Robustness of 3D Point Positions to Camera Baselines in Markerless AR Systems

Deepak Dwarakanath[1,2], Carsten Griwodz[1,3], Pål Halvorsen[1,3]
[1]University of Oslo, Oslo, Norway
[2]Image Metrology A/S, Horsholm, Denmark
[3]Simula Research Laboratory, Lysaker, Norway
deepakdw@ifi.uio.no, {griff,paalh}@simula.no

ABSTRACT

In the Augmented Reality (AR) applications, high quality relates to an accurate augmentation of virtual objects in the real scene. This can be accomplished only if the position of the observer is accurately known. This boils down to solving image-based location problem by an accurate camera pose (relative position and orientation) estimation, when a stereo or multiple camera setup is used. Consider a relevant application scenario as in a movie production set, where the director is able to preview a scene as an integrated view of the real scene augmented with animated 3D models. The main camera shoots the scene, where as secondary stereo camera pair is used for image registration and localization. The director can view the integrated preview from any viewpoint perfectly, as long as the camera pose estimation is accurate.

Moreover, in the case of a markerless AR system, the challenge for camera pose estimation, is strongly influenced by the precision of detected feature correspondences between the images. Unfortunately, several of the state-of-art feature extractors (detectors and descriptors) cannot guarantee a consistent accuracy of camera pose estimation, especially at varied camera baselines (viewpoints). As a consequence, the precise augmentation of objects, as desired in an AR application, is compromised. Hence, it becomes necessary to understand the magnitude of this error in relation to the camera baseline depending on the chosen feature extractors.

We, therefore, assess the quality of the position and the orientation of 3D reconstruction by evaluating 26 feature extractor combinations over 50 different camera baselines. To be directly relevant for AR applications, we evaluate by measuring the reconstruction error in 3D space, instead of re-projection error in 2D space. After the experiment, we have found the SIFT and KAZE feature extractors to be highly accurate and more robust to large camera baselines than others. Importantly, as a result of our study, we provide a recommendation for system builders to help them make a better choice of the feature extractor and/or the camera density required for their application.

MMSys'16, May 10-13, 2016, Klagenfurt, Austria

DOI: http://dx.doi.org/10.1145/2910017.2910611

CCS Concepts

•**Information systems** → **Multimedia information systems;**

Keywords

Augmented Reality, Feature Extraction, Pose Estimation, 3D Accuracy

1. INTRODUCTION

The multimedia industry has paid quite a lot of attention to 3D imaging as in head mount virtual reality systems [1, 2], augmented reality mobile applications [3, 4, 5], interactive augmented reality systems [6, 7], free-viewpoint rendering [8], etc. These applications use two or more cameras to perform tasks such as augmenting 3D models in video sequences, depth estimation, virtual view synthesis, etc. The underlying principle of such multi-camera systems is the estimation of camera pose, i.e., relative camera position and orientation with respect to other cameras.

A central theme in Augmented Reality (AR) research is the enhancement of the human senses by changing what human observers see with their eyes, or annotating it. Of these, modification is more challenging because accurate knowledge of the images that the observers see is required before changes can be made. This knowledge may be derived by augmenting the observers with cameras mounted on their heads [9], and perhaps reconstructing their entire view. Our project goal in POPART[1], however, is to provide an augmented, accurate preview of a film set. This is meant to provide an integrated view of real-life actors with prototype animated 3D models in real-time to director and photographer, weeks or months before post-production is finished. This implies that we augment the image that is seen by the main film camera, and that we have one or two static cameras to estimate the dynamic objects. The static film set itself is, in our case, reconstructed in advance of the filming.

The accuracy of the camera pose estimation plays an important role in order to determine the quality of these applications. Cameras are usually pre-calibrated offline (often, focal length and principal axis are determined using a checkerboard - Matlab Toolbox[2]). When the system is deployed, the camera pose is estimated automatically based on sparse feature points extracted from the images that are

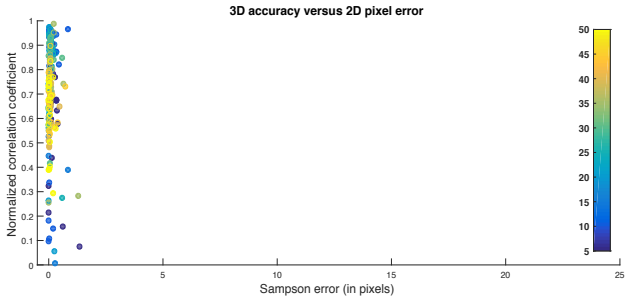
[1]http://www.popartproject.eu, EU Horizon2020 project number 644874
[2]http://www.vision.caltech.edu/bouguetj/calib_doc

captured by these cameras. This is also known as Feature-Based Calibration (FBC).

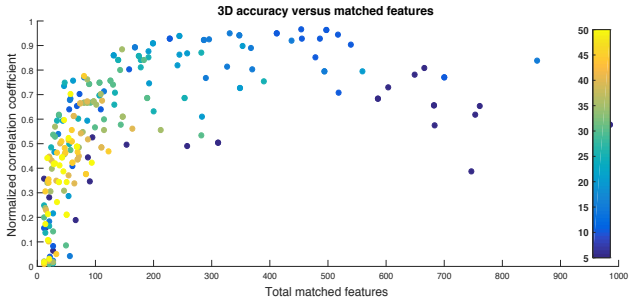
In multi-camera systems, the following statements are commonly accepted:

- A high number of matched feature points in a stereo pair results in a better camera pose estimation.
- Minimizing 2D pixel error calculated between matched pairs results in higher accuracy of 3D estimation, based on epipolar geometry [10].

The first point holds good for iteration-based estimation algorithms (e.g., RANSAC [11]). The second point, however, is not always true. We illustrate this in figure 1, which represents a scatter plot of 3D accuracy versus 2D pixel error and number of matched feature points extracted from images of stereo pair at various baselines (relative displacement between the stereo cameras). Figure 1(a) illustrates that low pixel error does not guarantee high 3D accuracy and, similarly, figure 1(b), that high 3D accuracy is not always obtained by a larger number of feature matches.



(a)



(b)

Figure 1: Scatterplots of matched feature points and 2D pixel error with 3D accuracy.

In this paper, we explore one of the important factors determining the accuracy of camera pose estimation and thereby 3D estimation, i.e., change in the camera baseline, which breaks the common assumptions made above. This paper also casts light upon the quality of current state-of-art feature extractors (combination of detectors & descriptors) [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] used for FBC or camera pose estimation today.

Each of these feature extractors has its own behavioral traits. Some of them claim invariance to change in camera baseline, but the extent of their tolerance is uncertain. Therefore, we evaluate various combinations of feature extractors with a brute-force matcher to determine their robustness to change in the camera baseline. Our study is meant to provide system builders with a better understanding of the operational limits of the state-of-art feature detectors and descriptors. It will help them to make better choices in designing 3D multimedia applications using multi-camera systems. Besides choice of algorithm, it may be helpful in estimating the number and position of cameras that are required for reconstructing rigid structures in a well-known space, with a desired accuracy.

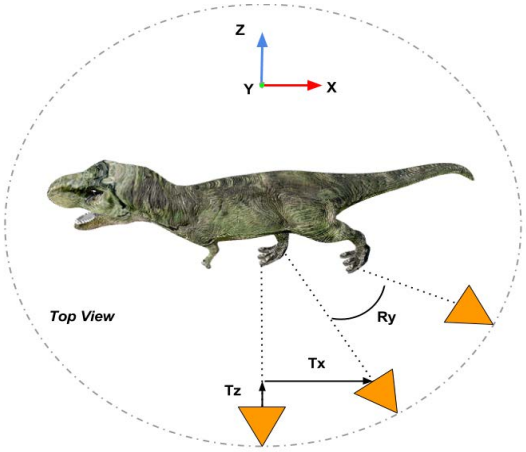


Figure 2: Cameras arranged in a circular configuration around the 3D model.

We have considered a multi-camera scenario as in figure 2, where a number of cameras are placed in a circular configuration around and looking at an object of interest, equidistant from the object's geometric center. We have chosen this configuration to concentrate on changes in baseline, and avoid changing either the objects' size in the frames or the camera's focal length between baseline configurations. This would be unavoidable, if we changed camera baselines along a line. So, with these configurations, we study the performance of feature extractors on stereo pairs. Furthermore, we have chosen to work on pure virtual scenes, which guarantees that we know the exact ground truth of 3D points position and their corresponding pixel positions, and use it for the assessment of reconstruction quality. The quality of AR applications is determined by the observer's relative position in 3D space. We, therefore assess the quality in the reconstructed 3D space, which seems more realistic for our scenario, than the usual re-projection error in 2D space.

The rest of the paper is organized as follows: section 2 describes other related feature evaluation studies. The evaluation system is explained in section 3 and the results are discussed in detail in section 4, along with the recommendations for designing 3D applications. Finally, we conclude by stating the usefulness of the evaluation study and outline the scope for future work, in section 5.

2. RELATED WORK

Previously, we have seen that the evaluation of most of the state-of-art feature extractors, i.e., detectors or descriptors, use various evaluation criteria. The feature detector KAZE [16] and feature descriptors FREAK [22] and BRIEF [21] evaluate themselves with other known feature

detectors using recall and precision metrics, which relates to a total number of correct feature matches found. Along with recall and precision, BRISK [15], STAR [19], FAST [20] and AKAZE [17], evaluate themselves in comparison to others, by the metric repeatability, which measures the extent of overlap between the detected regions in an image pair. In both SIFT [12] and SURF [13], the evaluation is carried out on various viewpoints, but not in comparison to other features. However, the performance criteria is still repeatability. Sometimes, the distance between the descriptors is considered to be an evaluation metric, as in ORB [14]. In all the above cases, the evaluation criteria focuses only on the correctness of the feature matches and this may not be enough to evaluate the feature extractors for accuracy in 3D applications and robustness to camera baseline changes.

Point feature matching algorithms for stereo were evaluated by Juhász et al. [23], but only for a particular baseline based on the re-projection error metric. In our paper, we evaluate a range of baselines to study their effects. Interest point detectors and descriptors were evaluated for tracking applications by Steffen et al. [24], where detectors were tested on various conditions such as scale, rotation, baseline, light, etc., using repeatability metric. Further, feature detectors were compared based on tracking success rate, which was computed based on the re-projection error. However, KAZE, AKAZE, BRISK, BRIEF and FREAK are not included in their study, unlike ours. Moreover, instead of measuring the re-projection error in 2D, we measure the accuracy in 3D space directly, relying on a dataset consisting of known 3D models. We believe that 3D space metrics are more suitable for AR related applications.

Michael et al. [25] evaluated SIFT feature extractors for viewpoint invariance, by comparing the descriptor properties over various baselines. Their evaluation basically outlines the quality of obtaining correct matches, but it does not guarantee high 3D accuracy.

Florian et al. [26] evaluated feature tracking for pose estimation in underwater environment. However, their evaluation is limited to very few feature detectors and descriptors with a very specific testing condition.

Pierre et al. [27] evaluated feature extractors for 3D object recognition applications over various viewpoints and lighting conditions, but with a limited number of candidates for evaluation.

Comparatively, in our paper, we evaluate a wide range of feature extractor combinations, to describe its capability for 3D applications directly, over various camera viewpoints.

3. EVALUATION SYSTEM

Our setup for evaluating feature extractors is depicted in figure 3. It comprises of the following steps: dataset generation, feature extraction, pose estimation, 3D estimation and 3D accuracy computation. The evaluation is carried out based on the accuracy of the 3D points that are estimated using the 2D test points, in comparison with the ground truth derived from the 3D model. Our experiment is implemented in C++ using the OpenCV (Open Source Computer Vision) library and results are presented using Matlab.

3.1 Dataset Generation

Ground truth data is generated based on the application scenario illustrated in figure 2. Here, we consider a number of possible positions where cameras can be placed around the

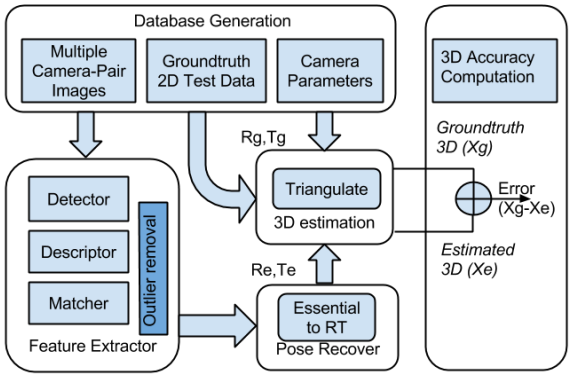


Figure 3: Experimental setup

3D model. Subsequently, we considered that many stereo camera pairs to capture images of a 3D model at various baselines (refers to relative displacement of stereo cameras). For every subsequent stereo pair, the camera motion is circularly displaced, maintaining equal distance from geometric center of the 3D object. This configuration is deliberately chosen so that scaling effects on feature extractors can be nullified and the focus stays on evaluating only baseline variation. Using 3D models is an advantage, in terms of having full control over the dataset being generated. The dataset is generated using a total of 9 3D models (depicted in figure 4 and obtained from CG Trader[3]), for baselines varying from 1 to 50 degrees angular displacement. This results in the necessary ground truth values as follows:

- Totally, 450 stereo pair images with 1 degree resolution, are generated for 9 models. Images are of resolution 600x600 with 24 bit depth.
- The ground truth 3D points are generated using four points, representing an origin and three points of unit length in three axes direction. These 3D points $[X_g]$ are sufficient to represent a model measured in world co-ordinate system, with the geometric center of the model as the origin. This type of 3D data is well suited as ground truth data, which is compared with the estimated 3D data, to compute the changes in the position and rotation in 3D space.
- The ground truth 2D feature points [x_g^1 and x_g^2] in stereo pairs corresponds to the projection of true 3D points onto the image plane. This is considered as the 2D test data, which is used in the experiment to evaluate the feature extractors.
- The camera intrinsic parameters $[K]$ comprises camera's focal lengths (f_x,f_y) and principal axes (p_x,p_y). All cameras have identical intrinsics in all tests. In our experiment, the focal length is 520 pixels and the principal axes are 300 pixels.

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

[3]http://www.cgtrader.com

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4: 3D models used for the experiment. From each model, 50 stereo image pairs are generated, corresponding to various baselines.

- The camera extrinsic parameters represents relative rotation and translation of stereo pair ($[R_g,T_g]$).

$$R_{g3X3} = \begin{bmatrix} r_{x1} & r_{y1} & r_{z1} \\ r_{x2} & r_{y2} & r_{z2} \\ r_{x1} & r_{y3} & r_{z3} \end{bmatrix}, T_{g3X1} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

3.2 Feature Extractors

The term feature extractor refers to a combination of state-of-art detector and descriptor. After feature extraction, the features are matched and outliers are removed.

We have tested feature extractors by combining the detectors SIFT, SURF, BRISK, KAZE, AKAZE, ORB, MSER, STAR and FAST, with their own descriptors, and combined with BRIEF and FREAK descriptors. In total, we evaluated 26 feature extractor combinations. To compute feature correspondences in a stereo pair, we applied a brute-force matcher on the descriptors, combined with Random Sample Consensus (RANSAC) [11] for removal of outliers. Each feature extractor was applied to every camera pair configuration to extract feature correspondences $[x_e^1, x_e^2]$ between the stereo images. All the state-of-art feature detectors and descriptors used for the evaluation in this paper are briefly explained with their properties in table 1.

3.3 Pose Recovery

In our tests, pose recovery estimates the pose (camera position and orientation) of the right camera with respect to the left camera in a stereo pair. Feature correspondences from the feature extractors on every stereo pair are used to estimate the camera pose $[R_e,T_e]$. Feature correspondences $[x_e^1, x_e^2]$ are used to estimate the essential matrix $[E_{ss}]$ directly, given the camera instrinsics $[K]$, by applying the 5-Point algorithm [28]. The essential matrix is a specialized case of fundamental matrix expressed in normalized image coordinates that describes the relation between the stereo pair in terms of epipolar constraint $[x_e^{2T} E_{ss} x_e^1 = 0]$.

Finally, the camera pose is recovered using a single value decomposition, $E_{ss} = [T_e]R_e$, and selection of the optimal solution using the cheirality constraint [10]. Thereby, the estimated camera position is always upto scale expressed in model coordinates.

3.4 3D Estimation and Accuracy Computation

Usually, an estimated 3D point is projected onto a 2D image and compared with a known value to compute reprojection error, which represents the accuracy of the estimation. Instead of following this approach, we estimate the error in 3D space that is more comparable to real-time applications, using Normalized Correlation Coefficient (η).

For feature based calibration, in our tests, the feature extracted correspondences are consumed in estimating the camera pose. Using the same feature correspondences to estimate the 3D points is not a fair experiment to evaluate feature extractors for 3D applications.

Therefore, to evaluate feature extractors for feature based calibration, we compute 3D accuracy as a difference between the experimental data and the ground truth data. The ground truth 3D data (X_g) is obtained as a result of back projecting their corresponding ground truth 2D test data (x_g^1, x_g^2), using the ground truth camera pose (R_g,T_g). Similarly, experimental 3D data (X_e) is estimated from the same 2D ground truth test data (x_g^1, x_g^2) using the estimated camera pose (R_e,T_e). The back projection of feature corresponding points of two stereo pair is accomplished by triangulation [10]. Here, T_g & T_e are expressed upto scale, and all distances are always expressed in the model coordinates.

Thus, the 3D accuracy can be quantified as η, a measure over all three axes components between X_e and X_g. η provides a similarity measure of estimated 3D points with the ground truth 3D points, which is represented as a normalized accuracy value [0-low and 1-high].

$$\eta^\dagger = \frac{\sum(X_e^\dagger - mean(X_e^\dagger)) * (X_g^\dagger - mean(X_g^\dagger))}{\sqrt{\sum(X_e^\dagger - mean(X_e^\dagger))^2 * \sum(X_g^\dagger - mean(X_g^\dagger))^2}}$$

$$\eta = \sum_{\dagger=x,y,z} \frac{\eta^\dagger}{3}$$

where † represents 3D axes components x, y and z.

4. RESULTS AND DISCUSSION

The experiment described in section 3 is carried out on a total of 450(stereo pairs) * 26(feature extractor combinations), i.e., 11700 datasets. Our test results, which are based on virtual models in an empty scene, can be compared directly to a film scenario that applies blue screen, i.e. where the background consists of large, artificial, untextured surfaces. In other cases where textured background provides depth to the scene, our tests are relevant only for objects at certain depth. Other factors in real scenes, such as blur or challenging lighting conditions, are considered future work.

In our results, the "baseline" of the stereo camera pair is represented in terms of relative angular separation between the cameras, where both cameras are directly facing the 3D model and the camera movement with respect to each other is as in a turn-table configuration.

All combinations of feature extractors are evaluated at every stage in the pipeline (described in figure 3), i.e., 2D pixel error, camera pose error and 3D estimation error. As

Feature Extractor	Properties	Detection	Description
SIFT [12]	Scale and rotation invariant. Robust to change in illumination, 3D viewpoint and noise.	Interesting points are identified using Difference of Gaussian (DoG) over several linear scales of images. Then, the location and scale of keypoints are accurately computed using neighbor pixels.	The descriptor is represented by histograms of image gradients that are computed at every image point around the keypoints detected.
SURF [12]	Scale and rotation invariant. Features are distinctive, robust to noise, geometric and photometric deformations. It can be computed quickly.	Using integral images makes the image convolution faster. The detector is based on Hessian-matrix based approximation of blob-like interesting points using Gaussian scale space.	The descriptor is based on distribution of interesting points in its neighborhood. This is similar to SIFT but instead of using gradients, distribution of first order Haar Wavelets responses are used.
ORB [14]	Designed to perform two magnitudes faster than SIFT.	This is a FAST detector with addition of an accurate orientation component using intensity centroid.	"Rotation-Aware" binary descriptor based on the BRIEF descriptor. Computed by introducing a learning method for de-correlating the BRIEF features under rotational invariance.
BRISK [15]	Adaptive feature detector designed to lower computational complexity compared to SURF.	It is a combination of FAST detector in scale space and identifier of keypoints by fitting a quadratic function.	The descriptor is a bit-string assembly from intensity comparisons, retrieved by dedicated sampling of each keypoint neighborhood.
KAZE [16]	Scale and rotation invariant. Attains high accuracy in object boundaries. Robust to noise.	Similar to SIFT, except that the keypoints are detected in nonlinear scale space using "Additive Operator Splitting" techniques and variable conductance diffusion.	Uses a modified SURF descriptor, which adds a two-stage Gaussian weighting scheme.
AKAZE [17]	Accelerated KAZE - motivated to compute faster with similar scale and rotational invariance and lower storage requirement properties, compared to KAZE.	Instead of using non-linear scale space as in KAZE, a numerical scheme called "Fast Explicit Diffusion" in a pyramid framework is used.	A "Modified-Local Difference" binary descriptor, which exploits gradient and intensity information from nonlinear scale space.
MSER [18]	Affine-invariant feature extractor suitable for wide baselines in stereo. Robust to change in scale, illumination, out-of-plane rotation, occlusion and viewpoints.	Distinguished regions are detected and affine invariant procedure is carried out to estimate the stable invariant regions, from which the keypoints are measured.	n/a
STAR [19]	A suite of scale invariant center-surround detectors focused on visual odometry applications. Stable and repeatable in viewpoint changes. (CenSurE)	The CenSurE features are computed at the extrema over multiple scales using full image resolution using center-surround filters. There is an approximation to scale space based on Laplacian of Gaussian.	n/a
FAST [20]	High Speed corner detector extensively used in machine learning methods and is suitable for real-time applications.	Considers a circle comprising of 16 pixels in an image. Then every pixel is compared with only 4 neighbors to classify if it is a corner or not.	n/a
BRIEF [21]	A highly distinct binary descriptor designed to compute faster. Invariant to large in-plane rotation.	n/a	Binary string descriptor relying on image patches-pairwise intensity comparisons. A classifier is trained with image patches form various viewpoints.
FREAK [22]	Inspired by the human visual system - retina, this descriptor is a cascade of binary strings aimed at faster computation.	n/a	Computed by efficiently comparing image intensities over a retinal sampling pattern containing Gaussian kernel information.

Table 1: Brief overview of feature extractors that are used for feature based calibration.

a reference, 2D ground truth feature correspondences are passed through the pipeline with known camera parameters and reference plots at every stage are generated. These are referred to as "IDEAL" feature extractor combination, throughout the experiment. In this way, every step in the pipeline is tested as a black box to operate correctly.

The estimation error is expressed as averaged over every 5 degrees of the camera baseline. The variability of the error data within every 5 degrees is shown in figure 9. This variability makes it hard to present the comparison of the feature extractors visually. Therefore, the mean value was chosen to gain better readability.

4.1 2D pixel error

The 2D pixel error (P_{error}) is expressed as the squared Sampson error, which is the first-order approximation to the geometric error [10]. The P_{error} between feature points in a stereo pair is computed as in equation 1, where F is the fundamental matrix computed using N feature correspondences (x, x'). This metric determines how close every point in one image is to its corresponding epipolar line in the other image of the stereo pair. For an ideal match, $P_{error} = 0$.

$$P_{error} = \sum_{i=1}^{N_p} \frac{(x_i' F x_i)^2}{(Fx_i)_1^2 + (Fx_i)_2^2 + (F^T x\prime_i)_1^2 + (F^T x\prime_i)_2^2} \quad (1)$$

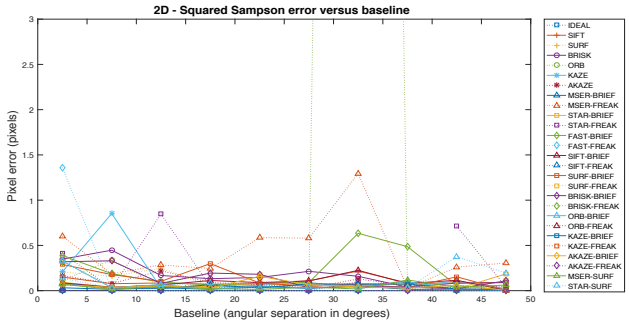


Figure 5: 2D error (Squared Sampson) based on epipolar constraint over varied baselines[4].

The P_{error} measured in 2D for stereo pairs varying in baseline is shown in figure 5. This error is computed for all meaningful combinations of feature extractors (described in the table 1). We can observe that the pixel error stays fairly low (although fluctuating) over all camera baselines. However, this does not guarantee a consistent accuracy of 3D estimation for all camera baselines as seen in figure 1(a). This is evident when we observe the effect of baseline variation on camera pose and 3D estimation error.

4.2 Camera pose error

Based on the estimated feature correspondences, the camera pose of stereo cameras are estimated. The pose estimated is compared with known camera extrinsics from the dataset (section 3.1), and thereby, the deviations of the estimated camera rotation and translation parameters from the ground truth value are computed. These deviations are the sum of deviations in all three axes, for both rotation and translation and are plotted in figures 6 and 7, respectively. Each figure is categorized into sub-figures based on the descriptors used, i.e., (i) figures 6(a) and 7(a) depict detectors having their own descriptors (with an exception for MSER and STAR, which uses SURF descriptor as in their original contribution), (ii) figures 6(b) and 7(b) depict detectors with BRIEF descriptor and (iii) figures 6(c) and 7(c) depict detectors with FREAK descriptor.

It is noticeable from the figures 6 and 7 that pose errors do not follow the same pattern as in figure 5. As the baseline of the stereo camera increases, the pose estimation error increases (figures 6(a), 6(b), 7(a) and 7(b)) or stays high throughout (figures 6(c) and 7(c)). This is observed to be due to the following reasons:

1. When wrong feature matches between the stereo pairs exist, the estimation of fundamental matrix becomes incorrect. This is quite obvious.
2. When correct feature matches between the stereo pairs exists, and if the feature matches are confined to a small area, i.e., a set of 2D match points corresponds to only a part of the 3D model, then the estimation of fundamental matrix becomes incorrect as there is not enough information about rotation or translation covering the whole 3d model.

In both of the above cases, an incorrect fundamental matrix and thereby an incorrect estimation of essential matrix results in an incorrect pose estimation. The 2D pixel error seems like a biased measure because the same number of feature points are used to both estimate fundamental matrix and to compute pixel error based on the fundamental matrix. Due to this nature, although we have an incorrect fundamental matrix, the 2D pixel error still stays low over all baselines (figure 5), as an effect of using RANSAC.

4.2.1 Penalty for invalidity

In the process of estimating camera pose, three types of invalidity can occur.

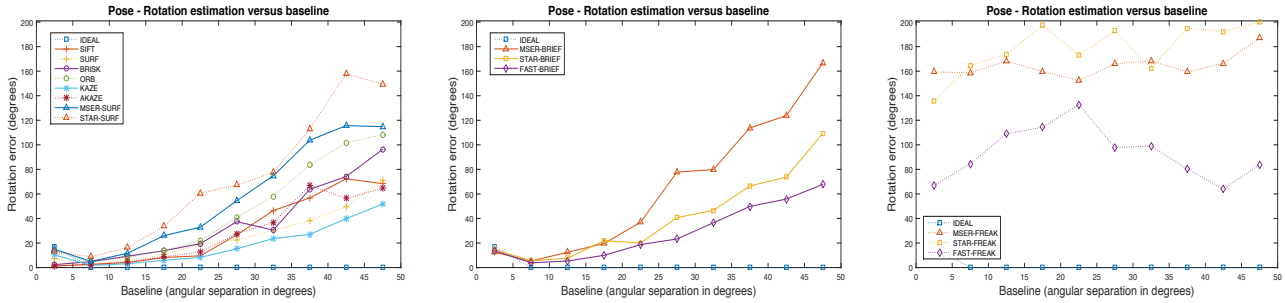
- Type 1 - when rotation error in either of the three directions is more than 90^o (as in figure 6(c)), then the camera seems to be rotated more than expected, in a true situation.
- Type 2 - as in figure 7(c), if any of the translation error is more than unity, then it means that the right camera is estimated to be on the left side.
- Type 3 - this is not directly related to pose estimation, but this error occurs when the feature extraction gives zero matches. This error also relates to non-estimation of fundamental matrix due to very few matches.

In the above cases, the camera pose estimation is deemed invalid. This situation can occur, when the number of feature correspondences in a stereo pair are zero or very few or wrong to a large extent. In these cases, we penalize the feature extractor, whenever any of the above types of invalidity occurs. Therefore, every feature extractor combination gets a penalty score for the invalidity.

In our tests, the penalties for every feature combination is given in figure 11. The maximum penalty score is 450, which represents samples that constitutes 9 models of 50 baselines each. It is clearly observable that most of the combinations with FREAK descriptor have higher penalty score.

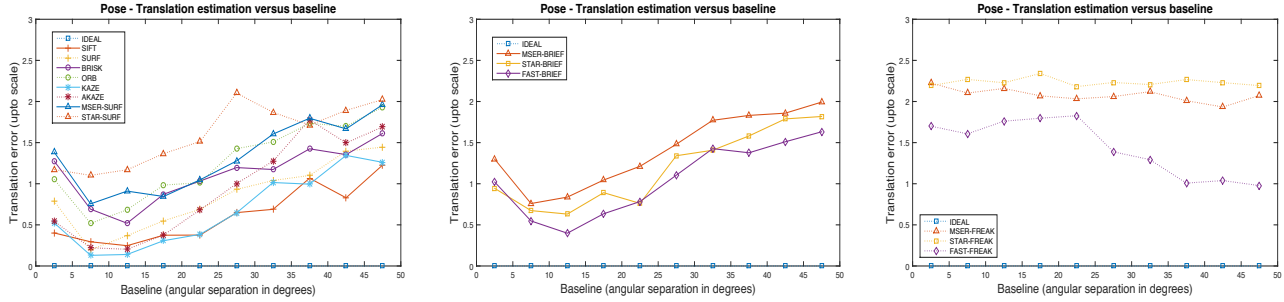
The sensitivity of the pose estimation can be observed by IDEAL features. The pose estimation seems to be sensitive

[4]The X-axis depicts baseline expressed between 1-50 degrees. Along the Y-axis, the error is averaged over every 5 degrees, to increase readability. Details in section 4.



(a) Own Descriptors (b) Brief Descriptor (c) Freak Descriptor

Figure 6: Mean estimation error of relative stereo camera rotation over varied camera baselines[4].



(a) Own Descriptors (b) Brief Descriptor (c) Freak Descriptor

Figure 7: Mean estimation error of relative stereo camera position over varied camera baselines[4].

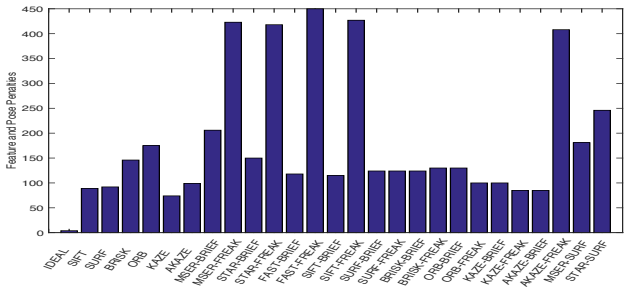


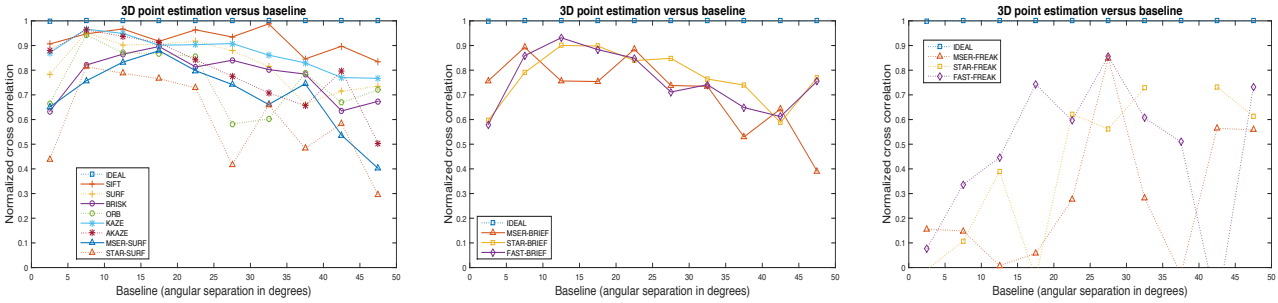
Figure 11: Penalty values for all feature extractors.

to rotation at low baselines (figure 6(a)). In figure 11, we see that IDEAL has about 4 penalties, and these are at very low baseline. This confirms that pose estimation algortihm has limitation at very low baselines. This sensitivity does not affect our comparative study on feature extractors as the penalty scored sample is considered invalid. However, we will use the penalty score to define the success rate or reliability of the feature extractor in further sections.

4.3 3D estimation error

Using the feature correspondences and the recovered camera pose, the corresponding 3D points are estimated and are compared to their ground truth values. The resulting samples are filtered based on the penalty score (described in section 4.2.1). Only the samples that are not penalized are considered valid and are used for further evaluation. The resulting 3D estimation error is plotted against varied baselines as shown in figure 8. In this figure, the 3D accuracy, expressed as normalized correlation coefficient (η), tends to reduce as the baseline of the camera increases. The error in camera pose propagates to 3D accuracy. 3D estimation is conceptually, the point of intersection of two rays back projected from a pair of feature matches. The back projection is carried out using the camera instrinsic and extrinsic (position and orientation) parameters. While camera instrinsics are maintained the same for the stereo pairs, the change in pose affects the 3D accuracy, i.e., lower the camera pose error, higher is the 3D accuracy. This is why, markerless pose estimation becomes important in 3D applications.

Figure 8(a) shows the performance of feature detectors using their own descriptors (SIFT, SURF, BRISK, ORB, KAZE, AKAZE). To compare the performance when other type of descriptors are used, we have evaluated each of these detectors with BRIEF and FREAK descriptors and the results are shown in figure 10. We have also evaluated other detectors such as MSER, STAR and FAST, which do not have their own descriptors, but using BRIEF and FREAK descriptors as shown in figures 8(b) and 8(c). In figure 8(a), we also include MSER and STAR detectors but with SURF descriptor, because they are evaluated based on SURF descriptor in [18] and [19], respectively. All the above mentioned feature extractor combinations are evaluated based of mean value of η over every 5 degrees, and there respective variances are shown in figure 9.



(a) Own Descriptors (b) Brief Descriptor (c) Freak Descriptor

Figure 8: Mean 3D estimation error (normalized correlation co-efficient) over varied camera baselines[4].

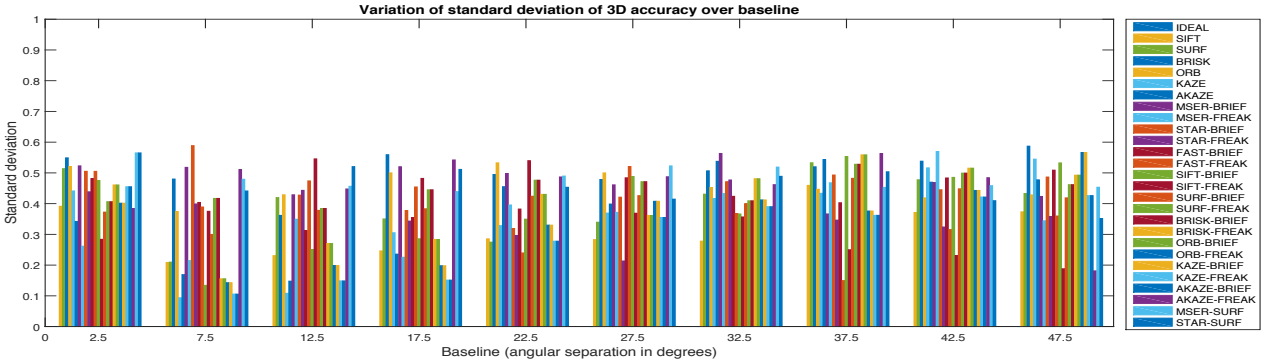


Figure 9: Standard deviation of 3D estimation over varied baselines[4].

The quality of feature extractors affect the 3D accuracy, but to what extent and how robust are they to large baselines, is what needs to be evaluated. Hence, we study the behavior and limitations of various feature extractors, especially for varied baselines. We shall now evaluate the performance of the feature extractors based on normalized cross correlation and discuss their application traits in terms of 3D mean squared errors, computation time and reliability.

4.4 Performance evaluation

At a very low baselines (less than $\approx 5^o$), the feature extractors seems to not perform very well. As explained before the sensitivity of pose estimation algorithm plays a role here. However, at very small baselines, even a small deviation in the accuracy of feature correspondences yields a large pose estimation error and thereby triangulation errors.

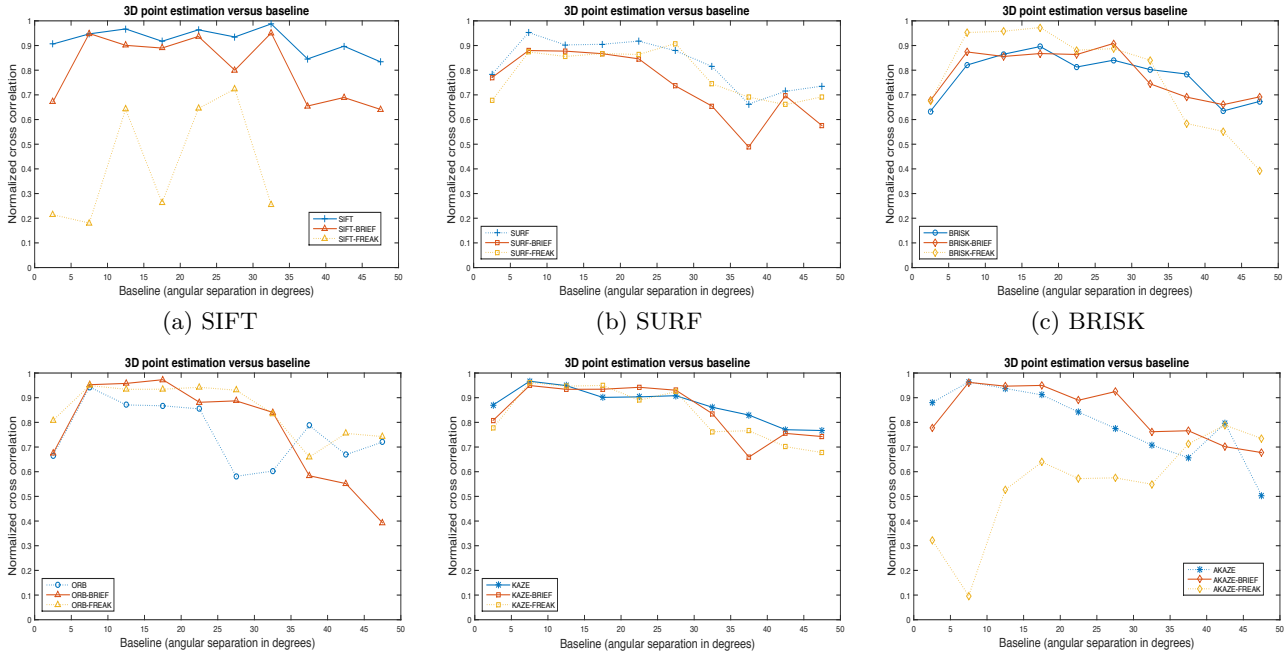
From figure 8(a), we can observe that KAZE (detector with its own descriptor), outperforms all other feature extractor combinations upto $\approx 20^o$. SIFT performs close to KAZE upto $\approx 20^o$, and thereafter outperforms KAZE at higher baselines. However, KAZE and SIFT both perform better than other feature extractors. The detectors KAZE and SIFT differ in the scale space representation, while the descriptor remains the same. As it is claimed in [16], KAZE performs as good as SIFT. However, it holds good only upto a limit specified.

The SURF and the ORB perform with almost equal accuracy upto $\approx 20^o$ baseline, and then SURF maintains the accuracy much better than ORB. Correspondingly, figures 6(a) and 7(a) show how the rotational and translational error of ORB increases after $\approx 20^o$ baseline and stays higher than SURF. This is probably because the modified BRIEF descriptor used in ORB is not as efficient as SURF descriptor, which is based on Haar wavelets, in terms of rotational invariance for higher baselines. ORB claims to be an alternative to SURF in [14], but we see that after the specified baseline limit, ORB cannot perform better than SURF.

Although AKAZE is shown to have better performance over other detectors (in [17]), we see that AKAZE performs as good as KAZE upto $\approx 20^o$ baseline and then, the performance drops down severely. Pose estimation error shows the same trend (figures 6(a) and 7(a)). However, by using AKAZE the computation time reduces comparatively.

The BRISK performs as good as ORB upto $\approx 20^o$ baseline, then seems to outperform ORB thereafter. The detectors BRISK and ORB are designed with a motivation to reduce computation time, but we notice that it is at the cost of reduction in 3D accuracy.

The MSER and STAR detectors have been evaluated using SURF descriptor in their original work. Therefore we intended to use these combinations as well. However, it seems that SURF descriptor is better off with its own detector rather than MSER or STAR. From figure 6(a), we can see that rotational errors are more prominent for MSER and STAR in combination with SURF descriptor. So, comparatively, SURF detector seems better than MSER and STAR.



(a) SIFT (b) SURF (c) BRISK

(d) ORB (e) KAZE (f) AKAZE

Figure 10: Mean 3D estimation error with varied baselines[4].

From figures 8(b) and 8(c), we can observe that MSER, STAR and FAST detectors perform with almost similar accuracy with two individual descriptors, BRIEF and FREAK. However, BRIEF descriptor seems to be well suited for these detectors compared to the FREAK descriptor. With BRIEF descriptor, STAR and FAST seem to perform with a similar pattern as in SURF atleast upto $\approx 25^o$, while BRIEF with MSER detector seems to match ORB, especially between $\approx 25^o - 40^o$ baselines and thereafter degrades. BRIEF descriptor is claimed to be as good as SURF descriptor in [21] and a modified BRIEF is used in ORB, and hence the similar performance pattern. The STAR detector seems to be better with BRIEF than SURF descriptor. The MSER detector with BRIEF and SURF descriptor shows similar performance pattern, however, SURF descriptor creation is faster than BRIEF.

On the other hand, all three detectors with FREAK descriptor in figure 8(c) seems to perform worse compared to the rest. From these observations, it is hard to generalize the behavior of BRIEF and FREAK descriptors when it is combined only with MSER, STAR and FAST detectors. So, we extended the descriptor evaluation with other detectors which basically have their own descriptors defined. Consequently, the respective results are shown in figure 10.

Feature extractors, such as SIFT and AKAZE, using the BRIEF descriptor (figures 10(a) & 10(f), respectively), maintain their accuracy similar to that of using their own descriptors, upto $\approx 35^o$ baseline. Moreover, using BRIEF descriptor is advantageous in terms of computation time.

The accuracy of SURF and KAZE stays almost the same when used with both BRIEF and FREAK descriptors as shown in figures 10(b) and 10(e), but again only upto $\approx 25^o$ baseline. So the possibility of making a choice of descriptor is higher for these detectors.

In case of BRISK and ORB, as shown in figures 10(c) and 10(d), both BRIEF and FREAK descriptors performs better than their own descriptor upto $\approx 35^o$. So, the BRIEF descriptor seems more robust to baseline changes than the modified BRIEF (used in ORB) and the BRISK descriptor.

So, the BRIEF descriptor seems to be a good choice in combination with BRISK, ORB, KAZE and AKAZE detectors for upto $\approx 35^o$ baseline. And, the FREAK descriptor is seemingly a good choice for BRISK and ORB for upto $\approx 35^o$. Moreover, FREAK descriptor could be the best choice for SURF and KAZE detectors, whose performance is comparable to SIFT and KAZE with their own descriptors.

Overall, some of the feature extractors have outperformed others and some of the descriptors have shown better performance when combined with certain detectors over others. The important aspect to notice here is that each feature extractor has performance relatively better in certain baseline range. From the evaluation of the state-of-art feature extractors, we can summarize the observed as follows:

- For baselines ($<5^o$):
  $SIFT$, $KAZE$ and $AKAZE$ seem to be good performers, however rotation-translation ambiguity exists.
- For baselines ($5^o - 30^o$):
  $SIFT$, $SURF$ and $KAZE$ with their own descriptors; $BRIEF$ descriptor with all detectors except $MSER$, $STAR$ and $FAST$; $FREAK$ descriptor with $SURF$, $BRISK$, ORB and $KAZE$ are good performers.
- For baselines ($>30^o$):
  $SIFT$ and $KAZE$ perform better than others. However, $SURF$ detector with both $SURF$ and $FREAK$ descriptors; $BRIEF$ descriptor with $BRISK$, $KAZE$ and $AKAZE$ are the next candidates.

4.5 Design recommendation

Although η gives a relative performance measure of feature extractors, it is difficult to use this information directly for practical applications. For making a sensible choice of feature extractors for a specific 3D application, feature extractors need an absolute measure that gives a sense of quality of service (QoS). The QoS depends of the type of application and its requirements. We therefore provide an extension to our evaluation of features based on QoS. We represent QoS in terms Mean Squared Error (MSE) of reconstructed 3D point positions & orientations, and reliability & computation time of the feature extractors.

The comparative observation of accuracy between feature evaluation based on η also holds good in the case of MSEs in most of the cases. However, one should not expect a direct relation because η measures the similarity and MSE measures euclidean distance between estimated and ground truth 3D points, at different baseline ranges.

Our ground truth data is represented as three unit vectors originating from the geometric center of the model. The positional and rotational changes in the 3D reconstructed points are computed as the deviations from the ground truth 3D points. This gives an idea of how the reconstructed 3D structure would be transformed in 3D space, due to the errors in feature based calibration, i.e., camera pose estimation. The reconstructed 3D points are observed to maintain the orthogonality of the 3D unit vectors randomly over various models tested under various baselines. This is because pose estimation algorithm [28] along with singular value decomposition does not yield perfect solution when singularities are present. However, this limitation of the pose estimator has a potential for further investigation, and is not in the scope of this paper.

The table 2 provides an overview of statistics of MSE of 3D points, for three categories of baseline ranges - *Small* (5^o-20^o), *Medium* (20^o-35^o) and *Large* (35^o-50^o). The MSE is expressed in the 3D model units for positional deviation and in degrees for rotational deviation. The table also specifies the computation time required by the feature extractors, which is relevant information for real-time applications.

As explained in section 4.2.1, we have filtered the invalid data occurred during pose estimation and noted down the penalties. These penalties correspond to the success rate of the feature extractor over several samples on all baseline ranges. Therefore, we use the penalties to represent the "Reliability" of the feature extractor, which shows the probability of success over 450 samples. This parameter is also reflected in table 2. The comparisons made so far in relation to η or MSE is at the cost of reliability of every feature extractor. Hence, the reliability parameter in the table becomes very important apart from accuracy and computation time, in making a choice of feature extractor.

The result shown in the table is useful for any 3D application, which uses markerless camera pose estimation. Some relevant applications for discussion are the AR applications such as head mount display systems [1, 2], mobile applications [3, 4, 5], interactive systems [6, 7] and free view rendering application such as [8]. All these applications rely on markerless camera pose estimation, where the accuracy of the camera pose estimated becomes really important. Some applications demand real-time performance as well. The camera placements vary from small to large baseline range in these applications. Hence, our study of feature extractors and their evaluation based on various baselines for 3D error in terms of position and orientation is very helpful for such applications.

Let us consider an application scenario using *Small* baseline range and a feature extractor is required to be chosen. From the table, both KAZE and AKAZE have good accuracy in terms of 3D position and rotation, but one may choose AKAZE if the application demands fast computation time. However, this choice is at the cost of reliability, because KAZE seems to be more reliable than AKAZE. On the other hand, AKAZE+BRIEF offers accuracy similar to KAZE and is equally reliable, moreover, much faster than KAZE. So, in this case, the application could choose AKAZE+BRIEF.

Now, let us consider another application, where number of cameras around an object needs to be determined using KAZE (assuming KAZE is chosen for its high reliability). Here, KAZE offers the best positional accuracy at *Medium* baseline range. Say, if we consider a baseline of about 30^o, then number of cameras required to capture an object in 360^o, is about 12. On the other hand, if one can compromise on the positional accuracy slightly, at the same time gain higher rotational accuracy, one would choose to operate with KAZE at *Large* baseline range. In this case, for a baseline of about 45^o, one could capture the same object with only 8 cameras, which is more cost effective for applications.

In this way, table 2 can be used as a recommendation for practical 3D applications, where one can either choose feature extractors or estimate the camera density around the object of interest, based on the desired quality of service.

5. CONCLUSION

In this paper, we focused on stereo vision for 3D applications such as AR and free-view rendering, where the accuracy of position and orientation of 3D points play an important role. This paper is motivated by claiming that low 2D pixel error does not guarantee good 3D accuracy, however, 3D accuracy is dependent on the quality of feature based calibration (FBC). One of the major factors determining the quality of FBC is the camera baseline.

We designed an experiment to evaluate 26 feature extractor combination and discussed the comparative study of feature extractors over 50 camera baselines. We observed that each of the feature extractors had a certain operating range for various baseline range. However, the performance of SIFT and KAZE seemed promising, in terms of accuracy and robustness to large camera baselines.

Finally, we provided a recommendation for practical 3D applications, as in table 2, which specifies quality of service in terms of 3D position & orientation accuracy of reconstructed 3D points and computation time & reliability of feature extractors. This information is very useful for the 3D application designers, which will enable them to:

1. Select the feature extractor based on an acceptable accuracy or an acceptable execution time, with a cost of reliability.

2. Decide the camera density required to capture an object of interest, for a desired quality of service.

We believe that the system built for the movie production scenario (POPART), will benefit from our recommendations, by gaining the ability to preview integrated scene

Feature extractors	**Baseline**($5^o - 20^o$)		**Baseline**($20^o - 35^o$)		**Baseline**($35^o - 50^o$)		**Time**	**Relia-bility**
	Rotation [degrees]	***Position*** [model]	***Rotation*** [degrees]	***Position*** [model]	***Rotation*** [degrees]	***Position*** [model]	[seconds]	[percent]
	mean(deviance)		mean(deviance)		mean(deviance)			
IDEAL	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00	99.11
SIFT	13.09 (7.17)	8.23 (1.96)	2.14 (1.06)	2.83 (2.64)	2.64 (0.56)	4.22 (1.11)	17.34	80.22
SURF	15.58 (5.94)	12.04 (2.26)	5.59 (0.64)	6.27 (0.30)	3.63 (0.89)	5.33 (0.80)	5.47	79.56
BRISK	20.21 (8.94)	25.31 (13.48)	6.43 (2.41)	18.73 (13.03)	3.82 (0.52)	88.69 (141.00)	1.75	67.56
ORB	21.04 (9.31)	9.41 (0.47)	8.29 (1.54)	33.30 (41.34)	3.93 (0.05)	9.22 (6.91)	0.85	61.11
KAZE	12.12 (3.84)	7.76 (2.23)	4.78 (1.25)	6.34 (1.64)	2.92 (0.26)	7.27 (2.13)	27.67	83.56
AKAZE	11.68 (2.76)	6.91 (3.97)	7.51 (0.74)	12.36 (5.03)	4.61 (1.22)	14.48 (13.06)	4.96	78.00
MSER-SURF	19.95 (11.36)	261.94 (422.71)	8.12 (1.33)	20.55 (10.17)	5.10 (0.19)	16.09 (12.27)	7.55	59.78
STAR-SURF	29.67 (12.08)	53.51 (47.55)	11.34 (3.75)	16.08 (7.11)	7.08 (0.98)	22.15 (8.20)	0.75	45.33
MSER-BRIEF	24.01 (4.87)	44.64 (45.12)	7.03 (1.17)	7.86 (3.64)	6.25 (1.77)	8.88 (5.47)	2.50	54.22
STAR-BRIEF	21.30 (11.42)	154.87 (254.05)	6.52 (0.22)	7.58 (1.61)	4.99 (0.89)	6.18 (3.66)	0.65	66.67
FAST-BRIEF	18.00 (9.78)	24.70 (9.83)	7.37 (1.36)	8.65 (2.04)	5.12 (1.48)	9.53 (4.65)	4.73	73.78
SIFT-BRIEF	14.00 (2.80)	33.85 (27.00)	4.98 (1.65)	7.82 (7.56)	4.81 (0.49)	7.68 (0.54)	7.75	74.44
SURF-BRIEF	18.70 (7.20)	16.40 (3.54)	8.93 (1.07)	270.54 (446.17)	5.63 (1.25)	11.04 (3.82)	3.22	72.44
BRISK-BRIEF	20.10 (8.38)	21.75 (17.02)	6.46 (1.57)	22.14 (25.46)	4.91 (1.00)	49.14 (46.38)	3.76	72.44
ORB-BRIEF	15.70 (10.42)	4.79 (1.70)	4.84 (0.42)	7.61 (4.39)	4.41 (0.46)	59.16 (80.32)	0.80	71.11
KAZE-BRIEF	13.63 (7.43)	38.77 (47.78)	4.33 (0.43)	4.64 (0.34)	4.18 (0.96)	16.78 (10.96)	21.12	77.78
AKAZE-BRIEF	12.86 (6.91)	10.07 (6.27)	5.37 (2.04)	16.56 (10.51)	4.45 (0.20)	8.52 (1.14)	4.48	81.11
MSER-FREAK	61.67 (4.78)	60.57 (48.24)	15.67 (9.97)	2.77 (2.04)	8.72 (3.38)	11.38 (18.21)	7.29	6.00
STAR-FREAK	52.15 (29.05)	9.95 (6.09)	15.82 (6.92)	1.49 (0.29)	4.43 (0.00)	0.74 (0.11)	1.13	7.11
FAST-FREAK	52.70 (21.19)	8.42 (9.06)	9.50 (3.29)	23.62 (39.51)	6.38 (3.73)	11.14 (17.77)	6.09	0.00
SIFT-FREAK	51.65 (11.97)	5.74 (5.23)	22.41 (14.61)	30.14 (50.25)	10.78 (0.00)	3.78 (0.00)	9.29	5.11
SURF-FREAK	20.10 (8.38)	21.75 (17.02)	6.46 (1.57)	22.14 (25.46)	4.91 (1.00)	49.14 (46.38)	3.23	72.44
BRISK-FREAK	15.70 (10.42)	4.79 (1.70)	4.84 (0.42)	7.61 (4.39)	4.41 (0.46)	59.16 (80.32)	1.21	71.11
ORB-FREAK	13.63 (7.43)	38.77 (47.78)	4.33 (0.43)	4.64 (0.34)	4.18 (0.96)	16.78 (10.96)	21.10	77.78
KAZE-FREAK	12.86 (6.91)	10.07 (6.27)	5.37 (2.04)	16.56 (10.51)	4.45 (0.20)	8.52 (1.14)	7.88	81.11
AKAZE-FREAK	53.24 (25.05)	28.34 (37.48)	14.56 (8.09)	4.46 (5.23)	6.85 (0.15)	5.16 (5.38)	9.13	9.33

Table 2: Practical recommendation for 3D applications. [Here "Rotation" is the mean 3D rotational change (expressed in degrees) and "Position" is the mean 3D positional shift (expressed in model units), of all the estimation 3D unit vectors that represent a model in 3D space.]

more accurately in real time or decide better camera positions, and thereby ease their post-production tasks.

In the future, we would like to continue to explore the factors affecting the quality of camera pose estimation, especially the spatial distribution of feature correspondences in the stereo pair and also, evaluate the feature extractors for their invariance to illumination changes. It could also be interesting to study the limitations of the pose estimation algorithms, in general.

6. REFERENCES

[1] Chunrong Yuan. Markerless pose tracking for augmented reality. In *Proc. of ISVC*, pages 721–730, 2006.

[2] Miguel Ribo, Axel Pinz, and Anton L. Fuhrmann. A new optical tracking system for virtual and augmented reality applications. In *Proc. of IEEE - IMTC*, pages 1932–1936, 2001.

[3] Stéphane Bres and Bruno Tellez. Localisation and augmented reality for mobile applications in cultural heritage . In *Proc. of Workshop 3D ARCH*, 2009.

[4] Victor Fragoso, Steffen Gauglitz, Shane Zamora, Jim Kleban, and Matthew Turk. TranslatAR: A mobile augmented reality translator. In *Proc. of IEEE - WACV*, pages 497–502, 2011.

[5] Jonathan Ventura and Tobias Höllerer. Wide-area scene mapping for mobile visual tracking. In *Proc. of ISMAR*, pages 3–12, 2012.

[6] João Paulo Lima, Francisco Simões, Lucas Figueiredo, and Judith Kelner. Model based markerless 3d tracking applied to augmented reality. *Journal on 3D Interactive Systems*, 1, 2010.

[7] Hideyuki Suenaga, Huy Hoang Tran, Hongen Liao, Ken Masamune, Takeyoshi Dohi, Kazuto Hoshi, and Tsuyoshi Takato. Vision-based markerless registration using stereo vision and an augmented reality surgical navigation system: a pilot study. *BMC Medical Imaging*, 15(1):1–11, 2015.

[8] DongBo Min, Donghyun Kim, SangUn Yun, and Kwanghoon Sohn. 2d/3d freeview video generation for 3dtv system. *Signal Processing: Image Communication*, 24(1-2):31–48, 2009.

[9] Chris Aimone, James Fung, and Steve Mann. An eyetap video-based featureless projective motion estimation assisted by gyroscopic tracking for wearable computer mediated reality. *Springer - Personal and Ubiquitous Computing*, 7(5):236–248, 2003.

[10] Andrew Hartley and Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006.

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *ACM - Communications*, 24(6):381–395, 1981.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc J. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[14] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proc. of IEEE - ICCV*, pages 2564–2571, 2011.

[15] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: binary robust invariant scalable keypoints. In *Proc. of IEEE - ICCV*, pages 2548–2555, 2011.

[16] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE features. In *Proc. of ECCV (Part VI)*, pages 214–227, 2012.

[17] Pablo Fernández Alcantarilla, Jesus Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proc. of BMVC*, 2013.

[18] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of BMVC*, pages 1–10, 2002.

[19] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Proc. of ECCV (Part IV)*, pages 102–115, 2008.

[20] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. of ECCV (part I)*, pages 430–443, 2006.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: binary robust independent elementary features. In *Proc. of ECCV (Part IV)*, pages 778–792, 2010.

[22] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: fast retina keypoint. In *Proc. of IEEE - CVPR*, pages 510–517, 2012.

[23] Endre Juhász, Attila Tanács, and Zoltan Kato. Evaluation of point matching methods for wide-baseline stereo correspondence on mobile platforms. In *Proc. of ISPA*, pages 813–818, 2013.

[24] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *Int. J. Comput. Vision*, 94(3):335–360, 2011.

[25] Michael Yin Yang, Yanpeng Cao, and John McDonald. Fusion of camera images and laser scans for wide baseline 3d scene alignment in urban environments. *Journal of Photogrammetry and Remote Sensing*, 66(6):S52–S61, 2011.

[26] Florian Shkurti, Ioannis Rekleitis, and Gregory Dudek. Feature tracking evaluation for pose estimation in underwater environments. In *Proc. of CRV*, pages 160–167, 2011.

[27] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3d objects. *Interntaional Journla of Computer Vision*, 73(3):263–284, 2007.

[28] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions - Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.