

p_k^i

. *[#x3b2]*$_3$ is a weighting constant. We compute absolute depth error rather than fractional error, since SfM is more confident than spatial depth estimation.

Fig. 4 Illustration for how to compute the error in depth between the estimated value and depth for a given [#x3b1]i

3.5.4

Time-consistency term

In case of more than two frames, the quality of the 3D structure estimation varies from one frame to another, and it depends highly on the relative camera motion components (larger T_x and T_y translational motions result in better 3D structure estimation). Therefore we add some penalty in order to guide depth estimation at time *t* given the estimation at time *t* - 1. This smoothens the overall estimated structure variations in time. Hence, for each superpixel

S_i^{t-1}

, we find its correspondence

S_i^t

based on the motion parameters and the size of common area. Additionally, we consider some visual features, such as colour and texture. Eventually, some superpixels will not have correspondence because of changing the field of view. We select the point

p_k^i

at the centre of the

S_i^{t-1}

and we form a ray from camera centre to this point. This ray intersects with superpixel

S_i^t

at point

$p_k^{i\prime}$

. The formulated potential function penalises the distance across the ray between the two points (7)

$$\phi_i\left(\alpha_i^t,\ \alpha_i^{t-1}\right) = \beta_4 \delta\left(d_k^{i\prime},\ \hat{d}_k^i\right)$$

Here *[#x3b2]*$_4$ is the smoothness term. We intend to only use one point to leave some freedom in-plane orientation and for better 3D reconstruction refinement.

3.6

Parameters learning and inference

In our MRF formulation, we preserve the convexity as all terms are linear or L_1 norm, which is solved using linear programming. To learn the parameters, we first proceed with the first two terms of (3). We assume unity value for parameters *[#x3b2]*$_1$ and *[#x3b2]*$_2$. The two parameters *[#x3b8]* and *[#x3bd]* are learned individually [22] using a dataset with ground truth. For the rest of the parameters, *[#x3b2]*$_1$ and *[#x3b2]*$_2$ defines how the method is spatially oriented, whereas large *[#x3b2]*$_3$ turns the method into conventional SfM. *[#x3b2]*$_4$ allows previous estimation to influence the current one. Hence, the weighting constants *[#x3b2]*$_{1,\ldots,4}$ depends on the context, although they could be learned through cross validation.

4

Experimental results

It exists few datasets and benchmarks to evaluate 3D structure estimation methods from image sequences [26-28]. In our experimental evaluation, we use the 'KITTI vision benchmark suite' [29], which comprises various sets of image sequences taken from a moving vehicle as in the assumed setup [raw data section, sequences # 0001-0013, 0048, 0056, 0059, 0091-0106]. As our aim is mainly to evaluate the proposed fusion scheme, we performed a 3D reconstruction on the given dataset using the three methods; 3D structure estimation from single image (SIE), the optical flow based SfM explained in Section 3.3, and finally the proposed combined method. Thanks to the provided laser scanner data, we could compute the error for each case as direct differences between the estimation and ground truth. To be turned into representative measures, we computed the ratio between the error for each of the two baseline and the combined methods; here, we convert the sparse SfM to dense by using weighted average to compute the depth for an intermediate point (unlike the results shown in Table 2, where we consider sparse SfM). Table 1 shows the error ratios between these methods averaged over the used image sequences, this way we can evaluate the performance of the combined method with respect to spatial or temporal component. The table also shows the standard deviation associated with each ratio, which gives an idea about the stability of the results for different scenarios. From these results, we could conclude that both spatial and temporal depth estimations are partially complementary as the combined method has better performance than each individual method. Fig. 5 shows an example of results obtained using each method, the triangulations shown in Fig. 5*d* helps in computing dense estimation.

Fig. 5 An example of depth estimation results
a Depth estimation from single image
b Depth estimation using SfM technique
c Estimated depth using the combined method, each three points define a plane
d Triangulations associated with the depth estimation shown in c

Table 2 Relative error distribution as a function of depth

Depth range, m	0?10	10?20	20?30	30?40	40?50	50?80	All
SIE (dense)	23.52	29.05	29.44	23.21	22.02	24.74	26.41
SfM (sparse)	11.82	9.06	14.92	18.81	23.69	40.69	16.65
SfM (sparse, FM)	5.10	4.82	8.64	8.87	13.48	30.15	9.14
combined (dense)	14.94	16.86	17.88	22.22	21.97	28.37	18.65

From another side, we evaluate the error distribution for each of the three methods as a function of depth. Table 2 shows the relative error

$$|(\hat{d}/d) - 1|$$

between the estimated depth

$$\hat{d}$$

and laser scanner (we set the maximum distance to 80 m, which is the limit of the used laser scanner) measure *d*. In the case of dense depth, we only compute the error for the points, where laser scanner data is available. While in the case of sparse depth, we look for the nearest neighbouring point within a small distance, if no such point exists, the estimated point is not considered in the computation.

The second column in Table 2 is for results of SFM points, whereas the third column is only for points used to compute the fundamental matrix (100-150 points in average). As expected, the sparse SfM points tend to be more accurate for close distances, whereas the large error for distances larger than 50 m ensures the fact that SfM is blind for large distances. The depth estimation from single image shows similar depth error for all depth ranges. Although the combined method gains an improvement over SIE over all ranges. Another remark that we note here is the improvement in the combined method with the large depth range w.r.t SfM (sparse case), which is [#x223c]12%. In total, although the error in SfM is slightly smaller than the combined method; as a return, the combined method provides a dense depth estimation.

We found it interesting to study the effect of the number of matching points in SfM on the final relative error of the combined method. Fig. 6 shows results obtained for 180 matching frames. Each couple of matching frames is associated with one point that relates the number of matching points (a) or the number of inliers used to compute the FM using RANSAC (b) against the relative error in the combined method. It is clear that in both figures there is an improvement in the results when we have more matching points, which is a more reliable depth source.

Fig. 6 Estimated depth relative error $\vert \lpar \hat d/d\rpar - 1\vert $|(d^/d)-1| against
a Number of matching feature points (frame-to-frame)
b Number of inliers feature points used to compute the FM using RANSAC

We also evaluate the robustness of trajectory estimation and compare its accuracy to the ground truth that is provided by an inertial navigation system (GPS/IMU). Fig. 7 shows two examples (0009 and 0095) of the computed trajectory and the provided ground truth superimposed onto a Google Earth maps. The estimated trajectories gave an average translation error of 6.8% and rotation error 0.0187 °\m. Compared with non-constrained trajectory estimation, we had an overall average improvement of translation error by 0.9% and rotation error by 3%. As expected, this improvement mainly applies to the *y*-direction (vertical) while it is equal for rotations.

Fig. 7 Estimated trajectory versus ground truth obtained from inertial navigation system (GPS/IMU) superimposed onto a Google Earth image of KITTI dataset sequences
a 0009
b 0095

Our implementation requires 85 s to perform the 3D estimation for three frames on a multi-core Linux PC (Intel I7, 8 GB random access memory (RAM)). Most of this time is allocated for the intensive spatial features extraction, whereas the feature points extraction and matching runs in parallel. For longer sequences, the time increases linearly, since our method performs local refinement.

5

Discussion and conclusions

We have presented a novel framework to perform 3D structure estimation from an image sequence, which combines both spatial and temporal depths information to provide more reliable reconstruction. Temporal depth features are obtained using a sparse optical flow based SfM technique. The spatial depth features are obtained through a broad global and local feature extraction phase that tries to capture monocular depth cues. Both depth features are fused in an MRF model to be solved jointly. The experiments show that the joint method overcomes the performance of the estimation from the single image. Also, it provides a dense depth estimation, which is an advantage over SfM. By analysing the depth estimation relative error w.r.t depth range, we conclude that both used depth features are complementary to each other. Monocular depth features are independent from depth range, and SfM is blind for large distances. We also conclude that the joint method provides better performance than computing dense depth map, using sparse SfM without taking colour consistency into account.

Although it is not our primary objective, trajectory estimation proved to be robust and accurate after introducing the constraints which are adapted to vehicle motion. Based on the results published in KITTI visual odometry benchmark [29], the proposed framework provides odometry estimation that is close to stereo-based visual odometry methods.

Our future work will be devoted to better estimate depth from spatial information by exploiting geometric monocular depth cues, such as vanishing points, horizon lines, vertical and horizontal assumptions etc. Another interesting direction for future research would be to exploit initial assumption about the scene's structure by assigning semantic labels to image areas, such as sky and road.

LOAD-DATE: June 19, 2014

LANGUAGE: ENGLISH

BIBLIOGRAPHY:

REFERENCES

1 Nawaf M.M., Trémeau A.: 'Joint spatio-temporal depth features fusion framework for 3d structure estimation in urban environment'. Workshops and Demonstrations Computer Vision, (ECCV 2012), 2012, pp. 526-535

2 Aanæs H.: 'Methods for structure from motion'. PhD thesis, Danmarks Tekniske Universitet, 2003

3 Vedaldi A., Guidi G., Soatto S.: 'Moving forward in structure from motion'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'07), 2007, pp. 1-7

4 Saxena A., Sun M., Ng A.: 'Make3d: Learning 3d scene structure from a single still 439 image', IEEE Trans. Pattern 440 Anal. Mach. Intell., 2009, 31, (5), pp. 824-840 (doi: 10.1109/TPAMI.2008.132)

5 Liu B., Gould S., Koller D.: 'Single image depth estimation from predicted semantic labels'. CVPR, 442 IEEE Conf., 2010, pp. 1253-1260

6 Felzenszwalb P., Huttenlocher D.: 'Efficient graph-based image segmentation', Int. J. 444 Comput. Vis., 2004, 59, (2), pp. 167-181 (doi: 10.1023/B:VISI.0000022288.19776.77)

7 Humayun A., Mac Aodha O., Brostow G.: 'Learning to find occlusion regions'. IEEE Conf. CVPR'11, 2011, pp. 2161-2168

8 Lowe D.: 'Distinctive image features from scale-invariant keypoints', Int. J. Comput. Vis., 2004, 60, (2), pp. 91-110 (doi: 10.1023/B:VISI.0000029664.99615.94)

9 Hartley R.I., Zisserman A.: 'Multiple view geometry in computer vision' (Cambridge University Press, 2004, 2nd edn.)

10 Triggs B., McLauchlan P., Hartley R., Fitzgibbon A.: 'Bundle adjustment: a modern synthesis', Vis. Algorithms: Theory Pract., 2000, 1883, pp. 153-177

11 Raguram R., Frahm J., Pollefeys M.: 'A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus', Comput. Vis. - ECCV, 2008, pp. 500-513

12 Lindeberg T., Garding J.: 'Shape from texture from a multi-scale perspective'. Proc. Fourth Int. Conf. on Computer Vision, 1993, 5303, pp. 683-691

13 Torralba A., Oliva A.: 'Depth estimation from image structure', IEEE Trans. Pattern Anal. Mach. Intell., 2002, 24, (9), pp. 1226-1238 (doi: 10.1109/TPAMI.2002.1033214)

14 Hoiem D., Efros A., Hebert M.: 'Automatic photo pop-up', ACM Trans. Graph., 2005, 24, (3), pp. 577-584 (doi: 10.1145/1073204.1073232)

15 Hoiem D., Efros A., Hebert M.: 'Recovering surface layout from an image', Int. J. Comput. Vis., 2007, 75, (1), pp. 151-172 (doi: 10.1007/s11263-006-0031-y)

16 Pfeiffer D., Erbs F., Franke U.: 'Pixels, stixels, and objects'. Workshops and Demonstrations Computer Vision, (ECCV 2012), 2012, pp. 1-10

17 Alvarez J., Gevers T., LeCun Y., Lopez A.: 'Road scene segmentation from a single image ECCV 2012, Part VII', 2012 (LNCS, 7578), pp. 376-389

18 Sturgess P., Alahari K., Ladick L., Torr P.: 'Combining appearance and structure from motion features for road scene understanding'. Proc. British Machine Vision Conf. (BMVC'09), 2009, pp. 1226-1238

19 Martinez-Carranza J., Calway A.: 'Efficient visual odometry using a structure-driven temporal map'. Proc. 2012 IEEE Int. Conf. on Robotics and Automation (ICRA), 2012, pp. 5210-5215

20 Nistér D., Naroditsky O., Bergen J.: 'Visual odometry'. Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2004 (CVPR 2004), 2004, vol. 1, pp. I-652

21 Esteban I., Dorst L., Dijk J.: 'Closed form solution for the scale ambiguity problem in monocular visual odometry'. Proc. Third Int. Conf. on Intelligent robotics and applications - Volume Part I. (ICIRA'10), Berlin, Heidelberg, 2010, pp. 665-679, ISBN 3-642-16583-4, 479 978-3-642-16583-2

22 Saxena A.: 'Monocular depth perception and robotic grasping of novel objects'. PhD thesis, Stanford University, 2009

23 Li P., Gunnewiek R.K.: 'Scene reconstruction using MRF optimization with image content adaptive energy functions'. Proc. 10th Int. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS'08), 2008, pp. 872-882

24 Bao S., Savarese S.: 'Semantic structure from motion'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 2025-2032

25 Cigla C., Alatan A.: 'An improved stereo matching algorithm with ground plane and temporal smoothness constraints'. ECCV 2012 Ws/Demos, Part II, 2012 (LNCS 7584), pp. 134-147

26 Hirschmuller H., Scharstein D.: 'Evaluation of cost functions for stereo matching'. IEEE Conf. on Computer Vision and Pattern Recognition, 2007 (CVPR'07), 2007, pp. 1-8

27 Meister S., Jähne B., Kondermann D.: 'Outdoor stereo camera system for the generation of real-world benchmark data sets', Opt. Eng., 2012, 51, (02), pp. 021107 (doi: 10.1117/1.OE.51.2.021107)

28 Pandey G., McBride J.R., Eustice R.M.: 'Ford campus vision and **lidar** data set', Int. J. Robot. Res., 2011, 30, (13), pp. 1543-1552 (doi: 10.1177/0278364911400640)

29 Geiger A., Lenz P., Urtasun R.: 'Are we ready for autonomous driving? the kitti vision benchmark suite'. Computer Vision and Pattern Recognition (CVPR), Providence, USA, 2012

PUBLICATION-TYPE: Magazine

---- End of Request ----
Print Request: Current Document: 4
Time Of Request: Friday, February 09, 2018 01:18:02 EST