Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 13 (2021) 101–113

DOI: 10.2478/auseme-2021-0008

Image Processing Methods for Gesture-Based Robot Control

Abdelouahab ZAATRI[1], Hamama ABOUD[2]

[1] Brothers Mentouri University Constantine1,
Department of Mechanical Engineering, Algeria,
e-mail:azaatri@yahoo.com
[2] Brothers Mentouri University Constantine1,
Department of Transportation Engineering, Algeria,
e-mail: hamamaaboud@yahoo.fr

Manuscript received February 14, 2021; revised September 20, 2021

Abstract: In this paper we discuss some image processing methods that can be used for motion recognition of human body parts such as hands or arms in order to interact with robots. This interaction is usually associated to gesture-based control. The considered image processing methods have been experienced for feature recognition in applications involving human robot interaction. They are namely: Sequential Similarity Detection Algorithm (SSDA), an appearance-based approach that uses image databases to model objects, and Kanade-Lucas-Tomasi (KLT) algorithm which is usually used for feature tracking. We illustrate the gesture-based interaction by using KLT algorithm. We discuss the adaptation of each of these methods to the context of gesture-based robot interaction and some of their related issues.

Keywords: Gesture-based control, human-robot interaction, feature recognition, movement tracking, robotics.

1. Introduction

In order to facilitate human-robot interaction and control, the recent design of various robots has fostered the emergence of several control modes. One can cite the following interaction modes: speech-based interaction, pointing on image-based interaction control, gesture-based control, myoelectric-based control, and even more recently brain-based control [1], [2].

As the many developed modes, gesture-based interaction has been studied, designed and implemented by many authors. A survey concerning gesture-based interaction is presented in [3], [4]. Technically, in most applications, gesture-

based interaction is mainly based on object recognition and tracking approaches involving CCD camera sensors. Several image processing methods have been used for image recognition such as feature-based methods, gradient-based methods, learning methods, etc. [5].

In principle, in gesture-based interaction, any part of the human body can be used for interacting with robots such as hands, arms, heads, eyes, lips, cheeks. Any tool moved by the human can also be used such as pencils, flags, sticks, etc. Applications of gesture-based robot interactions are numerous and can be adapted and extended according to many needs and contexts. Gestural expressions can be employed in regions where the speech is useless. This can happen in noisy places, in underwater areas, and in empty space where the medium cannot convey the sound waves as with astronauts [6]. It can be also used for learning by demonstration and for imitation as reproducing operations in medical care or tele-surgery [7]. This interaction mode is also interesting in some military activity to communicate and direct remote teams, autonomous and unmanned systems [8]. Moreover, it can be used in assistive robotics for supervising deaf people, for surveillance of disabled people, etc. As an example for such applications, we can cite one well known application that uses cheek movements to write and speak texts as in the case of the famous physicist Stephen Hawking [9], [11].

In addition, the gesture-based interaction can be designed with contact or without contact. Examples of gesture interaction with contact is used for teaching on blackboards, tables and other supports. In these situations, markers and colors can be employed to facilitate the tracking of elements of interest [10], [11]. However, with the advent of Covid19 pandemic, gesture-based control without contact is gaining a special importance by avoiding touch and contact with contaminated people and objects like door handles, machines, etc.

In this paper we discuss the human movement detection with applications oriented to gesture-based interactions. Some image processing methods that can serve as a core in this type of applications will be presented and discussed. We have largely applied them in object recognition and tested them with various robotics systems [11], [12], [13].

The first proposed method is a feature-based algorithm generally employed for stereo-vision to ensure correspondence between features in different images. This technique is called Sequential Similarity Detection Algorithm (SSDA) [14]. The second method belongs to recognition systems which are appearance-based. It uses a database of images to constitute models [15]. The last method uses the Kanade-Lucas-Tomasi (KLT) algorithm which belongs to gradient based trackers [16]. We will give briefly an illustration of gesture-based interaction experimented with KLT algorithm.

2. Description of some object recognition methods

This section provides a brief description of the three techniques of image processing we have experienced in object recognition and tracking of image features. We describe also the way they can be adapted and used for gesture-based interaction and control.

A. Sequential Similarity Detection Algorithm (SSDA)

SSDA is one of the basic algorithms which is typically used for 3D applications involving stereovision [14]. By stereovision, we mean that from two different images containing an object of interest, one can extract the 3D coordinates of this object in real world. In fact, the 3D information is hidden in the disparity between a same feature point observable in left and right images which are acquired from two different positions with a known displacement. Finding the corresponding points in the left and right images is the main problem in stereovision. Many techniques attempt to solve this difficult problem but with limited success depending on the context such as variation of luminosity, structure of the object, etc. [14].

Technically SSDA uses a similarity measure that is the correlation of light intensity between the neighborhoods of the right and left corresponding points situated in two images. First, a feature point in the left image is detected and a window is generated surrounding it. Its corresponding point in the right image is constrained to lie on a horizontal line of the image which is called epipolar line. Since many points of the right image can be candidates, therefore a window is generated sequentially around any candidate point. Then, the sum of light intensity difference over the windows of the left point and any candidate of the right image is computed and then squared. This sum-of-squared difference is defined as an evaluation function of the similarity between the left point and the candidate points of the right image. The best candidate that matches the left point corresponding to the minimal value of the evaluation function D, is the corresponding point. Mathematically, the algorithm determines *min(D)*[14], where:

$$D = \frac{1}{\sigma_L^2} \sum_i^n \sum_j^m \left\{ I_L(i,j) - I_R(i,j) \right\}^2 \tag{1}$$

with

$$\sigma_L^2 = \frac{1}{m.n} \sum_i^n \sum_j^m \left\{ I_L(i,j) - \mu_L \right\} \tag{2}$$

where $I_L(i, j)$ and $I_R(i, j)$ are respectively the light intensities of the pixel in left and right images situated at location (*i, j*), *n* and *m* are the width and length of the window generated around the location (*i, j*), μ_L and σ_L are respectively the mean and variance of the light intensity of the window containing the point in the left image.

We have implemented the SSDA which is a feature-based algorithm for object recognition and localization in many robotics applications [11], [12], [15]. To implement SSDA for movement detection or gesture-based interaction, we have adapted SSDA to match features from a temporal sequence of images. As we are concerned with relatively slow motion of the body part, only a very limited set of images during the movement can be used. The images are taken at a regular period *T*. Once two images separated by *T* are acquired, the SSDA is processed in order to determining the disparity in successive images. The disparity $disp(.)$ is defined as the difference between the positions of the feature point detected in right (u_{right}, v_{right}) and left (u_{left}, v_{left}) images in an image reference frame, such as:

$$disp(u) = u_{left} - u_{right} \tag{3}$$

$$disp(v) = v_{left} - v_{right} \tag{4}$$

To detect the occurrence of a movement of a body part, we do not need for this application to work in real word but it suffices to work in the image space. Therefore, we can infer the amplitude and the orientation of the motion, respectively.

This technique requires to specify in the initial static position one or more elements that constitute the model of the object to be tracked. To improve the efficiency of our system, we consider its sensitivity against detection of small movements which are not relevant and not intended as commands. These noisy movements can be caused by the imperfections of the practical used tools, by the changes in luminosity, by the normal shaking movements of the human body, etc. Errors can also be caused by the procedures that are inherent to the vision system itself. Examples are mismatches in determining corresponding points, mismatches in estimating orientations, etc.

To this end, an error detection technique has been defined based on a movement threshold value introduced to differentiate real movements from noisy ones. Then, correspondingly, threshold disparities are imposed to avoid the occurrence of noisy movements as follows:

$$disp(u) >= disp(u_{\min}) \tag{5}$$

$$disp(v) >= disp(v_{\min}) \tag{6}$$

where $u_{\min}$ and $v_{\min}$ are the thresholds values over which the movement can be considered as effective.

B. Object recognition system with an appearance based method

The existing approaches for image recognition in the literature are mainly of two types: model-based and appearance-basedmethods [15], [16], [17]. Model-based object recognition systems try to match a defined 3D model by its representation in a given image with 2D features such as lines, vertices and ellipses. Then, they try to extract from other images these features in order to recognise the objects. Appearance-based systems use the luminance information of an object. The idea is not to impose what has to be seen in the image (points, lines) but rather to use what is really seen in the image to characterise an object. In this respect, interest points are local features with high informational content. Examples are corners, T-junctions and locations where the texture varies significantly. Those interest points are on the other hand invariant with respect to geometric transformations such as rotations and translations. So that appearance-based systems learn objects by looking to them from different viewpoints and under different lighting conditions and use those images to model the object [15].

We briefly describe the recognition approach that belongs to appearance-based methods we have used and experienced. The corresponding software for object recognition which has been developed is described in [15]. This method adopts an invariant-based approach based on affinely invariant regions. Invariance is considered under affine geometric changes and linear photometric changes. This corresponds to the assumption that the scene is locally planar and not occluded and that no specular reflections occur. Precisely, the problem is as follows: given a point in two images of the same scene, taken from different viewpoints, find one or more regions around it, such that the same regions are found in both images independently, i.e. without knowledge about the other image. Correspondence between features is found by comparing affinely invariant regions. To reduce the complexity of the problem, restriction is made to finding affine invariant regions for corner points making use of the nearby edges [15].

In this implementation, the software is exploited as follows. First, a database of images containing the relevant objects (the models) is built off-line. The objects that will serve as models are placed into natural scenes. The operator moves his body part let's say his hand in a particular manner according to a given code which corresponds to a particular robot command. A set of images are captured from different views. The images are then processed in order to extract the features of the objects of interest that serve as models for the

automatic object recognition. To make this database more relevant, a region of enclosing the object of interest can be defined.

The process of feature extraction uses canny edge detector, corner detection and region finding. From the region finding invariant features concerning the model are extracted and are stored into the database. A model in the database may be represented in many scenes, under different viewpoints. The vision system takes a current image. This image is processed by the same procedures used to build the database models. The features of the objects are extracted from the current image. Then, the recognition process starts. It aims to match an object among those found in the current image to the one retrieved from the database.

C. Kanade–Lucas–Tomasi (KLT) feature tracker

The KLT algorithm is one of the most popular methods for feature tracking which was introduced by Lucas and Kanade [18] and later extended in the works of Tomasi and Kanade [19], [20]. As the SSDA, its main goal is to find corresponding features in different images. KLT makes use of spatial intensity information to direct the search for the position that yields the best match.

It uses an optical flow approach. The KLT algorithm proceeds in two steps. In the first step, it automatically detects a set of feature points in the initial image which have sufficient texture such as corners. These features are considered as best trackable in next images. In the second step, as with the SSDA, the correspondence is established by minimizing a function which is the sum of squared distances that measures the dissimilarity between a selected point and its potential corresponding one in the next image. By iterating this process sequentially between images, the features can be tracked. The final objective of the algorithm is to provide the coordinates of a feature in the sequence of images. Static features in images have a negligible displacement while moving features can be detected through their relative displacements.

KLT is a framework with the source code made available in the public domain, for both commercial and non-commercial use for the computer vision community. It is implemented in some platforms [21], [22]. The use of this software requires to specify some parameters among which the maximum number of tracked features needed for the application under consideration. This number is automatically detected in the first image if possible depending on the image structure and context. Nevertheless, in the next images, this number may decrease depending on the encountered situation since some features could be lost by occlusion, variations of luminosity, change in object orientation, change in distance of the object from the camera, etc.

Once the software is launched, the selected features are listed and colored in the initial images and in the next images the recognized features are colored as

well. This software is suitable for movement detection because the calculation of displacements of a feature in the sequence of frames enables consequently to detect the movement of the tracked object and its direction and orientation. One main advantage of the KLT is the fact that it does not need necessarily a specific model of an object to be tracked. The automatically detected features which are considered as trackable suffice to track the object of interest. Afterwards, in applications such as gestural interaction, it is primarily the movement that matters.

3. An example of application using KLT tracker

A. General structure of gesture-based robot control

The principle of gesture-based interaction systems is simple. It requires sensors that capture sequences of images during the movement of a human body part or of any tool moved by him/her. Then a software analyzes these sequences of images in order to track the movements of the elements of interest. Once a significant movement is detected and tracked; then, the final configuration of the gesture is identified. It is interpreted with respect to a code that links detected movements with corresponding robot commands. This configuration of the gesture is finally sent as a robot command corresponding to a particular action or task to be performed by the robot. *Fig. 1* summarizes the principle of gesture-based robot interaction showing the sequence of the designed and implemented general operations.

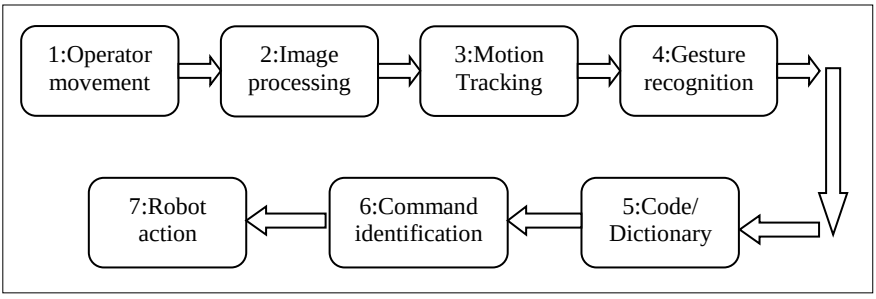

Figure 1: Sequence of involved general operations of gesture-based interaction

B. System description and experiments with KLT

Fig. 2 illustrates the general structure of one of our experimental system which was designed for gesture-based robot control and which was

implemented around KLT software [13]. On the right side of *Fig.2*, one can distinguish the human operator who is assumed to move a part of his body (hand, arm, leg, face, etc.) in front of a CCD camera. The camera captures a sequence of images which is analyzed by the KLT software to identify and track the operator's movements. On the right side of *Fig. 2*, one can distinguish a homemade serial robot manipulator that has to perform the corresponding commands according actually to human hand gestures. A virtual robot structurally shaped as the real robot in virtual environment was also developed by means of Java 3D platform. It enables to perform experimentations of the gestural-based robot control by simulations.

Many groups of experiments have been carried out with the serial robot manipulator shown in *Fig. 2*. The user was located in front of the camera up to 60 cm. The user performed a series of gestures (hand movements) in different directions. A maximum number of 30 feature points have been defined in the initial image. The image sequences are introduced as inputs to the KLT software which tracks the displacement of the identified features in successive images. The image coordinates of the tracked features (u, v) are identified. As we consider planar movements, these coordinates are converted via the well-known inverse perspective transformation into spatial coordinates (x, y). The direction of the movement is determined and consequently the robot moves its end effector to the corresponding directions. The command can be sent first for simulation to the virtual robot and then for execution to the real robot.

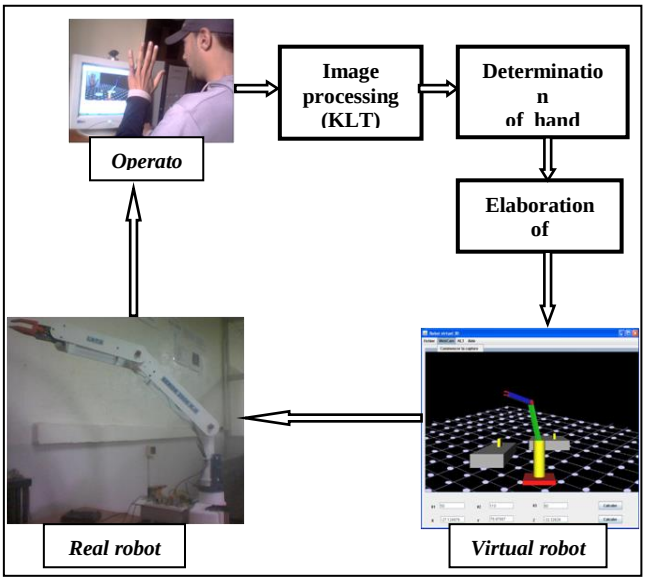

Figure 2: Experimental gesture-based system using KLT

The experiments have been carried out in two types of environments. In particular, seven groups of experiments have been performed in ordinary natural environments with different backgrounds. The environments contained a lot of objects in the background of the interesting object (user's hand) with possibly varying illumination. *Fig. 3.a* shows as an example of illustration two following images (at the left side of *Fig. 3.a*) captured by a CCD camera and analyzed by KLT. One can distinguish in the first image in red color the points to be tracked. In the following image, one can distinguish the points which have been recognized with many irrelevant elements which constitute a noise. Because of this, many points are lost in the second image and the success of the tracking decreases according to the complexity of the scene and of the environment. However, since we have only one object (hand) that is supposed to move, therefore very few corresponding points suffice to detect the movements. According to our experiments, about 60%were successful. Therefore, it appears necessary to control the environment to obtain more success and to make the gestural-based robot control effective.

A similar group of seven experiments has been performed in conditioned environments. In these experiments, the user's hand appears clearly without other objects in the background. The illumination is considered as stabilized. *Fig. 3.b* shows two successive images (at the right side of *Fig. 3.a*) captured during the execution of a hand movement. One can observe that the hand tracking is more successful compared to unconditioned and noisy environments. Experiments have clearly confirmed that conditioned environments highly improve the performance of the tracking process of the KLT algorithm. The success rate was about 90 %. The performance of this system can be improved also by using specific marks and colors to make the moving hand more easily detectable [11].

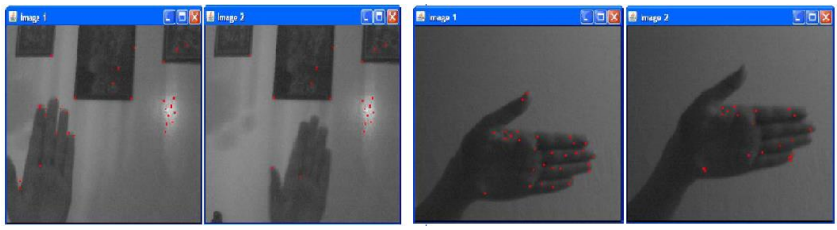

Figure 3: Experiments in different environments

a)-Natural environment (the two images on the left), b)-Conditioned environment (the two images on the right)

Considering the speed of the gesture-based interactions, some authors have tested the speed of KLT algorithm with different implementations on different platforms [23], [24]. Of course, it depends on the CPU, on the image size, on camera resolution, on the defined maximal number of features to be tracked, etc. In our case, we have used Java media framework to develop a program that extracts frames from the video sequence. The frames are of JPEG format and are then converted to PGM format. The camera resolution is 352*288 pixels. The process has been tested in real time to the KLT tracker. Since our applications require only relatively low speed motions, the execution of the image-processing based on KLT algorithm tested on ordinary computers in conditioned environments was about tenth of a second. Thus, this timing was considered as satisfying. However, it is possible to speed up real time applications that require faster execution time by specific equipment and optimization procedures [23], [24].

4. Discussion

In our analysis of this human-robot interaction mode, we consider two cases. The first case consists of focusing on tracking the movement. What counts here is not only the object to recognize as in static object recognition processes but it is the movement of objects itself. The second one consists of considering the configuration change of the object of interest once it stabilizes after the movement from the initial configuration to the final one. In fact, in this last case, the relevant information is contained in the final configuration that needs to be converted into a robot command according to a pre-established code.

All the described softwares can be adapted to track continuously in the video sequence the movement of the objects of interest. However, the problem of continuously tracking which is time consuming can be avoided in most applications since it suffices to detect the motion and to recognize the final configuration in order to identify the corresponding command to be issued. Again, according to our analysis, we inferred that each described image recognition method may be more efficient depending on the conditions of the application in its context. Here are some suggested cases to show how one can select the relevant method to apply with respect to the given contexts.

If we can define an easily recognizable structure that serves as a model of the human body part to be tracked (arm, hand, head, leg, etc.) and if the environment is not too complex; then the SSDA can be the best suitable software to apply.

If we have a limited set of movements that constitute a code and if the environment can be complex with possible variable illumination; then the appearance based method can be the best suitable software to apply. This

method can recognize the appropriate code after the end of the movement by simply comparing the actual obtained image with images contained in the database.

If the application does not require a specific model and the relevant information is contained in the movement itself and if the environment is not simple structured; therefore, KLT can be the best suitable software to apply as it does not require necessarily an initial specific model to detect. It tracks the movement of the human body part and when it stabilizes, it can identify the configuration to decode.

Of course, all these tracking vision methods are subject to some limitations, which are mainly due to the sensitivity of the vision systems, to lighting conditions, to the environment of work and to the application at hand. To improve the performance of the vision and tracking systems, our experiments have shown that it is recommended to condition the environment of work and to structure the tracked object. Moreover, other experiments have shown that using colors and markers may also improve the system performance by facilitating and guiding the tracking and recognition processes.

This gesture-based interaction mode was tested on various kinds of robots such as robot manipulators, mobile robots and cable-based robots as well [11]. Experimental tests showed that the gesture control mode can be effectively and worthily used for interaction with robot and telerobotics systems. New applications can be implemented where human are using gestures to interact with each other such as the take-off and landing of unmanned systems, aircraft fighters in military airport and marine vessels.

5. Conclusion

This article presents some methods of image processing which have been experimented for object recognition and 3D identification of locations. They are namely: Sequential Similarity Detection Algorithm (SSDA, Kanade-Lucas-Tomasi (KLT) algorithm, and an appearance-based approach that uses image databases to model objects. It discusses their applicability for designing and implementing gesture-based interaction with robots. An application using KLT tracker is presented.

Our analysis and experiments have shown that each of the presented methods can be adapted for a particular gestural system according to the body part used, the robotics application and the conditions of experiments. Of course, all these methods suffer from specific limitations, which are mainly due to the sensitivity of the vision system and of the environments. The conditioning of

the environment is an important factor that remarkably improves the performance of the vision system.

The experiments performed with various types of robots have confirmed the possibility of using this gesture detection mode as an interesting one for many applications in robotics and telerobotics domains. It can be also worthily included into multimodal human-robot interaction systems.

References

[1] Zaatri, A., and Ousalah, M., "Integration and design of multi-modal interfaces for supervisory control systems", *Journal of information fusion*, vol.4, pp. 135–150, 2003.

[2] Abiri, A., Heise, G., Zhao,X., Jiang,Y., and Abiri, F., "Brain Computer Interface for Gesture Control of a Social Robot: an Offline Study", in *Proc. 25th Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran*, May 2-4, 2017, pp. 113–117.

[3] Mitra, S., and Acharya,T., "Gesture Recognition: A Survey", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp-311-324, May 2007.

[4] Sigalas, M, Baltzakis, H., and Trahanias, P., "Gesture recognition based on arm tracking for human-robot interaction", in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2010, pp. 5424–5429.

[5] Mathe, E., Mylonas, P., Spyrou, E., and Mylonas, P., "Arm Gesture Recognition using a Convolutional Neural Network", in Proc. *13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP*), 2018, pp. 37–42.

[6] Luo, J., Liu,Y.,and Ju. Z., "An Interactive Astronaut-Robot System with Gesture Control", *Comput. Intell. Neurosci.* vol.2016 (1), pp. 1-11, 2016.

[7] Staub, C., Can, C., Knoll, A., Nitsch,V., Karl, I., and Färber, B., "Implementation and Evaluation of a gesture-based Input Method in Robotic Surgery", in Proc. *IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*, Qinghuandao, Hebei, China, October 2011, pp.1–7.

[8] Elliott, R, E., Hill,S. G., and Barnes, M., "Gesture-Based Controls for Robots: Overview and Implications for Use by Soldiers", Human Research and Engineering Directorate, ARL, US Army Research Laboratory, ARL-TR-7715, July 2016.

[9] Kewley-Port, D., and Nearey, T. M., "Speech synthesizer produced voices for disabled, including Stephen Hawking", *The Journal of the Acoustical Society of America*, vol. 148, pp. R1-R2, 2020.

[10] Galván-Ruiz, J., Travieso-González, C.M., Tejera-Fettmilch, A., Pinan-Roescher, A., Esteban-Hernández, L., and Domínguez-Quintana, L., "Perspective and Evolution of Gesture Recognition for Sign Language: A Review", *Sensors (Basel)*. vol. 20, no. 12, pp. 3571, June 2020.

[11] Zaatri, A., and Bouchemal, B., "Some interactive control modes for planar cable-driven robots", *World Journal of Engineering*, vol. 10, no. 5, pp. 485–490, 2013.

[12] Zaatri, A., Tuytelaars, T., Waarsing, R., Van Brussel, H., and Van Gool, L., "Design and Implementation of a Supervised Intelligent Function", in Proc. *International Conference on Artificial Intelligence, IC-AI00, Las Vegas, USA*, 2000, pp. 799–805.

[13] Aboud, H., and Zaatri, A., "Suivi de gestes pour la commande des robots", in Proc. *Congres Algerien de Mécanique (CAM-2011), Guelma*, November 2011, pp. 14–17.

[14] Shirai, Y., "Three-Dimensional Computer Vision", Springer.1987.

[15] Tuytelaars, T., Zaatri, A., Van Gool, L., and Van Brussel, "Automatic Object Recognition as Part of an Integrated Supervisory Control System", in Proc.*ICRA, San Francisco, USA*; 2000, pp. 3707–3712.
[16] Tuytelaars, T., and Mikolajczyk, K., "Local Invariant Feature Detectors: A Survey, Foundations and Trends", *Computer Graphics and Vision.*, vol. 3, no. 3, pp. 177–280, 2007.
[17] Baker, S., and Mattews, I., "Lucas-Kanade 20 Years On: A Unifying Framework", *International Journal of Computer Vision.,* Kluwer Academic Publishers, vol. 56, no. 3, pp. 221–255, 2004.
[18] Besl, P. J., and Jain, R. C., "Three-dimensional object recognition", *ACM Comput. Surv.*, (CSUR)17, pp.75–145, 1985.
[19] Lucas, B., and Kanade, T., "An iterative image registration technique with an application to stereo vision", in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.
[20] Tomasi, C., and Kanade, T., "Shape and Motion from Image Streams: a Factorization Method—Part 3 -Detection and Tracking of Point Features", Technical Report CMU-CS-91–132, April 1991.
[21] https://cecas.clemson.edu/~stb/klt/
[22] http://www.java2s.com/ref/jar/download-klttracker131jar-file.html
[23] Fassold, H., et al., "Realtime KLT Feature Point Tracking for High Definition Video", Proceedings, https://www.yumpu.com/en/document/view/42917243/realtime-klt-feature-point-tracking-for-high-definition-semedia, 2009.
[24] Barnes, B., et al., "Evaluation of Feature Detectors for KLT based Feature Tracking using the Ondroid U3", in *Proceedings of Australasian Conference on Robotics and Automation*, The University of Melbourne, Melbourne, Australia, 2014, pp. 1–9.