Low-Complexity and Low-Delay Structure from Motion Approach for Advanced Driver Assist Systems

J. Andrade¹, C. Prakash¹, F. Akhbari², and L. Karam¹
¹Arizona State University, Tempe, Arizona, USA.
²Intel Corporation, Chandler, Arizona, USA.

Abstract - Structure from Motion (SfM) is a fundamental capability in Advanced Driver Assist Systems (ADAS) which require accuracy and performance. This paper presents an improved solution to depth estimation based on a low-complexity and low-delay (LOCAD) 3D SfM approach that requires as few as two images from a calibrated camera and that makes use of a new Multi-scale Fast Feature Point Detector (MFFPD). Results are presented to illustrate the improved performance of the proposed method as compared to existing depth estimation methods.

Keywords: depth estimation, SfM, ADAS, feature detector

1 Introduction

ADAS have enjoyed considerable growth during the last ten years providing commercial applications such as collision awareness and intervention systems. These systems utilize onboard sensors such as tachometers, cameras and RADAR systems to measure depth and relative velocity between vehicles. Video cameras and vision systems provide cost-effective solutions and are employed in a wide array of ADAS use cases [1], including object detection, tracking and depth estimation.

There are a variety of methods devoted to depth estimation using cameras, most of which use stereo cameras [2] allowing a straightforward utilization of the epipolar geometry constraints [3]. However a monocular vision system can also be used to recover 3D information about the scene in a more cost effective way. The accuracy of a depth estimation algorithm directly correlates with the quality of the features it is working with. An efficient feature detector must be able to produce a rich set of features covering all critical corners in a given frame. Invariance to factors such as motion, scaling, illumination and geometric transformation is key to the quality of the edges and corners detected by a quality feature detector. Not only the feature detector must be able to detect distinct features that can be tracked across multiple frames with minimum effort, but it also must do so in real time, matching the input video stream’s frame rate.

In an earlier work [4], the authors proposed a robust sparse depth estimation system for ADAS using a real-time key-point detector and a multi-frame based SfM method. However, this latter method exhibits a relatively high computational complexity as it makes use of iterative optimization and a high-complexity metric upgrade stage in order to make the depth estimation robust to drifts in camera calibration parameters. In addition, the multi-frame (also known as multi-view SfM) method [4] introduces excessive pipeline delay as it requires 4 to 8 video frames for performing depth estimation. Moreover, the feature detector used in [4] can be shown to produce falsely detected key-points along diagonal edges.

In this paper, we address the above issues by providing a fast Low-Complexity And low-Delay (LOCAD) depth estimation algorithm that requires as few as two images or two video frames from a calibrated camera. The proposed LOCAD 3D algorithm also consists of an improved Multi-scale Fast Feature Point Detector (MFFPD) that is robust to edge orientation and that results in a significantly reduced false key-point detection. The proposed LOCAD 3D SfM method can perform depth estimation from a single video stream that is captured using a calibrated camera. The new solution is low-delay as it does not require more than two frames from a video stream. Moreover, the proposed LOCAD 3D algorithm exhibits a significantly reduced complexity since it eliminates the need for the computationally intensive metric upgrade.

This paper is organized as follows. Section 2 presents a background on feature detectors and SfM. Section 3 presents the proposed MFFPD that is used as part of the proposed SfM approach. Section 4 describes the proposed LOCAD 3D SfM method. Results and comparison with existing depth estimation methods are presented in Section 5. A conclusion is given in Section 6.

2 Background

2.1 Key-point Detection

The Harris corner detector [5] is one of the most popular detectors for corner detection. The algorithm uses the structure tensor matrix to approximate the second-order derivative of sum of squared differences over a local neighborhood. Many improved versions of the Harris corner detector were proposed such as the one presented by Shi and Tomasi [6] which employs non maximal suppression to generate a ranking for a list of good corners.

There exists many algorithms [7] [8] that provide high accuracy in terms of repeatability and invariance towards scale and rotation. While these algorithms provide good accuracy, they are computationally expensive and are time consuming on a general CPU platform with limited or most
likely no parallel processing capabilities. Therefore, such detectors might not be appropriate for an application with hard real-time constraints such as ADAS use cases.

Numerous other key-point detectors were proposed in the recent past with emphasis on real-time performance and low-computational complexity. These algorithms exploit the pixel intensities around the neighborhood of the pixel of interest [9] [10] [11] [12]. One such algorithm proposed by is called the Features from Accelerated Segment Test, abbreviated as FAST [10]. The algorithm calculates the absolute differences between the center pixel and 16 pixels that form a circle around the reference pixel. Machine learning techniques are used to pick the order in which the circular neighborhood pixels are selected for processing in order to accelerate the computation.

While FAST is known to provide high repeatability, it has a few drawbacks, including false detection at the corners [11]. In order to overcome this drawback, Rublee et al. proposed the oFAST detector [11] which uses the Harris measure [5] and intensity centroid [13] to filter falsely detected corners and hence make the detection more robust. However, the addition of the Harris measure and intensity centroid increases the computational intensity. Another major drawback of FAST is its dependency on training dataset since it utilizes machine learning techniques for corner detection. AGAST [12] was proposed in order to overcome this drawback which frees the algorithm from depending on any dataset. AGAST also provides improvements in terms of execution speed.

A key-point descriptor is often used for robust key-point matching across different views of the scene with consideration for scale and rotation. SIFT [7] and SURF [8] provide highly robust key-point matching but are computationally intensive. Recent state-of-the-art algorithms such as BRISK [14] and FREAK [15] serve as real-time key-point descriptor while producing reasonable accuracy and repeatability. BRISK also implements a multi-scale version of the FAST that can be used to obtain scale invariance [14].

Recently, Nain et al. proposed a key-point detector, referred to as Fast Feature Point Detector or FFPD for short, that uses only a 4-pixel neighborhood, instead of the 16-pixel neighborhood used in FAST and AGAST, to detect key-points [16]. Additionally, in FFPD, non-maximal suppression of key-points is performed by calculating an information content (IC) based score in order to determine the significance of each detected key-point [16]. The authors of [16] have shown that the algorithm is resilient to noise and uses only 28 operations per pixel. However, FFPD was developed for single-scale applications and, hence, it cannot provide scale invariance when coupled with a key-point descriptor.

A multi-scale fast feature point detector, known as MFFPD was proposed recently by the authors to address the above mentioned issues [4]. Similar to FFPD, MFFPD uses a 4 pixel mask to detect key-points. As one of the improvements over FFPD, MFFPD eliminates the need for calculation of IC which, in turn, improves the speed of execution. It was shown [4] that MFFPD has a robust scoring pattern along the same scale as well as across multiple scales and that it exhibits better speed and accuracy as compared to the state-of-the-art multi-scale FAST when using the BRISK descriptor [4]. In this paper, we propose an improvement to the MFFPD. The improved detector provides improved performance with respect to reduced false key-point detection over diagonal edges. The improved MFFPD detector does not require many branching operations, is highly parallelizable, and makes use of simple masking operations. These characteristics make the implementation highly favorable for computing resources that support SIMD architecture, for example, GPUs. More details about the improved MFFPD key-point detector are presented in Section III.

### 2.2 Structure from Motion (SfM)

In [17] a real-time SFM of a static scene from a relatively narrow region is presented. The method requires a collection of several hundreds of images under short baseline displacement. The approach does not use feature tracking but depends on stable photometric information between neighboring frames. This makes this method not resilient to local illumination changes.

In [18] a real-time SFM system specifically designed to work with a calibrated handheld camera and in a small AR (Augmented Reality) workspace is presented. The tracking and mapping tasks are processed in parallel threads using a multi-core computer. A restrictive condition over the scene is that it should be mostly static.

As in the aforementioned approaches, we also assume that internal camera parameters are kept fixed and therefore a pre-calibrated camera is used; however, in contrast to the previous approaches, depth estimation in ADAS is expected to work reliably under wide baseline displacements, continuously changing scenes, and using a very small number of video frames (in our case, only 2) for acceptable real-time performance.
3 Multi-Scale Fast Feature Point Detector

A block diagram illustrating the main stages of the proposed MFFPD key-point detector, is shown in Figure 1. A scale-space pyramid of \( n \) octaves and \( n \) intra-octaves is first built as described in the BRISK framework [14] using the input image in order to obtain high accuracy. All comparisons and experimental results in this paper are generated using

A pixel is considered an initial key-point candidate if \((PD1 \text{ OR } PD2) \text{ AND } (SD1 \text{ OR } SD2) > P1\). These additional checks ensure that the initial key-point candidates selected do not contain pixels belonging to diagonal edges.

It is also important to note that, while the addition of diagonal checks adds to computation, it is performed only on the pixels that are corners or diagonal edges, and not on all the pixels in the image. As most of the pixels in the image are already filtered using the first stage of the detector, the diagonal checks adds to computation, it is performed only on the pixels that are corners or diagonal edges, and not on all the pixels in the image. As most of the pixels in the image are already filtered using the first stage of the detector, the diagonal check operations are performed only on a small fraction of pixels and hence the computational latency is

\[
P_D = \frac{1}{2} \left( I_1 - I_2 \right)
\]

\[
S_D = \frac{1}{2} \left( I_3 - I_4 \right)
\]

where \( I_1, I_2, I_3, I_4 \) correspond to the intensity values at the four corners of the 2x2 neighborhood containing the pixel.

In order to estimate the depth information that is lost during the image acquisition \( \mathbb{R}^3 \rightarrow \mathbb{R}^2 \), correspondences among multiple images are used [19], [20], [3].

Most of the SFM approaches require some kind of feature detection to be performed on every image or every video frame that is used. The SIFT key-point detector presented by Lowe [7] has been dominantly used for this purpose.

For best 3D reconstruction, the detected features have to be matched across all images. This task was initially accomplished using methods like patch correlation, which were only valid under restrictive conditions e.g., short baseline rigid motion. However feature detectors that provide a feature descriptor [21]; a.k.a. feature signature, allow matching under more general conditions such as wide baseline matching which is nowadays a requirement for depth estimation for ADAS. Feature correspondences across multiple images are used to estimate the camera pose for each image, i.e., rotation and translation, which allows performing the 3D projective reconstruction using methods like triangulation. In order to upgrade the projective reconstruction to a metric reconstruction, the intrinsic information about the imaging sensor should be used. When this information is not available it can be estimated using auto calibration methods [22]. Finally, in batch or non-real-time applications, a global optimization process known as bundle adjustment [23] can be used to refine results and produce a jointly optimal scene reconstruction; however, this approach is slow and may not be feasible in applications such as the ADAS space, where actions have to be triggered in real time.
minimal. Moreover, by performing the diagonal check and filtering pixels belonging to diagonal edges, we eliminate all further processing on these pixels. Therefore, the improved MFFPD detector produces key-points that are robust to diagonal edges while maintaining the speed of the original MFFPD.

The flow of the remainder of the algorithm is similar to that of [4]. The initial key-points are refined using Pseudo-Gaussian masks in order to remove pixels that are falsely classified as key-points due to noise. The product of H1, H2, V1 and V2 is allocated as the score for the refined key-points in each scale space. It is shown in [4] that this scoring pattern is more robust in terms of non-maximal suppression than the IC-based score of FFPD. The proposed scoring pattern also eliminates the need for additional computations that are required for calculation of IC.

The improved MFFPD detector is highly robust to false detections along diagonal edges while maintaining the speed of the original MFFPD. In order to check the robustness of the improved MFFPD, key-point detection was performed on a synthetically generated image of size 200×200 containing a white square rotated at 45° from the horizontal. Figure 2 shows the comparisons between the original and the improved MFFPD using the same threshold parameters. It can be seen from Figure 2 that, in contrast to the original MFFPD, the improved MFFPD does not result in false detections along diagonal edges. The detectors were also compared using natural images and the improved MFFPD produced better accuracy with respect to false detection of corners. An example is shown in Figure 3. A comparison of the execution speed between the original MFFPD and the improved MFFPD is shown in Table 1. The improved MFFPD results in a lower execution speed as compared to the original MFFPD as the computation required in the further stages of the key-point detection are eliminated for a large number of falsely detected key-points along the edges. All experiments were conducted on the same computer platform with a 4-core Intel i7-3770 3.48GHz processor and 8GB RAM.

4 Low-Complexity and Low-Delay Structure from motion Using MFFPD Key-points

We propose a low-complexity and low-delay (LOCAD 3D) SfM algorithm that is able to provide depth estimations using as few as two frames from a single calibrated camera. Figure 4 presents the flowchart of the proposed LOCAD 3D SfM algorithm. The pipeline utilizes two images I1, I2 of the same scene with time stamps t1, t2. I1 and I2 do not need to be consecutive images as long as there is minimum partial commonality in the scenes. In other words, both images have information of the same scene but from different poses of the camera. The camera pose change between I1 and I2 can be described in terms of the rotation matrix \( R \in \mathbb{R}^{3 \times 3} \) and the translation vector \( t \in \mathbb{R}^{3 \times 1} \). The two frames I1 and I2 are analyzed in terms of their features \( x^1_i \) and \( x^2_i \), respectively. The features, i.e., \( x^1_i, x^2_j \), are detected using the proposed improved MFFPD detector and are matched using the BRISK feature descriptor, providing the correspondences \( x^1_k \leftrightarrow x^2_k \), \( k = 1, 2, \ldots, L \) which are utilized to compute the Fundamental matrix \( F \in \mathbb{R}^{3 \times 3} \). The \( F \) matrix together with the internal camera parameters, which are computed off-line, are used to define the Essential matrix, which provides the external parameters \( R, t \) within a four-fold ambiguity framework [3]. The aforementioned ambiguity can be solved experimentally by testing positiveness on 3D reconstructed points’ depths. A triangulation stage is used to obtain 3D points from the 2D correspondences once the camera matrices for each pair of images I1, I2 have been defined. Since the \( E \) matrix provides pose parameters up to an unknown scale factor, it has to be corrected using a ground truth depth. More details about the different components of the proposed approach are further given in the subsequent sections.

4.1 Internal camera calibration

The toolbox provided by [22] was used to determine the \( K \) matrix. The \( K \) matrix is assumed to remain constant during the image or video acquisition. In other words the assumption is that the camera's zoom and focus are fixed.

4.2 Computation of Fundamental and Essential matrices

The \( F \) matrix can be computed with as few as eight noiseless feature correspondences between \( I_1 \) and \( I_2 \) [3]. However, in order to compensate for noise, MFFPD is used jointly with RANSAC [24] for a more robust estimation of \( F \). In RANSAC, a subset of correspondences, greater than eight, is used to compute \( F \) as follows:
\[ x_k^1 \cdot F \cdot x_k^2 = 0, k = 1, \ldots, L \] (5)

The computed \( F \) is then scored by the number of inliers, i.e., the number of correspondence pairs that satisfy (5) within a predefined tolerance, the \( F \) matrix that gets the higher number of inliers is used to compute the \( E \) matrix. Since a unique camera is used, the Essential matrix \( E \in \mathbb{R}^{3 \times 3} \) is computed as:

\[ E = K^T F K \] (6)

### 4.3 Camera projection matrices

Camera projection matrices \( P_1, P_2 \) extracted from \( F \) are subject to the well-known projective ambiguity whereas the Essential matrix provides camera projection matrices up to a known scale factor and a four-fold ambiguity.

The unknown scale factor ambiguity is considerably easier and faster to correct compared to the projective ambiguity which is also known as metric upgrade [25]. The four-fold ambiguity on the other hand, means that given \( \mathcal{E} \), the corresponding \( \mathcal{E} \) with homogeneous coordinates \( (X,Y,Z,T) \) and a camera projection matrix \( P \in \mathbb{R}^{3 \times 4} \), the 2D projected point \( x \) has homogeneous coordinates given by:

\[ x(x,y,1) = P \cdot X(X,Y,Z,T) \] (9)

The depth of \( X \) in front of the principal plane of the camera can be determined as follows [3]:

\[ \text{Depth}(X,P) = \frac{\text{sign}(\text{Det}(M)) \cdot x}{T \cdot \| r^3 \|} \] (10)

where \( P = [M | c] \) and \( r^3 \) is the third row of \( M \in \mathbb{R}^{3 \times 3} \). In order to address the noise effects, we look for the pair of camera projection matrices that provide positive depths in all matched features \( x_1^1 \leftrightarrow x_1^2 \). If that condition is not met the process starting with the computation of the fundamental matrix \( F \) is repeated. The random behavior of RANSAC used to compute \( F \) provides a slightly different solution for \( F \) in every run. At the end, if the set max number of iterations does not result in the total positiveness condition, the requirement is relaxed by selecting the pair of camera projection matrices that provides positive depth in at least 99\% of the features.

### 4.4 Triangulation

In order to solve the four-fold ambiguity over the camera matrices, the 3D reconstruction of the matched features \( x_1^1 \leftrightarrow x_1^2 \) is performed for all four possible pairs of \( P_1 \leftrightarrow P_2^{-4} \). Figure 5 illustrates the 3D reconstruction from 2D correspondences, where \( C_1 \) and \( C_2 \) are the camera centers of the equivalent cameras corresponding, respectively, to frames \( I_1 \) and \( I_2 \) (note that only one camera is acquiring the frames but each frame can be thought of as being acquired by a separate camera). \( C_1 \) corresponds to the \([0,0,0]\) 3D coordinates i.e., the origin of the World Coordinates System (WCS), and \( C_2 \) corresponds to the right null zero of \( P_2 \) [3].

As shown in Figure 5, for every pair of 2D correspondences on the set \( x_1^1 \leftrightarrow x_2^2 \), two 3D lines \( \ell_1, \ell_2 \) can be generated passing by \( C_1 \), and \( C_2 \), respectively. In the noiseless case the intersection of \( \ell_1 \) and \( \ell_2 \) defines the 3D position of the point. However when noise is present the intersection of \( \ell_1 \) and \( \ell_2 \) is not guaranteed. In this case, a triangulation algorithm that minimizes the geometric error is utilized [3]. The originally detected feature point correspondences are refined by finding neighboring feature points that satisfy the epipolar constraint (5) and that would result in intersecting back-projected lines \( \ell_1, \ell_2 \).

### 4.5 Camera projection matrix selection

According to Figure 5, the reconstructed 3D points \( X \) should be in front of both cameras; i.e., their \( Z \) coordinate with respect to the WCS is always positive. This observation is used to select the correct pair of camera projection matrices out of the four possible \( P_1 \leftrightarrow P_2^{-4} \). Given a 3D point \( X \) with homogeneous coordinates \( (X,Y,Z,T) \) and a camera projection matrix \( P \in \mathbb{R}^{3 \times 4} \), the 2D projected point \( x \) has homogeneous coordinates given by:

\[ x(x,y,1) = P \cdot X(X,Y,Z,T) \] (9)

The depth of \( X \) in front of the principal plane of the camera can be determined as follows [3]:

\[ \text{Depth}(X,P) = \frac{\text{sign}(\text{Det}(M)) \cdot x}{T \cdot \| r^3 \|} \] (10)

where \( P = [M | c] \) and \( r^3 \) is the third row of \( M \in \mathbb{R}^{3 \times 3} \). In order to address the noise effects, we look for the pair of camera projection matrices that provide positive depths in all matched features \( x_1^1 \leftrightarrow x_1^2 \). If that condition is not met the process starting with the computation of the fundamental matrix \( F \) is repeated. The random behavior of RANSAC used to compute \( F \) provides a slightly different solution for \( F \) in every run. At the end, if the set max number of iterations does not result in the total positiveness condition, the requirement is relaxed by selecting the pair of camera projection matrices that provides positive depth in at least 99\% of the features.

### 4.6 Moving average and scale upgrade

In order to produce a jointly optimized scene reconstruction, bundle adjustment [23] can be used. However, bundle adjustment is computationally expensive and a lighter algorithm is warranted. For a faster execution time with minimum impact to the quality of the result, a moving average is applied across sequential in place of bundle adjustment. The scale ambiguity cannot be avoided but it is easily resolved with ground truth depth information of a single 3D point as [26].

### 5 Results

In order to test the proposed approach, two experiments were carried out, one in a controlled indoor environment and one under real outdoor conditions.

Images used in both experiments were captured using a calibrated camera Sony IMX135 Exmor RS CMOS sensor with pixel size 1.12\( \mu \)m, and the resolution of the images was 1280\( \times \)960. The ground-truth depths of objects in the scene of both experiments were measured manually. The proposed LOCAD 3D algorithm was tested on a sequence of eight frames for the indoor experiment and nine frames for the outdoor one. The key-points were detected using the improved MFFPD on the initial image or frame. In both experiments these features were tracked in subsequent images using the KLT with an error threshold of 10 pixels.
and the average re-projection error threshold for estimating the projective matrix was 0.25 pixels.

The indoor experimental setup, which is shown in Figure 6, has four regions at four different depths. Detected features within every region are also shown in Figure 6. The average depth estimation error for every region is shown in Figure 7 where the ground truth depth of region four was used to scale the computed depth results.

Samples from the outdoor image data set are shown in Figure 8. In order to examine the accuracy of our depth estimation results we have defined three regions in the scene, where points in each region have nearly the same depth. These regions are shown in Figure 9 together with their detected feature sets. For clarity, only 6% of the detected features have been randomly selected and plotted. In Figure 10 the ground-truth of the three defined regions are plotted together with the estimated depth measurements for every pair of correspondences $x^l_i, x^r_i$. Due to the fact that features over region 2 (features over the motorcycle) are used as ground-truth references for scaling purposes, that region presents zero mean variation. However, regions 1 and 3 show 3.1% and 0.6% variations, respectively. Motivation for using the term “variation” instead of “error” is the fact that features within a region (in the real-world outdoor scene) do not precisely have the same exact depth.

Table 2 shows a performance comparison in terms of execution time and re-projection error between the proposed method and the one presented in [4]. For this purpose, both implementations were run on a desktop with an Intel i7 2.19GHz processor with 8GB of RAM. Both algorithms were implemented in OpenCV2.4.9 and compiled using the Intel C++ compiler 14.0 in Microsoft Visual Studio 2013. It should be noted that the method of [4] requires several frames (about 6 to 8) for depth estimation as opposed to only 2 frames for the proposed method. In addition, the existing method [4] requires several complex optimization procedures (bundle adjustment and metric upgrade) that are not needed in the proposed method. Since the proposed method does not use bundle adjustment nor metric upgrade, it is substantially
faster while incurring only a minimal penalty in the re-projection error.

6 Conclusion

A low-complexity and low-delay (LOCAD) SfM algorithm that requires as few as two images is proposed for depth estimation. The proposed LOCAD SfM depth estimation method consists of a fast improved multi-scale feature point detector (MFFPD) The improved key-point detector produces better performance in terms of execution speed and reduced false key-point detection and is more robust to edge orientation. Since a single monocular camera is used, the system requires a ground-truth value in order to define the scale factor. Results are over an order of magnitude faster with respect to existing SfM algorithms with just a minimal penalty on the re-projection error.

REFERENCES


