Vision-based Pose Estimation for Augmented Reality : Comparison Study

Hayet Belghit, Abdelkader Bellarbi, Nadia Zenati, Samir Otmane

Vision-based Pose Estimation for Augmented Reality : Comparison Study

Hayet Belghit[1], Abdelkader Bellarbi[2], Nadia Zenati[3], Samir Otmane[4]
[1,2,3] CDTA, Baba Hassen, Algiers, Algeria.
[4] IBISC, Univ Evry, Université Paris-Saclay, 91025, Evry, France.

[1] hbelghit@cdta.dz, [2] abellarbi@cdta.dz, [3] nzenati@cdta.dz, [4] samir.otmane@ibisc.univ-evry.fr

***Abstract*—Augmented reality aims to enrich our real world by inserting 3D virtual objects. In order to accomplish this goal, it is important that virtual elements are rendered and aligned in the real scene in an accurate and visually acceptable way. The solution of this problem can be related to a pose estimation and 3D camera localization. This paper presents a survey on different approaches of 3D pose estimation in augmented reality and gives classification of key-points-based techniques. The study given in this paper may help both developers and researchers in the field of augmented reality.**

Keywords—Augmented Reality, Computer Vision, Pose Estimation, Descriptors.

I. INTRODUCTION

Augmented reality (AR) is the technology that enhance human's real-world perception with computer generated elements by superimposing the virtual world on the real world.

The first AR interface was developed by Sutherland in the1960's [1]. However, in 1992, Thomas Caudell and David Mizell [2] used the term of augmented reality to describe a semi-transparent helmet, used by aeronautical electricians and visualizing virtual information on real images. Nowadays, AR is becoming popular, and it is used in many applications [3][4][5][6].

A lot of definitions have been then given to Augmented Reality. Each one defined it according to a specific aspect [7][8][9][10]. However, most of these definitions mentioned that to ensure a coherent AR system, we have to align the virtual and the real world which amounts to estimate the pose of the real camera. Thus, this issue has attracted a large scientific community. Therefore, many types of sensors have been considered: mechanical, ultrasound, magnetic, inertial, GPS, compass, gyroscope, and accelerometer. Nevertheless, the camera is the most used one.

A lot of researches have been conducted in this field. However, a few many reviews and surveys have been done in order to list and classify the proposed techniques. Teichrieb et al. [11] presented a review on online monocular marker-less augmented reality, dividing the approaches into two categories: model based (edge based, optical flow based and texture based) and structure from motion based (real time SFM, Mono SLAM). More recently, Marchand et al. [12] presented a survey on augmented reality describing the mathematical aspect of pose estimation techniques.

A survey of mobile AR is presented in [13] that describes the latest technologies and methods to improve runtime performance and energy efficiency for practical implementation. In the same context, we can find the history of mobile augmented reality in [14]. Rabbi and Ullah [15] presented a survey on AR challenges and tracking techniques.

The aim of this paper is to provide a technical classification of most of approaches for 3D pose estimation, we also focus on key-points-based techniques and present a reach comparison of both detectors and descriptors of the state of the art. The study given in this paper may help developers and researchers in the field of augmented reality.

The remain of this paper is organized as follow: section 2 describe the AR principle, section 3 presents the pose estimation techniques according to available data (3D or 2D), section 4 is dealing with features detection and description techniques, here we present a comparison considering computing time, recognition rate and memory space, finally we give a brief conclusion of this work.

II. AUGMENTED REALITY PRINCIPLE

In order to achieve a coherent augmented scene, that combines both virtual and real worlds, we have to align the real and the virtual cameras. In other words, we have to assign to the virtual camera, the same properties (extrinsic and intrinsic) as those of the real camera. Thus, we need to determine in real time the position and the orientation of the camera for each frame in the real scene. The following figure (Fig. 1) illustrates the 2D-3D registration problem.

Let R_w, R_C, R_{vw}, R_{vc} and R_i respectively represent the real-world landmark, the camera landmark, the virtual world landmark, the virtual camera landmark and the image landmark. In order to get a coherent composition of the real and virtual world, the two real and virtual cameras should have the same position and the same parameters (focal, field of view (FOV), etc.) according to the reference points of the two real and virtual worlds (R_w, R_{vw}). Hence, the only unknown is the pose of the real camera relatively to the real-world landmark.

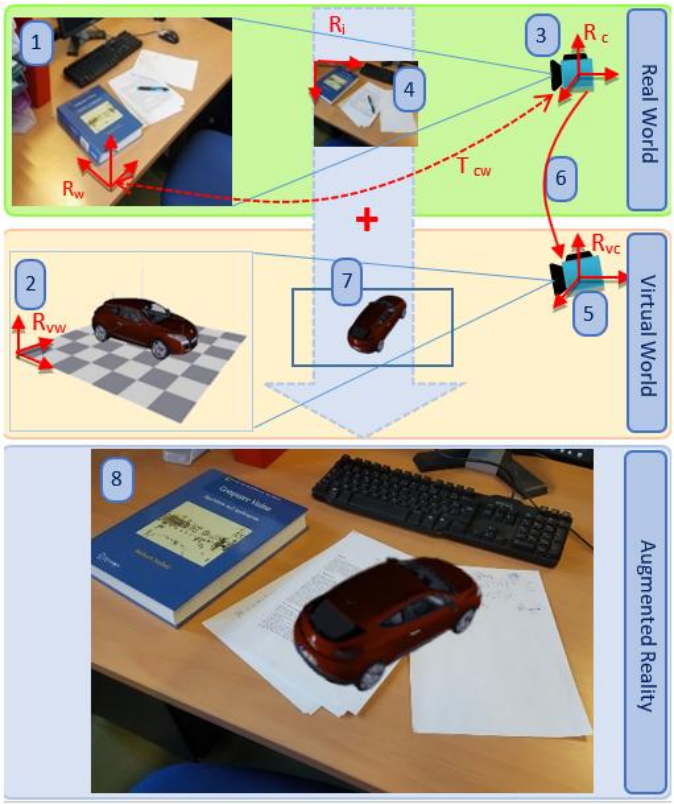

Fig. 1. General principle of registration process in augmented reality. 1) Real environment. 2) Virtual environment. 3) Camera. 4) Captured image. 5) Virtual camera. 6) Alignment of the virtual camera with the actual camera. 7) Projection space. 8) Augmented reality.

Let P be a point in the real world coordinate $(X_w, Y_w, Z_w)^T$ in R_w and $(X_c, Y_c, Z_c)^T$ in R_c, the transformation from R_w to R_c is described as follows [16] (1):

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = r \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t = (r\ t) \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad \ldots\ldots\ldots (1)$$

Where (r t) represents the transformation between the two landmarks (world and camera). This is defined by the translation vector (t) and the rotation matrix (r) of R_w to R_c. Let Q be the perspective projection of P on the image plane. The coordinates of this projection can be calculated as follows [16] (2):

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{A} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = A \underbrace{(r\ t)}_{T} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \ldots (2)$$

Where "A" is the matrix of intrinsic parameters (α_u, α_v: the ratio between the focal length and the horizontal and vertical size of the pixel, u_0, v_0: the intersection of the optical axis with the image plane) and T the matrix of extrinsic parameters. We assume that "A" is known, so that we obtain the following equation (3):

$$q = A^{-1}Q = TP \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

As mentioned, in order to insert a virtual object in a real scene in a coherent way, we have to know the pose of the camera that we represent here by the matrix "T". Thus, if we have a set of points $P_i(X_i, Y_i, Z_i)$ and their projections $q_i(x_i, y_i)$, we can determine the transformation T.

We present in the following the different approaches that determine the pose of the camera, or in other words, to solve the following equation (4):

$$q_i = TP_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

III. Pose Estimation in AR

We illustrate in Figure 2 the different approaches allowing the pose estimation according to the available data (3D or 2D), from the P-nP problem to the SLAM [11], in addition to the planar scene. Fig. 2 presents a classification of pose estimation techniques.

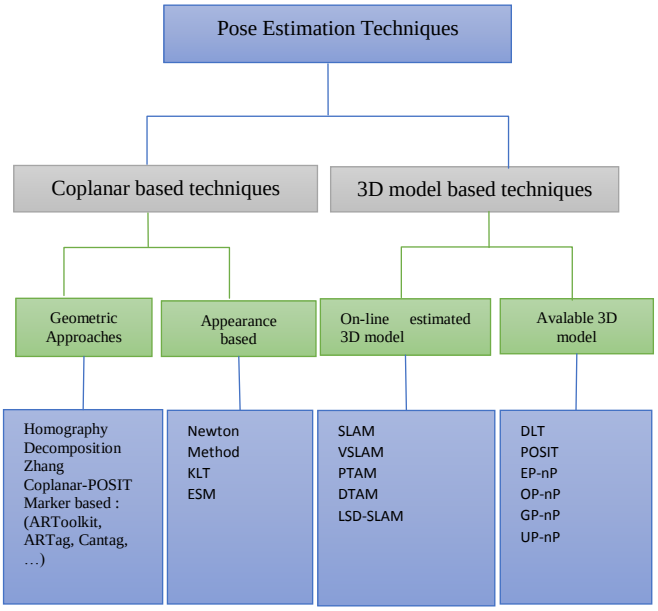

Fig. **2.** Classification of the different approaches for 3D pose estimation in AR.

A. Pose Estimation based on 3D model

3D Pose can be estimated using a minimum of 3 points. Indeed, the pose can be represented by six parameters (3 angles of rotation and 3 translations), therefore 3 points would be sufficient to solve the equation (4), which corresponds to the problem P-3P (Perspective 3 Points).

Solving this problem comes down to:

- First, we estimate the Z_{ic} for each point with respect to the R_c reference via the cosine law theorem [17] using the triangle CP_iP_j (with C is the origin of the reference R_c).
- Second, we estimate the transformation T which makes it possible to carry out the passage from R_m to R_c.

As a second alternative, the least squares method can be used. It gives an ambiguous solution and requires a fourth point to have a unique solution. This one is based mainly on the singular value decomposition SVD [12].

Kneip and al. [18] proposed a new solution to P-3P problem which calculates T directly in one step, without estimating the coordinates of the points with respect to the reference of the camera R_c. This is made possible by introducing the camera landmark $R_{c'}$ and the world landmark $R_{m'}$. Therefore, the projection of points from $R_{m'}$ to $R_{c'}$ reduces the problem to two conditions.

Although, P-3P approaches give solutions to pose estimation problem, but P-nP approaches give more accurate results by using more points. Quan and Lan [19] extended their P-3P algorithm to P-4P and then P-5P to finally reach P-nP. In the EP-nP approach [20], 3D point coordinates are expressed as a weighted sum of four virtual control points. The pose problem is then reduced to the estimation of the coordinates of these control points in the camera reference. This approach reduces computational complexity.

Direct Linear Transformation (DLT) is certainly the oldest one-step approaches [21]. Although not very accurate, this solution and its derivatives have largely been taken into account in AR applications.

P-nP is a non-linear problem. Among the solutions that deal with the non-linearity of the system, POSIT is an iterative approach proposed by Dementhon and Davis [22], the main idea consists on the use of an orthogonal projection system so that the problem becomes linear, then iteratively we return to the basic perspective projection system.

When N increases considerably, there is no solution with linear complexity for the problem P-nP. Possible solutions for this case include EPnP [20], OPnP [23], GPnP [24], and [25].

Other methods are based on tracking a model for pose estimation. The idea is to define a distance between the point of a contour in the image and the projection of the 3D line corresponding to the 3D model. The pose is estimated by minimizing the error between the selected points and the projected contours (Fig 3).

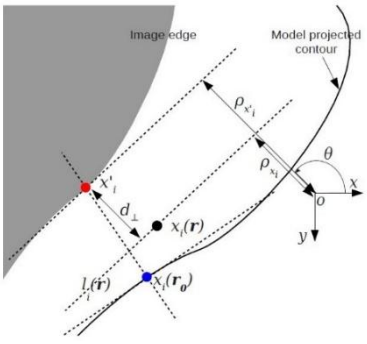

Fig. 3. Contour tracking for the pose estimation, extracted from [26]. From the initial pose r_0, a one-dimensional search is performed along the normal to the projected contour underlying the measurement point x_i (r_0). And minimizing the distance d between the point x'_i and the line $l_i(r)$.

Comport and associates [27] have proposed a tracking algorithm based on a 3D model. A nonlinear estimation of the pose is formulated using a virtual visual servoing approach.

The previous approaches are based on available 3D model. Other ones estimate at the same time the structure of the scene, and the pose of the camera. These approaches are called VSLAM (Vision based Simultaneous Localization and Mapping) [28] [29] [30].

Davison [28] used the extended Kalman filter for data integration. On the other hand, Eade and Drummond [29] used the particle filter. In such approaches, the data is sequentially integrated into the filter. The updates (camera position, speed, scene structure) are made sequentially. So, the number of estimated parameters increases with the map size.

Bundle Adjustment (BA) approaches estimate the movement of the camera by minimizing the error between the predicted points and the observed points, thus to build the map [31] [32].

Although some studies [33][31] have demonstrated the possibility of using SLAMs in AR, nevertheless this kind of approaches is lacking in terms of absolute localization and remains complex and expensive in computing time.

PTAM (Parallel Tracking and Mapping) [34] [35] dissociates tracking from mapping. Its main idea is to compute in parallel the map using BA technique and the pose using a tracking method. This approach is used in various application areas, particularly in AR [36] [37].

There are several works that are based on the use of a lot of cameras or other types of sensors, such as Kinect, like KinectFusion [38]. These kinds of systems make possible to directly determine the 3D position of the points. However, they require a slow learning and reconstruction step.

B. Pose Estimation based on coplanar informations

Considering a planar scene simplifies the pose estimation problem. It's amount to camera motion estimation process. In this section we present the approaches based on extraction of 2D information from the image and the geometric information of a planar scene. The objective is to estimate camera move between two images instead of estimating the pose; the 3D model is therefore replaced by a reference image.

1) Geometric Approaches

The objective is to estimate the camera 3D movement between two acquisitions using only the 2D information extracted from the two images. The homography between the two images is often used in this case [39], [40].

Let x_1 be a point of the image I_1 and x_2 a point of the image I_2 (I_1, I_2 two images of the same scene with a different view point). The two points (x_1, x2) are related by the homography H_1^2 as follows (5) [12]:

$$x_2 = H_1^2 x_1 \quad \text{.....(5)}$$

H_1^2 can be estimated using the Direct Linear Transformation (DLT) algorithm [21]. The pose is then calculated by a homography decomposition [41].

The simplicity of this approach made its use in AR a standard. Thus, several coded target identification systems are based on this approach. The idea is to place different markers by their colors or shapes in the real environment. Based on the fact that the markers are known a priori we can estimate the 3D camera pose.

More recently, DeTone et al. [42] have considered the homography estimation problem between two images as a learning problem. They applied a convolutional neural network (CNN) to solve it. However, this type of neural network is not relevant in term of computation time.

2) Appearance based Approaches

Considering the 2D model as a reference image, the objective is to estimate the movement between the captured image and the reference image in pixel scale. Since the model is defined by a set of pixels, we must find their new positions in the image. Instead of using homography to determine pose, alignment can be defined directly as a minimization problem of dissimilarities or maximization of similarities between the appearance of the area of interest in the reference image and the area of interest in the captured image.

For example, if the appearance is defined as the pixels intensity belonging to a patch, the dissimilarity is considered as the SSD (Sum of Squared Differences) differences.

The tracking algorithm proposed by Benhimane & Malis [43] is based on the minimization of the SSD (Sum of Squared Differences) between a given model and the current image by applying the ESM algorithm (Efficient Second Order Minimization) which has the same convergence properties as Newton's method, but with a faster computation time.

IV. FEATURES DETECTION AND DESCRIPTION

Recent advances in computer vision and the development of key-points matching methodologies make AR reaching a new level maturity. We present in the following the different techniques for image feature description.

A descriptor is a function applied to the patch in order to describe it, in an invariant way for any changes to the image (eg, rotation, lighting, noise, etc.).

The common pipeline for using descriptors is:

1. Select regions (patches) around the detected key-points in the image. These patches are square or circular shapes depending on the properties of the descriptor to be applied.
2. Describe each region (the patch) as a feature vector, using this descriptor.
3. Calculate the distance between vectors using a similarity measure.

In the state of the art, most of works focuses on the description of key-points. Those description techniques (descriptors) have been grouped into two main families, Floating Point Descriptors, and Binary Descriptors. We present in the following two comparative tables of the most known descriptors.

The first table (Table 2) illustrates the computing time of some known descriptors, we calculated the average time of a description of a patch for each of these descriptors. Thus, we noticed that the binary ones are more suitable for real-time applications (at least 15 frames per second), than the floating-point descriptors.

We note that the description is made on 500 points / frame and the number of frames per second is calculated according to the time of the description only (without adding the time of detection and matching).

TABLE 2. Description mean time of a patch (ms) and the number of frames / sec.

Descriptors	Description mean time of a patch (ms)	Number of frames per second (description of 500 points per frame)
SIFT [44]	3.121	0.64
SURF [45]	1.488	1.34
LDA-HASH [46]	4.21	0.47
BRISK [47]	0.072	27.77
FREAK [48]	0.094	21.27
ORB [49]	0.146	13.69
LDB [50]	0.139	14.38
LATCH [51]	0.437	4.57
MOBIL [52]	0.127	15.74
MOBIL_2B [53]	0.136	14.70
POLAR_MOBIL [54]	0.107	18.68

We present in Table 3, a comparison of more than 30 descriptors from the state of the art with their detectors. We have classified these descriptors according to certain criteria that we judged necessary for the implementation of an augmented reality application.

We evaluated the descriptors according to each criterion, namely *computation time*, *recognition rate* and *memory space*, using a scale from 1 to 5 (from + to +++++) shown as follows:

Computing time:	Recognition rate:	Memory:
+: Very slow.	+: Less robust.	+: Voluminous.
+++++: Very speed.	+++++: Robust.	+++++: Lightweight.

According to Table 3, we noticed that new descriptors based on deep learning such as [55], [56] and [57] give better results in terms of recognition rate. However, their major disadvantage is the calculation time, similarly to traditional floating-point descriptors such as SIFT [44], GLOH [58], LDE [59] or DAISY [60].

TABLE 3. Descriptors Comparison

Descriptor	Suggested Detector	Type	Computing time	Recognition rate	Memory space
SIFT [44]	DoG	Float	+	+++++	++
PCA-SIFT [61]	DoG (SIFT)	Float	++	++++	+++
GLOH [58]	DoG (SIFT)	Float	+	+++++	++
SURF [45]	Determinant of Hessian	Float	+	++++	++
LDE [59]	DoG	Float	++	+++++	+++
Daisy [60]	DoG (SIFT)	Float	+	+++++	++
BRIEF [62]	SURF Detector	Binary	+++	+++	++++
ORB [48]	FAST	Binary	++++	++++	++++
BRISK [47]	AGAST+FAST	Binary	+++++	++++	++++
FREAK [48]	BRISK Detector	Binary	+++++	++++	++++
ALOHA [63]	SURF Detector	Binary	+++	++++	+++
LDA-HASH [46]	DoG (SIFT)	Binary	+	+++++	+
KAZE [64]	Hessian Matrix + Scharr filter	Float	++	+++	+++
BinBoost [65]	DoG	Binary	++	++++	++
A-KAZE [66]	KAZE Detector	Binary	++++	++++	+++
LDB [50]	DoG (SIFT)	Binary	+++	++++	+++
OSRI [67]	DoG / Hessian /Harris-Affine	Binary	+++	++++	++++
USB [68]	DoG	Binary	+++	++++	++++
MOBIL [52]	FAST	Binary	++++	++++	++++
BSIFT [70]	SIFT	Binary	+++	++++	+++
BOLD [71]	Haris-Laplace	Binary	++++	++++	++++
MOBIL_2B [53]	FAST	Binary	++++	++++	++++
Deep Hashing [72]	Full Image	Binary	++	+++++	+++
LATCH [51]	Multi-scale Harris	Binary	++	++++	+++
MatchNet [55]	Convolutional neural network	Float	++	+++++	+++
[56]	Convolutional neural network	Float	+	+++++	++
3D ConvNets [57]	Convolutional neural network	Float	++	+++++	+
POLAR_MOBIL [54]	MOBIL_DETECTOR	Binary	++++	++++	++++

On the other hand, we noticed that the binary descriptors are better than the other two families in terms of memory and computing time. However, their robustness and their discriminative and distinctive powers are considerably limited. Except, some robust binary descriptors such BRISK, FREAK, POLAR_MOBIL and BOLD, which are suitable for using in real-time augmented reality applications.

V. CONCLUSION

This paper presents a global vision of the pose estimation problem used in augmented reality. We first presented the geometrical aspect of the pose estimation by presenting the different methods that make it possible to geometrically answer this problem, then we approached the approaches that integrate motion estimation. We classified these approaches according to the available information: 3D model or planar scene.

Then, we presented the extraction and description of features and we presented a comparison of different descriptors. According to the conducted comparison, we found that the recent binary descriptors are the most suitable for such augmented reality applications, thanks to their low computing time and memory consumption.

VI. REFERENCE

[1] I. E. Sutherland, « A head-mounted three-dimensional display », in Proceedings of the AFIPS '68 (Fall, part I), 1968, p. 757‑ 764.

[2] T. P. Caudell et D. W. Mizell, « Augmented reality: an application of heads-up display technology to manual manufacturing processes », in Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, 1992, vol. ii, p. 659‑ 669 vol.2.

[3] A. Bellarbi, C. Domingues, S. Otmane, S. Benbelkacem, et A. Dinis, « Underwater augmented reality game using the DOLPHYN », in Proceedings of the 18th ACM symposium on Virtual reality software and technology - VRST '12, 2012, p. 187.

[4] S. Benbelkacem, M. Belhocine, A. Bellarbi, N. Zenati-Henda, et M. Tadjine, « Augmented reality for photovoltaic pumping systems maintenance tasks », Renew. Energy, vol. 55, p. 428‑ 437, 2013.

[5] N. Zenati-Henda, A. Bellarbi, S. Benbelkacem, et M. Belhocine, « Augmented reality system based on hand gestures for remote maintenance », in International Conference on Multimedia Computing and Systems -Proceedings, 2014, p. 5‑ 8.

[6] Belghit, H., Bellarbi, A., Zenati, N., Benbelkacem, S., & Otmane, S. (2015, April). Vision-based collaborative & mobile augmented reality. In Proceedings of the 2015 Virtual Reality International Conference (p. 23). ACM.

[7] P. Milgram et F. Kishino, « Taxonomy of mixed reality visual displays », IEICE Trans. Inf. Syst., vol. E77‑ D, no 12, p. 1321‑ 1329, 1994.

[8] P. Fuchs et G. Moreau, Le traité de la réalité virtuelle. Volume 2, Les Presses de l'École des Mines, 2006.

[9] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, et B. MacIntyre, « Recent advances in augmented reality », IEEE Comput. Graph. Appl., vol. 21, no 6, p. 34‑ 47, 2001.

[10] Bellarbi, Abdelkader. "Vers l'immersion mobile en réalité augmentée : une approche basée sur le suivi robuste de cibles naturelles et sur l'interaction 3D." PhD diss., Université Paris Saclay ; Université d'Evry Val d'Essonne, 2017.

[11] Teichrieb, Veronica, et al. "A survey of online monocular markerless augmented reality." International Journal of Modeling and Simulation for the Petroleum Industry 1, no. 1 (2007).

[12] E. Marchand, H. Uchiyama, F. Spindler, E. Marchand, H. Uchiyama, et F. Spindler, « Pose estimation for augmented reality : a hands-on survey Pose estimation for augmented reality : a hands-on survey », IEEE Trans. Vis. Comput. Graph., 2016.

[13] Huang, Z., Hui, P., Peylo, C., & Chatzopoulos, D. (2013). Mobile augmented reality survey: a bottom-up approach. arXiv preprint arXiv:1309.4413.

[14] Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A., & Wagner, D. (2015). The history of mobile augmented reality. arXiv preprint arXiv:1505.01319

[15] Rabbi, I., & Ullah, S. (2013). A survey on augmented reality challenges and tracking. Acta graphica: znanstveni časopis za tiskarstvo i grafičke komunikacije, 24(1-2), 29-46.

[16] V. Lepetit, « Gestion des occultations en réalité augmentée », Thèse de doctorat, Université de Nancy, 2001.

[17] V. J. Katz, "The Mathematics of Egypt, Mesopotamia, China, India, and Islam: A Sourcebook". Princeton University Press, 2007.

[18] L. Kneip, D. Scaramuzza, et R. Siegwart, « A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation », in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011, p. 2969‑ 2976.

[19] L. Quan et Z. Lan, « Linear N-point camera pose determination », IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no 8, p. 774‑ 780, 1999.

[20] V. Lepetit, F. Moreno-Noguer, et P. Fua, « EPnP: An accurate O(n) solution to the PnP problem », Int. J. Comput. Vis., vol. 81, no 2, p. 155‑ 166, févr. 2009.

[21] R. Hartley et A. Zisserman, "Multiple View Geometry In Computer Vision", vol. 23, no 2. Cambridge University Press, 2005.

[22] D. F. Dementhon et L. S. Davis, « Model-based object pose in 25 lines of code », Int. J. Comput. Vis., vol. 15, no 1‑ 2, p. 123‑ 141, juin 1995.

[23] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, et M. Okutomi, « Revisiting the PnP problem: A fast, general and optimal solution », in Proceedings of the IEEE International Conference on Computer Vision, 2013, p. 2344‑ 2351.

[24] L. Kneip, P. Furgale, et R. Siegwart, « Using multi-camera systems in robotics: Efficient solutions to the NPnP problem », in Proceedings - IEEE International Conference on Robotics and Automation, 2013, p. 3770‑ 3776.

[25] L. Kneip, H. Li, et Y. Seo, « UPnP: An optimal O(n) solution to the absolute pose problem with universal applicability », in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8689 LNCS, no PART 1, p. 127‑ 142.

[26] M. A. Petit, M. E. Marchand, et M. K. Kanani, « Détection et suivi basé modèle pour des applications spatiales 1 Introduction » *Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'13*, Jun 2013, Cluny, France. pp.1-6, 2013.

[27] Comport, Andrew I., et al. "Real-time markerless tracking for augmented reality: the virtual visual servoing framework." IEEE Transactions on visualization and computer graphics 12.4 (2006): 615-628.

[28] A. J. Davison, « Real-time Simultaneous Localisation and Mapping with a Single Camera », ICCV, vol. 2, p. 1403‑ 1410, 2003.

[29] E. Eade et T. Drummond, « Scalable monocular SLAM », in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 1, p. 469‑ 476.

[30] M. Agrawal et K. Konolige, « FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping », IEEE Trans. Robot., vol. 24, no 5, p. 1066‑ 1077, 2008.

[31] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, et P. Cnrs, « Monocular Vision Based SLAM for Mobile Robots », 18th Int. Conf. Pattern Recognit., vol. 3, p. 1027‑ 1031, 2006.

[32] G. Sibley, C. Mei, I. Reid, et P. Newman, « Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment », Int. J. Rob. Res., vol. 29, no 8, p. 958‑ 980, juill. 2010.

[33] D. Nister et J. Bergen, « Visual Odometry », Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004., vol. 1, p. I-652-I-659 Vol.1, 2004.

[34] G. Klein et D. Murray, « Parallel tracking and mapping for small AR workspaces », in 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR, 2007, p. 1‑ 10.

[35] G. Klein et D. Murray, « Parallel tracking and mapping on a camera phone », in Science and Technology Proceedings - IEEE 2009 International Symposium on Mixed and Augmented Reality, ISMAR 2009, 2009, p. 83‑ 86.

[36] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, et D. Schmalstieg, « Pose tracking from natural features on mobile phones », in Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008, ISMAR 2008, 2008, p. 125‑ 134.

[37] VENTURA, Jonathan et HÖLLERER, Tobias. Wide-area scene mapping for mobile visual tracking. In : Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on. IEEE, 2012. p. 3-12.

[38] S. Izadi et al., « KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera », Proc. 24th Annu. ACM Symp. User Interface Softw. Technol., p. 559–568, 2011.

[39] D. Oberkampf, D. F. DeMenthon, et L. S. Davis, « Iterative Pose Estimation Using Coplanar Feature Points », Comput. Vis. Image Underst., vol. 63, no 3, p. 495‑ 511, mai 1996.

[40] Z. Zhang, « A flexible new technique for camera calibration », IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no 11, p. 1330‑ 1334, 2000.

[41] O. D. Faugeras et F. Lustman, « Motion and structure from motion in a piecewise planar environment », Int. J. Pattern Recognit. Artif. Intell., vol. 2, no 3, p. 485‑ 508, 1988.

[42] D. DeTone, T. Malisiewicz, et A. Rabinovich, « Deep Image Homography Estimation », in RSS Workshop on Limits and Potentials of Deep Learning in Robotics, 2016.

[43] S. Benhimane et E. Malis, « Real-time image-based tracking of planes using efficient second-order minimization », 2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IEEE Cat. No.04CH37566), vol. 1, p. 943‑ 948, 2004.

[44] D. Lowe, « Distinctive image features from scale-invariant keypoints », Int. J. Comput. Vis., vol. 60, no 2, p. 91‑ 110, nov. 2004.

[45] H. Bay, T. Tuytelaars, et L. Van Gool, « SURF: Speeded up robust features », in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3951 LNCS, p. 404‑ 417, 2006.

[46] C. Strecha, A. M. Bronstein, M. M. Bronstein, et P. Fua, « LDAHash: Improved matching with smaller descriptors », IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no 1, p. 66‑ 78, 2012.

[47] S. Leutenegger, M. Chli, et R. Y. Siegwart, « BRISK: Binary Robust invariant scalable keypoints », in Proceedings of the IEEE International Conference on Computer Vision ICCV, 2011, p. 2548‑ 2555.

[48] E. Rublee, V. Rabaud, K. Konolige, et G. Bradski, « ORB: an efficient alternative to SIFT or SURF », in International Conference on Computer Vision ICCV, 2011, p. 2564‑ 2571.

[49] A. Alahi, R. Ortiz, et P. Vandergheynst, « FREAK: Fast Retina Keypoint », Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., p. 510‑ 517, juin 2012.

[50] X. Yang et K. T. T. Cheng, « Local difference binary for ultrafast and distinctive feature description », IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no 1, p. 188‑ 194, 2014.

[51] G. Levi et T. Hassner, « LATCH: Learned arrangements of three patch codes », in IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016.

[52] A. Bellarbi, S. Otmane, N. Zenati, et S. Benbelkacem, « MOBIL: A moments based local binary descriptor », in IEEE International Symposium on Mixed and Augmented Reality, ISMAR, 2014, p. 251‑ 252.

[53] A. Bellarbi, N. Zenati-Henda, H. Belghit, M. Hamidia, S. Benbelkacem, et S. Otmane, « An Improved MOBIL Descriptor for Markerless Augmented Reality », in 3rd International Conference on Control, Engineering and Information Technology, CEIT 2015, 2015.

[54] BELLARBI, Abdelkader, ZENATI, Nadia, OTMANE, Samir, et al. Learning moment-based fast local binary descriptor. Journal of Electronic Imaging, 2017, vol. 26, no 2, p. 023006.

[55] Han, Xufeng, et al. "Matchnet: Unifying feature and metric learning for patch-based matching." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[56] Simo-Serra, Edgar, et al. "Discriminative learning of deep convolutional feature point descriptors." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[57] Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." Proceedings of the IEEE international conference on computer vision. 2015.

[58] K. Mikolajczyk et C. Schmid, « A performance evaluation of local descriptors », IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no 10, p. 1615‑ 1630, oct. 2005.

[59] G. Hua, M. Brown, et S. Winder, « Discriminant embedding for local image descriptors », Proc. IEEE Int. Conf. Comput. Vis., 2007.

[60] E. Tola, V. Lepetit, P. Fua, et S. Member, « DAISY: An efficient dense descriptor applied to wide-baseline stereo », IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no 5, p. 815‑ 830, mai 2010.

[61] Y. K. Y. Ke et R. Sukthankar, « PCA-SIFT: a more distinctive representation for local image descriptors », Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, 2004. CVPR 2004., vol. 2, p. 2‑ 9, 2004.

[62] M. Calonder, V. Lepetit, C. Strecha, et P. Fua, « BRIEF: Binary robust independent elementary features », Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6314 LNCS, no PART 4, p. 778‑ 792, 2010.

[63] S. Saha et V. Démoulin, « ALOHA: An efficient binary descriptor based on Haar features », in Proceedings - International Conference on Image Processing, ICIP, 2012, p. 2345‑ 2348.

[64] ALCANTARILLA, Pablo Fernández, BARTOLI, Adrien, et DAVISON, Andrew J. KAZE features. In : European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012. p. 214-227.

[65] TRZCINSKI, Tomasz, CHRISTOUDIAS, Mario, FUA, Pascal, et al. Boosting binary keypoint descriptors. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2013. p. 2874-2881.

[66] P. F. Alcantarilla, J. J. Nuevo, et A. Bartoli, « Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces », in Procedings of the British Machine Vision Conference 2013, 2013, p. 13.1-13.11.

[67] X. Xu, L. Tian, J. Feng, et J. Zhou, « OSRI: A rotationally invariant binary descriptor », IEEE Trans. Image Process., vol. 23, no 7, p. 2983‑ 2995, 2014.

[68] S. Zhang, Q. Tian, Q. Huang, W. Gao, et Y. Rui, « USB: Ultrashort binary descriptor for fast visual matching and retrieval », IEEE Trans. Image Process., vol. 23, no 8, p. 3671‑ 3683, 2014.

[69] Desai, Alok, Dah-Jye Lee, and Craig Wilson. "Using affine features for an efficient binary feature descriptor." Image Analysis and Interpretation (SSIAI), 2014 IEEE Southwest Symposium on. IEEE, 2014.

[70] ZHOU, Wengang, LI, Houqiang, HONG, Richang, et al. BSIFT: toward data-independent codebook for large scale image search. IEEE Trans. Image Processing, 2015, vol. 24, no 3, p. 967-979.

[71] V. Balntas, L. Tang, et K. Mikolajczyk, « BOLD - Binary online learned descriptor for efficient image matching », Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07‑ 12‑ June, p. 2367‑ 2375, 2015.

[72] V. E. Liong, J. Lu, G. Wang, P. Moulin, et J. Zhou, « Deep hashing for compact binary codes learning », Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07‑ 12‑ June, p. 2475‑ 2483, juin 2015.