Advanced Methods of Information Technology for Autonomous Robotics

Computer vision, geometric reasoning and graphics

Dejan Marković, Nicola Vitucci

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
dmarkovic@elet.polimi.it, vitucci@elet.polimi.it

Abstract—Robotics represents a complex research field that has seen a huge development in the last few decades. The big progress that Information Technology (IT) made both in terms of hardware (i.e. processing power) and software (i.e. development of new methods and techniques) allowed robots to slowly move out of the industrial environment (e.g. factories) into a variety of different, more challenging, application environments (e.g. space exploration). As a consequence, the focus has been shifted from a control theory perspective to a computer science and engineering point of view. The aim of this paper is to show some of the most important vision-related topics which can have a more or less direct impact on the robotics research field. The paper covers both some basic methods and some of the most recent techniques and hot topics in the IT.

I. INTRODUCTION

Research in the field of autonomous robotics has gained much importance since the introduction of robots in non-industrial (i.e. possibly dynamic and/or unknown) environments. In such environments, robots have to perform tasks that require a certain level of autonomy. The wide variety of problems that autonomous robots have to face while performing such tasks led to a joint effort in a number of different research fields: neurobiology (especially biological vision), psychology, cognitive science, physics, mathematics and artificial intelligence are some examples of the many different disciplines that may be involved in the process of giving autonomy to a robot.

The importance of autonomy can be understood while dealing with highly unstructured environments: it is obvious that a robot, to be actually usable, should interact directly with the surrounding environment and take its decisions according to its goals and the constraints it is subject to. The interaction involves the usage of sensors and actuators, and of course of a computation stage on data coming from such hardware, that in some cases can look similar to a human cognitive task.

In the following, computer vision will be shown as the basis for interaction with the real world, explaining why it is usually the preferred choice with respect to other sensory channels; then, applications of computer vision techniques and vision-related topics will be presented and discussed; in the end, some considerations will be made as a possible hint for further discussions.

II. COMPUTER VISION

In autonomous robotics the vision seems to be the most important sense for a variety of reasons. We can argue that it is because we tend to imagine autonomous robots similar to human beings. However, the fact is that for a robotic system to become truly autonomous it is not enough to simply navigate in a known static environment: it should be able to understand and interact with dynamic or unknown environments. Autonomous robots should be able to detect different (both expected and unexpected) events, make decisions and learn. Furthermore, autonomous robot systems should be able to perform different tasks (e.g. localize, track and manipulate objects within the environment and so on); in order to do so, robot systems have to collect and analyse information from the environment. In this process, Computer Vision systems play a key role: the task of computer vision is to extract information about the environment (3D world) from acquired images (it could be a single image, a video sequence or views from multiple cameras).

A. Related Fields

Computer vision can be defined as the science and technology of machines that see [1]. As such it is closely related to a number of fields (see Figure 1): for instance it elaborates image data, thus many methods used in computer vision are shared with the Image Processing and more generally with Signal Processing research fields. Computer vision algorithms make use of mathematical and engineering fields such as Geometry, Optimization, Probability Theory, Statistics, etc. From the above discussion on the use of computer vision in Robotics it is clear that the computer vision is also used in Pattern Recognition (e.g. face and object recognition), Machine Learning and Artificial Intelligence (e.g. navigation through the environment). Furthermore, computer vision sensors are designed using solid-state Physics (the propagation of light is explained by Optics) upon human vision system (Biological Vision system studied in Neurobiology). In industry and manufacturing we speak about Machine Vision as the application of computer vision in more controlled and less general working environments (with controlled lighting, positions of objects, etc.).
B. Short History

The evolution of computer vision and related topics started in the '60s. From 1960 on, the interest in digital image processing by computer grew up. The first computer vision systems started to appear in the mid '60s (Robert’s PhD and Guzman’s PhD at MIT). In 1968 the 1st Journal on Pattern Recognition came out, in 1969 the 1st textbook was published [2] and in 1970 the 1st International Conference in Pattern Recognition was held. Evolution continued in following years: in 1978 the International Association for Pattern Recognition was formed and in 1979 publication of IEEE Transactions on Pattern Analysis and Machine Intelligence started. The 1st International Conference on Computer Vision was held in 1987.

C. Applications

Computer vision has a variety of applications in different fields (Figure 2). In fact, the use of a vision sense is not limited simply to robotics. The application fields for computer vision are many, from medical research to military applications and space explorations. For example, computer vision can be used in medical research for detection of tumours or other malign changes, measurements of organ dimensions, blood flow, etc. In industry it can be applied for counting objects, reading serial numbers or searching for surface defects. Many are the applications in the military field: target detection, missile guidance and unmanned aerial vehicles, just to name a few. Furthermore, computer vision is essential in space exploration missions such as NASA’s Mars Exploration Rover and ESA’s ExoMars Rover. It also finds applications in security (visual surveillance systems), advertising, cinema and broadcast (think about visual effects and augmented reality).

As long as robotics is concerned, it appears clear that the visual sensors, while not as essential as in autonomous robotics, are used in industrial robots as well [3], [4], [5]. The vision-based control is the use of one or more cameras and a computer vision system to control the position of the robot’s end-effector relative to the work piece as required by the task [3]. Its use is made necessary in order to improve precision of the robotic manipulator in presence of positioning errors, uncertainties in task planning, incomplete knowledge of the working environment, etc.

In relation to the relative position of the visual sensor and the robotic arm, we distinguish two configurations (Figure 3): eye-in-hand configuration (i.e. when the camera is mounted on the moving robotic arm) and eye-to-hand configuration (i.e. when the camera is fixed). Furthermore we can distinguish between position-based visual servo control (Figure 4) and image-based visual servo control (Figure 5). In the first case features extracted from the image are used to estimate the position (in 3D world) of the target with respect to the camera; the control is then applied to the end effector in order to reach the desired position in space. In the second case, instead of defining error signals in terms of 3D space coordinates, the control values are computed directly from image features.
D. System Architecture

In order to accomplish a given task, computer vision systems apply several steps of data processing, from low-level processing that essentially leverages image processing techniques, to high-level processing that makes use of more specific knowledge about the given problem and its geometry. Some typical steps of a computer vision system are the following: image acquisition (done using different types of visual sensors); data pre-processing (e.g. denoising, contrast enhancement, etc.); feature extraction (e.g. edge, corner and shape detection); detection/segmentation (e.g. object detection) and high-level processing (e.g. object recognition). However, not all these steps are always present in computer vision systems: e.g. an industrial robot that needs to recognize objects of different colours will process the image data up to feature extraction step (i.e. colour extraction).

Among the low-level methods, the most common are blob detection, edge detection, corner detection, morphological operators, template matching and shape detection. These are some basic methods used in image processing. The high-level methods on the other hand make use of more a priori information together with pattern recognition and artificial intelligence techniques, and find applications for example in motion detection and tracking, SLAM, object recognition, etc. (see Sec. V).

III. IMAGE PROCESSING

In this section we make a short overview of some of image processing methods used as low-level processing in computer vision systems. The section is structured as a quick introduction to different methods, thus it covers only basic and most common techniques. For more details consult dedicated literature.

A. Blob Detection

Blob detection methods are used to determine if several connecting pixels in a group are related to each other. This information is useful for identifying separate objects or counting the number of objects in a scene. The distinction between different pixels can be achieved simply applying a specific threshold or classifying pixels by their colour. For further (higher level) processing, a pixel classification using a standardized set of colour terms can be useful (e.g. NBS/ISCC Color System).

B. Edge Detection

The edge can be detected finding the maximum of the gradient,

\[ \nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} i_x + \frac{\partial f(x, y)}{\partial y} i_y, \]

or finding the zero crossing of a laplacian,

\[ \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}, \]

(see examples in Figure 6). The gradient-based methods can be either non-directional (maximum of \(|\nabla f(x, y)|\)) or directional (maximum of \(f_x = |\partial f/\partial x|\) and/or \(f_y = |\partial f/\partial y|\)). The laplacian-based methods are usually more sensible to noise making noise reduction step a necessary prerequisite.

Both methods use a discrete approximation for the gradient/ laplacian operator, that can be seen as a convolution with a discrete filter:

\[ \nabla f_x(n_1, n_2) = f_x(n_1, n_2) * h_x(n_1, n_2); \]

\[ \nabla f_y(n_1, n_2) = f_y(n_1, n_2) * h_y(n_1, n_2); \]
Different filters, \( h(n_1, n_2) \), with different properties can be used. The standard gradient-based filters are: Roberts, Sobel, Prewitt, ecc. Some examples of common laplacian-based filters are: four neighbors, Prewitt, eight neighbors, ecc.

Canny edge detector [6] is built optimizing the three performance criteria: good detection (i.e. low probability of non detecting a real edge and low probability of detecting a false edge), good localization (i.e. detected edges should be as close as possible to their actual locations) and only one response to the single edge (implicitly contained in the first criterion but mathematically handled separately). Intuitively the mathematical form of the first criterion is the output Signal-to-Noise Ratio (SNR) and thus the detection is optimized maximizing the SNR. The localization is improved maximizing the inverse of the standard deviation from the actual location. Elimination of multiple responses (third criteria) is handled separately and is controlled by a parameter that fixes the number of noise maxima that could lead to a false response. The form of the filter is related to the third criterion and the most commonly used form is the derivative of the Gaussian. The length of the filter is related to the first two criteria: short filters (few samples) give better localization while long filters (limited in frequencies) better SNR. Threshold value determines the sensibility of the edge detection.

C. Corner Detection

Harris corner detector [7] classifies the regions of an image into one of three classes: flat region, edge and corner. Flat region is characterized by approximately constant intensity and thus all shifts of the windowed image patch (i.e. shifts of the window by a small amount in various directions) will result in only a small change. If the window straddles an edge, then a shift along the edge will result in a small change, but a shift perpendicular to the edge will result in a large change. If the windowed patch is a corner then all shifts will result in a large change. Denoting with \( I \) the image intensity and with \( w \) the window function we can write the change produced by a shift \((x, y)\) as

\[
E(x, y) = \sum_{u,v} w(u, v) |I(x + u, y + v) - I(u, v)|^2 .
\]

Using Taylor approximation we write

\[
E(x, y) = Ax^2 + 2Cxy + By^2 = [x, y] M [x, y]^T.
\]

Notice that \( E \) is closely related to the local autocorrelation function and the eigenvalues of \( M \) are proportional to its principal curvatures (rotationally invariant). As a consequence the classification can be done using eigenvalues of \( M \) as shown in Figure 7.

As the computation of eigenvalues (i.e. the eigenvalue decomposition) is computationally expensive usually the classification is done using Corner Response Measure

\[
R = \det M - k(\text{trace}M)^2 ,
\]

\( \nabla^2 f(n_1, n_2) = f(n_1, n_2) * h(n_1, n_2) \).

Figure 7: Classification via eigenvalues: Flat region - both eigenvalues are small, little change in local autocorrelation function, approximatively constant intensity; Edge - one eigenvalue is high and other low - local autocorrelation function changes along one direction (the one perpendicular to the edge), little change along the edge; Corner - both eigenvalues are high, local autocorrelation function changes along all directions.

\[
det M = \lambda_1 \lambda_2 ,
\]

\[
\text{trace} = \lambda_1 + \lambda_2 ,
\]

and \( \lambda_1 \) and \( \lambda_2 \) are the eigenvalues. In this case small \( |R| \) indicates a flat region, \( R < 0 \) an edge and \( R > 0 \) a corner.

D. Morphological Operators

Morphological operators [8] are mathematical tools used to analyse, represent and describe forms of a region in an image. The goal is to distinguish relevant from irrelevant information about the form of an object. There is a number of morphological operators that are usually obtained with a combination of two basic operators, which are the morphological dilation:

\[
A \oplus B = \{ h | B_h \cap A \neq \emptyset \} \tag{1}
\]

and the morphological erosion:

\[
A \ominus B = \{ h | B_h \subseteq A \} \tag{2}
\]

Examples of morphological dilation and erosion are shown in Figure 8. Using this two operators other operators can be defined, such as morphological aperture:

\[
A \circ B = (A \ominus B) \oplus B \tag{3}
\]

and the morphological closure

\[
A \bullet B = (A \oplus B) \ominus B \tag{4}
\]

Morphological operators are useful in different tasks. They can be used to extract borders, skeleton or convex-hull. An example of border extraction, \( \beta(A) = A \ominus (A \oplus B) \), is shown in Figure 9.
F. Shape Detection

The Hough transform can be used to identify different shapes in an image. In its classical form it is used to find lines. The Hough transform parametrize lines with two parameters, $\rho$ and $\theta$, as shown in Figure 10. To each line is associated a unique couple $(\rho, \theta)$ and thus a unique point in the Hough parameter space. A set of all lines that pass through a point $(x_0, y_0)$ is given by the equation

$$\rho = x_0 \cos(\theta) + y_0 \sin(\theta),$$

corresponding to a sinusoidal curve in the Hough space. As a consequence the Hough transform of an image with $K$ lines is a sum of many sinusoidal curves (the number of curves is equal to the number of points that form these lines) that meet in $K$ points.

Figure 10: Hough transform of a line.

IV. HIGH-LEVEL PROCESSING

A. Pattern Recognition

The pattern recognition is the assignment of a physical object or event to one of several prespecified categories [9]. The objects or events are called patterns; the set of patterns sharing common attributes and properties is the pattern class. The assignment of given patterns to prescribed classes is called recognition or classification process. A typical pattern recognition system is shown in Figure 11.

Figure 11: The pattern recognition process.

Sensors acquire signals and measurements from patterns. The raw data from sensors is usually affected by noise, thus denoising and possibly some other preprocessing operations on raw data are required. However, making classification directly using sensor data is difficult. Feature extraction aims to create discriminative features which are good for classification. Good features are the one which guarantee that patterns from the same class have similar feature values and patterns from different classes have different values. The correspondence between
feature values and classes is established during the learning process. There are two types of learning process, supervised and unsupervised learning: supervised learning algorithms require an expert (teacher) to provide, for every sample input pattern, the correct output class; in unsupervised learning algorithms, instead, the patterns are clustered depending on their similarity without knowing a priori their class, then the classifier assigns a class to a given pattern depending on its features.

Pattern recognition is used in many fields. Some of the application are: text (letter) recognition, speech recognition, face recognition, finger prints recognition, medical diagnosis (e.g. X-ray, EKG analysis), machine diagnostics, military applications (e.g. Automated Target Recognition), etc. In robotics, feature extraction is used in navigation, object recognition (e.g. for grasping), face recognition for robot-human interaction (e.g. entertainment, assistance), etc.

V. APPLICATIONS TO ROBOTICS

The discussion carried on so far can give an idea of the broad variety of problems the research in computer vision has faced in the past and is still facing nowadays. The methods and the techniques presented so far can now be considered “classical”, as they are widely used in the computer vision community as standard methods or as the basis for more recent algorithms.

What we are now interested in are the applications of these results to the robotics field. It may now look quite obvious that a robot “has to have eyes” to interact with the world; the problem is to understand why artificial vision is so important, what kind of problems can it help to face and how.

This section is divided in three parts related to computer vision, geometric reasoning and computer graphics. The reason for such a differentiation is that not every task related to images can be labeled as a “computer vision” task, as geometric reasoning is much more close to mathematics and set theory, while the generation and the use of synthetic images by a computer are best classified as “computer graphics” tasks.

A. Applications of computer vision

The computer vision techniques showed in the preceding sections can be useful for several robotic applications.

1) Mobile robotics: A direct application of computer vision techniques is related to mobile robot navigation. A problem arising often in this field is the localization problem, the “Where am I?” of a robot. The robot may or may not have a map of the surrounding environment; in the latter case, the robot has to face a chicken-or-egg problem, that is: how can it localize itself if it does not have a map? And how can it build a map if it does not know where it is?

The Simultaneous Localization And Mapping (SLAM) algorithms provide a way to (at least partially) solve this problem: the map is built along with the robot movements and recognition of already seen places. A major problem in this case is due to error accumulation: if there are errors in detecting landmarks, the same place can be wrongly recognized as a different place when it is seen again, thus leading to the building of a wrong map.

The features extracted from the environment have to be robust and easy to be recognized, thus computer vision techniques can be suitable for accomplishing this task. In [10] and [11] is respectively shown the use of a single camera (Monocular SLAM) and of a stereo camera (Stereo SLAM) to build persistent maps. In the first work the extracted features are large image patches, which are more reliable than simple features like corners, and the depth of the points is calculated using motion; in the second one, the depth is obtained directly from the acquired stereo images.

The most important problems arise when dealing with point correspondence: as it is difficult to decide whether a point or another feature is the same in two different images, different probabilistic techniques as Kalman filtering are used.

2) Object recognition: The problem of object recognition (Fig. 14) has been widely addressed in the past for different reasons and it still receives a lot of attention by the research community. Dealing with robotics, the importance of object recognition is above all considered in the object manipulation research field. One of the most important tasks for an autonomous robot is the ability to grasp objects and to move them, but the robot needs to know what is an object and, upon request, how to identify a specific class of objects or a specific instance of the class. The most obvious solution is the use of methods taken from the computer vision research.

There are many techniques relying on different levels of information that can be extracted from the analyzed image. Starting from the low level, one can use edge matching, gradient matching, pose consistency, Scale-Invariant Feature Transform (SIFT), template matching, part decomposition and so on. Furthermore, switching to “human-inspired” tasks and neuropsychology research field, an interesting approach which...
underwent some investigations is the Gestalt theory and its relationship with computer vision [12].

The Gestalt (literally *shape* or *representation*) theory is a psychology theory which accounts several principles used by humans while trying to recognize objects or to understand representations. What is so attracting for computer vision researchers is the possibility of directly implementing psychology principles on a working system, in order to make the system “see”. Some of the Gestalt principles are shown by examples in Fig. 13.

A problem arising very often in object recognition tasks is due to occlusions, i.e. objects covering other objects. There are different techniques to deal with this issue, although recently probabilistic methods have been attracting a wide interest. In [13] an algorithm to solve this problem has been proposed: it uses the concepts of Procrustean metric to define the “distance” between two shapes and of shape correspondence to find whether two shapes define the same object; the Procrustean metric has been chosen because it is invariant with respect to scale and rotations. The shape correspondence is still an open problem, yet some concepts from probability theory (like the maximum likelihood) are explored and integrated with several constraints to give some interesting results.

The main point of the work is that a partial shape can be “completed” in a way that it can be compared with learned shapes, in order to find the object class to which it belongs. This is useful to identify suitable grasping points and for building a grasping plan: if an object is occluded, the identification of suitable grasping points on its shape has to be followed by a plan stating that the occluding objects have to be removed first, starting from the objects that are already free. An example of this process is shown in Fig. 15, where the aim is to grasp the teddy bear which is occluded by a bottle and a toy banana. After object completion, two suitable grasping points are found and a plan for grasping the bear by removing the occluding objects first is built.

Another possible method to recognize object, as anticipated, is the recognition by part decomposition. The process of decomposing an object in parts has some useful applications in robotics, especially in industrial environments (e.g. in mechanical assembly lines). In [14] it is shown how a 3D object can be decomposed in regions by using geometric concepts as voxels and the distance transform; the found regions are smoothed and labeled so to obtain sensible “pieces”. As anticipated, this method can be useful for dealing with objects in mechanical environments, where an object has to be disassembled or, viceversa, where parts have to be put together to build more complex objects.

A further interesting application of part decomposition is discussed in [15] while dealing again with the grasping problem: to help finding a suitable grasp, an object is decomposed in “boxes” and a specific grasp for each box, depending on the aim of the overall grasping task, is evaluated. This approach is discussed in Sec. V-C and an example is shown in Fig. 18.

B. Applications of geometric reasoning

We now focus on a more theoretical and mathematical topic, namely the geometrical reasoning. This way of dealing
with images has been considered exclusive of humans, but in the last few decades it has been recognized as a promising approach also in automatic computation, above all in computer vision tasks. The importance of recognizing relationships among objects in terms of spatial relationships or connections [16] is now addressed as an interesting problem which can possibly turn out to be useful for image understanding. An example of such relationships can be seen in Fig. 16.

While the applications of geometric reasoning can be easily seen as a good means for image retrieval and semantic interpretation, on the robotics point of view they can be indirectly viewed as an improvement in scene understanding techniques, which are very important both in navigation and in object manipulation tasks. An example of use of geometric reasoning can be the situation in which the robot knows both that “in a kitchen one can find a fridge and an oven”, thus making the robot localization task easier, and that “the bottles are on the table”, in order to make the object localization and manipulation tasks easier. All these tasks require the semantics of all the involved terms to be grounded to the real world, and the definition of geometric relationships can be easily formalized in order to allow spatial reasoning.

1) Integration of fuzzy logic: The research on fuzzy logic, introduced by Zadeh in [17], is still prevailing and it is pervading more and more different research fields. The use of fuzzy logic in geometric reasoning has been proposed by [18] and [19] to take into account the vagueness which characterizes the data coming from the real world. The advantage of using such a representation is related to a clearer and more natural representation of the real world knowledge, when for example it is sufficient to know that an object is “large” or “close to another one” without the need to specify what these properties mean in numerical terms.

C. Applications of computer graphics

The advances in the image synthesis methods and the advent of more and more powerful computers made it possible to use advanced systems of simulation and virtual reality. The main differences can be found in the use of such systems, which can be related to the robot design phases or to actual and real-time robotic tasks, such as manipulation, navigation and so on.

The OpenGRASP (http://opengrasp.sourceforge.net/) simulator is an example of the usage of computer graphics for design and evaluation purposes. This simulator has been built within the GRASP project (http://www.grasp-project.eu/) for helping researchers in developing platforms for grasping and manipulation. In Fig. 19 two examples of the simulation environment are shown.

Examples of applications of computer graphics in real world tasks can be found in machine learning tasks: in [20] some synthetic images of household objects are generated, along with preferred grasping areas, in order to train the system for future grasping acts (Fig. 17). Sticking to grasping and manipulation, in [21] virtual reality is used to evaluate different types of grasp depending on a specific grasping task (e.g. grasping a cup to pour some liquid in it); in [15] an algorithm to generate the most suitable grasp depending also on the structure of an object (by using part decomposition) is presented (Fig. 18).

A different application of computer graphics can be found in [22], in which it is shown that an internal (simplified) simulation of the robot body can be useful to the robot itself, in order to run complete simulations of actions before actually performing them.

![RCC-8 example](image1.png)

Figure 16: RCC-8 example.

![Example of synthetic household objects with preferred grasping areas coloured in red](image2.png)

Figure 17: Example of synthetic household objects with preferred grasping areas coloured in red. The original image can be found in [20].

![Example of grasping by parts](image3.png)

Figure 18: Example of grasping by parts. The object is decomposed in boxes and each box is evaluated, then the final grasp is decided according to the task to accomplish. The original image can be found in [15].
VI. DISCUSSION

Many considerations can be made on the pros and cons of using image-related methods for robotic applications, focusing on more or less technical details and on their reliability, efficiency, plausibility and so on. Nevertheless, an interesting “philosophical” hint for the discussion can be found in [23]:

[...] There will be more effort towards devising processes that can be implemented in highly parallel associative hardware instead of some kind of stored-instruction computer. The currently fashionable preoccupation with mathematics needs to be counter-balanced by more exploration of parallel hardware implementation. [...] Human recognition capability greatly exceeds that of machines. We don’t have an exact and complete wiring diagram for the human visual system, but we can reasonably assert that it is highly parallel. After a neuron has fired, it can’t fire again until after about ten milliseconds. Without high parallelism, the human visual system couldn’t work as rapidly as it actually does.

Taking inspiration from these ideas, we may think to several points to discuss.

- **Are we too much worried about mathematics?** As we have seen in the first part of the paper, most of the presented techniques deal with mathematical concepts which can make the implementation - and the evaluation - actually possible. The point here is if we are too focused on the mathematical models rather than on the actual implementation issues, which can be possibly part of the solution instead of being mere “technicalities”.

- **Are the solutions we are thinking about “too difficult” with respect to the human visual system?** We know that the human brain actually does not perform “computations” as a computer does, so here the point is whether mimicking the human visual system (up to a certain extent) to implement “simple” and effective solutions can be considered a suitable paradigm or not.

- **Does the human visual system efficiency really depend on its underlying parallel architecture?** This problem has been widely addressed in the past, and its solution of course would be crucial for the realization of a human-inspired visual system in that it would formally show that the physical level has to be definitely taken into account. The problem is still unsolved, so there is still no clear evidence that a massive parallelization would perform better than the most efficient non-parallel algorithms.

- **Do we really need to emulate a human visual system on a robot? Can non-human devices perform better?** Until now, we have been taking for granted that the human visual system is the most effective and performing system. It is possible that we stick to this concept because we do not have a clear idea of a completely different (possibly artificial) alternative, thus we cannot actually evaluate the difference in the performances as we should define a metric of performance first.

Moving to a higher level, there are some other interesting ideas to discuss:

- **Dealing with computer graphics, is it more useful to humans or to robots?** The answer to this question seems to be quite obvious, but works as the ones which have been mentioned show that some interest has been shifted from the simulations intended for humans to a sort of “experiments in virtuo” intended for robots to evaluate, learn and plan.

- **Vision allows us to easily “browse” the scene in order to detect and focus on details and objects of interest. What about other senses?** This question is interesting in that it is not always noticed that the most used sensory channel is vision. A huge research has been carried on audio and haptic channels and some also on the olfactory channel, but actually the reason why vision is most used is its capability to deal with many stimuli at a time (focus of attention). So, for example, in order to detect an object of interest in the scene using a haptic channel we need to touch randomly many objects until we find the desired one. On the other hand, if we use a vision sense we can detect such an object from a single view (i.e. image) of the scene. Similarly to haptic channel, in order to select a audio track we have to hear different tracks one by one as hearing them simultaneously would make things much more difficult.

VII. CONCLUSIONS

In this article we have underlined the importance of IT methods and techniques for dealing with autonomous robotics tasks. We have showed that the vision sense is the most used for making the robots more autonomous, then we have showed the most studied and used techniques in the computer vision research field and their applications in the robotics field, making some differences among what is actually defined as computer vision from other image-related research fields, namely geometric reasoning and computer graphics. In the end we proposed some hints for a discussion, suggesting that further research in autonomous robotics should take into account some “philosophical” and methodological observations that can lead to the invention of new research guidelines.
REFERENCES