Object Detecting and Tracking with Obstacles

Levon Balagyozyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: balagyozyanlevon@gmail.com

***Abstract*— This paper investigates the object detecting and tracking techniques and suggests a possible solution for object tracking with obstacles in the frame content using computer vision tools.**

***Keywords*— Object, detection, tracking, recognition, localization, classification, segmentation, egomotion, optical flow, computer vision, algorithm, histogram, cascade, Haar cascade**

I. Introduction

In today's world, as technologies evolve with exponential speed, almost every aspect of our lives becomes more and more dependent on AI. One of the most growing AI fields is computer vision. Using computer vision tasks like image processing, object detection, object tracking, and text recognition makes tremendous achievements. One of these achievements is the autonomous security systems, which are heavily used in our daily lives. These use-cases vary from simple home door lock systems and/or smartphone AR camera toys to highly secure government applications and national bank security systems.

Object detecting and tracking in an environment with a presence of multiple obstacles is a very common task, but the existing solutions sometimes can be very expensive in computational resources and expensive in general.

This paper addresses several concepts of image processing, object detection, and object tracking and suggests optimal use cases for each of the spoken concepts, and also suggests the best general solution for this task.

II. Computer Vision

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images, videos, or live footage, for example, from security cameras. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do. Computer vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single footage frame or a sequence of frames. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multidimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems [1-5].

Computer vision suggests solutions for a variety of problems some of which are listed below:

- Recognition
  - Object recognition
  - Object classification
  - Object localization
  - Frame segmentation
  - Object detection
- Motion analysis
  - Egomotion
  - Tracking
  - Optical flow

III. Recognition

In computer vision, recognition is a general term that describes the process of analyzing, classifying digital images and identifying different objects in them. These three main problems of object recognition in the digital frame often create confusion about the meaning and difference between these terms.

A. Object recognition

The term object recognition is mainly used to encompass both image classification (a task requiring an algorithm to determine what object classes are present in the image), as well as object detection (a task requiring an algorithm to localize all objects present in the image) tasks.

B. Object classification

Take a look at Figure 1 that shown below:

Figure 1

It's obvious that there is a dog in the image. Let's pause for a moment to analyze how we recognize it instantly. As we already have knowledge about how a dog looks, we can classify the object type shown in the picture above as a dog. And this is a basic concept of the purpose of object classification (Figure 2).

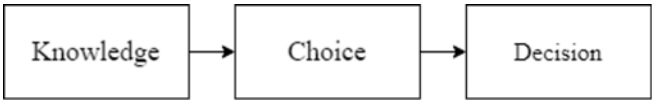

Figure 2. The basic concept of object classification.

This example describes a case where there is only one main object shown in the image. In the case of having multiple objects in the image, the object recognition algorithm must iterate over all possible predicted classes for every object to correctly classify all the objects.

After all classification jobs are finished, the object recognition algorithm stumbles upon determining objects' locations in the selected frame of the current content.

C. Object localization

Object localization refers to identifying the location of one or more objects in an image and drawing a bounding box around their extent (Figure 3).

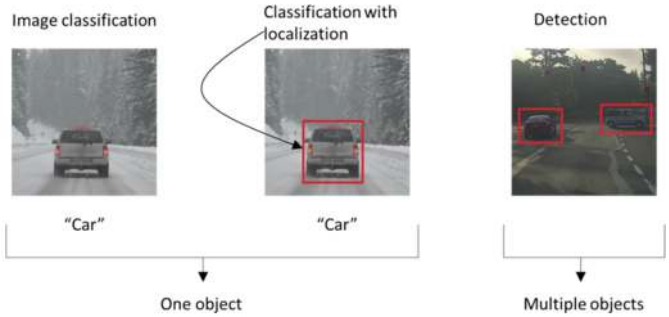

Figure 3. Object localization

D. Frame segmentation

When we have an object recognition problem, before applying the object detection, classification, and even localization algorithms, it is crucial to understand the consistency of the selected frame firstly. To solve this problem, we rely on the concept of frame segmentation. The algorithm of frame segmentation is basically dividing the selected frame into segments and individually analyzing them to get the essential data for the particular problem. The reason for this procedure is to get rid of the unnecessary data that the selected frame possibly can content.

It is common knowledge that any image consists of a set of pixels. The way the frame segmentation works is by grouping together the frame's pixels that have similar signatures (color range, pixel location, etc.) (Figure 4).

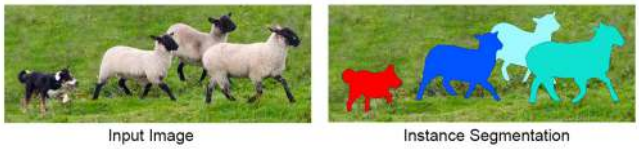

Figure 4. Frame segmentation example.

E. Object detection

As described above, the object localization task is responsible for the single object localization in the selected frame. In the case when the object recognition algorithm must deal with multiple object localization problems, object detection concepts can be used. Object detection does multiple object localization in a single frame.

IV. General Summary

Summing up the computer vision recognition problem. Now we have a theoretical understanding of how each of its main components work.

- Object classification - classifies the content of the current frame
- Object localization - locates a single object in the current frame
- frame segmentation - creates a pixel-wise mask for each object in the current frame
- Object detection - locates multiple objects in the current frame

In Figure 5 you can see examples of each problem.

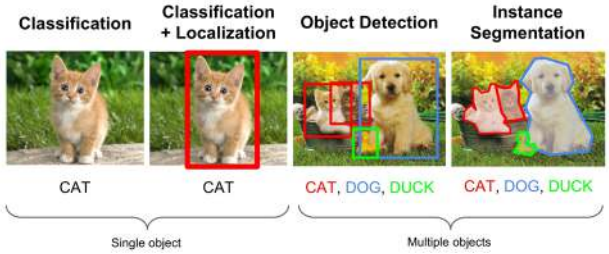

Figure 5. Examples of each computer vision problem

V. Motion Analyses

Motion analysis is the method that studies a sequence of graphical frames (that can be retrieved from a video and/or some footage from a high-speed camera) and gives us information about apparent motions in the given frames. This

algorithm may vary based on the application/scenario. In some scenarios, the camera can be steadily fixed and the objects of interest can be in random motion (Figure 7). In the other scenario, the shooting camera can be mounted on a moving object that moves around a point of interest. There can even be cases when both the shooting camera and the object of interest are moving.

For simplicity, let's say that the camera is an approximation of a pinhole camera. In this case, each pixel of the selected frame is reflected by a single ray coming from the corresponding point of the scene (to which our camera is pointing) that is illuminated by the light source (Figure 6).

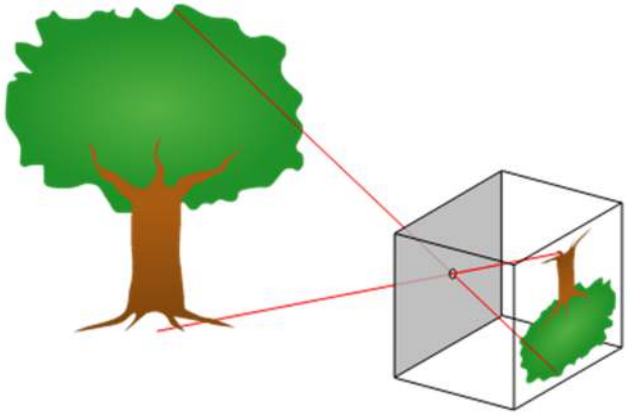

Figure 6. The pinhole camera example.

As you can see in this case, the motion analyzing algorithm can easily detect motion by simply comparing two frames taken at different times.

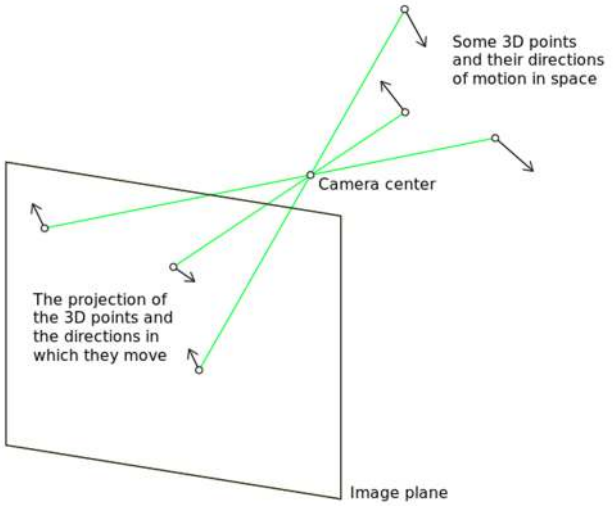

Figure 7. The steadily mounted camera and moving objects around it.

The simplest case for motion analysis is the motion detection application. That means that the working algorithm has to find out if any pixel has been changed throughout the sequence of frames. In the more complex applications, the motion analyzing algorithm must track groups of pixels (that are together forming a rigid object) during some time, and/or examine magnitudes and directions of the moving pixels. The algorithm is working with the information that corresponds to one frame at a time, which means that this algorithm gives us time-depended information about motion.

A. Egomotion

Egomotion is a motion of the camera in the space of the selected scene [6]. Computer vision applications are using egomotion principles to estimate the moving camera's motion corresponding to one or multiple objects in the observed scene [7]. For example, let's consider a scene that has been shot by a flying drone over a forest. We will see that trees are moving towards the camera. In this case, the egomotion algorithm can calculate where the selected tree is after some flight and inform the main drone algorithm if it is an obstacle for flight. Figure 8 shows another scenario of egomotion algorithm principles.

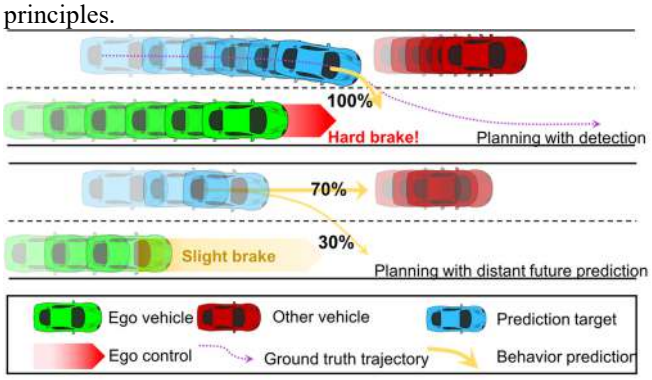

Figure 8. Egomotion principles by moving cars.

B. Tracking

If the motion analysis concepts are used to detect motion in the frame sequence that the video tracking or just tracking is an algorithm that helps to locate moving objects over time using video frames or directly reading frames from the camera hardware. Video tracking can be a very time-consuming process depending on the amount of data that needs to be processed. In some cases, to perform video tracking first, the application needs to use an object recognition technology to find the object of interest in the frames and then perform the video tracking algorithm to locate and track that object [8].

Figure 9. Tracking of the basketball ball.

The main principle of video tracking is to find the selected object in the sequence of frames (Figure 9). That problem can be hard to solve in case that the speed of moving objects is greater than the frame rate. Another difficulty while trying to find a solution for this problem is that the tracked object

changes the moving direction or the moving angle. One possible solution for the aforementioned problem is to generate a motion model. The motion model is a set of instructions that describes the possible movements of the object that might be captured in the video frames.

C. Optical Flow

In a sequence of frames, an observable pattern of motion caused by objects, surfaces, and edges moving relative to each other is called an optical flow (Figure 10 [9, 10]). Except for object detection and tracking, optical flow algorithms are used in various other areas such as image dominant plane extraction, movement detection, robot navigation, and visual odometry [11]. The optical flow algorithms are well known for being of big use in drone software. The problem of inferring not only the motion of the observer and objects in the scene but also the arrangement of objects and the environment is addressed by optical flow. Since knowledge of motion and the generation of mental maps of our world are critical components of animal (and human) vision, the transfer of this intellectual ability to a computer capability is also essential in the case of computer vision [12].

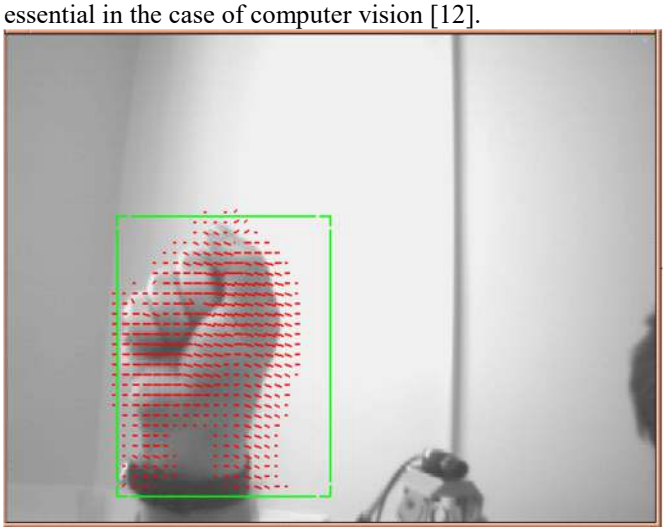

Figure 10. Optical Flow

VI. The Problem of Object Detection in an Environment with Obstacles

In an environment with a presence of a lot of obstacles, the object detection and tracking problem can be very difficult and in some cases even impossible. To solve this problem some solutions are described below. These solutions may vary in different scenarios.

1. Separation of the object of interests from the entire frame
2. Generating rough 3D map using single frame and descriptions for each detected object
3. Switch the view if it is applicable for the problem

A. Possible solution 1

This is a very simple solution for this problem, but it is very handy in many scenarios. The separation of the object of interest can be done in many ways. One of these ways is using object detection based on object color and shape. This method can be used in cases when the object of interest has one main color and has a non-complex shape (for example, “red ball” - the object has a circular shape in a 2D frame and it has the one main color, which is red). For separation, the object detection algorithm needs to be initialized with the object of interest, then the tracking algorithm needs to extract that object from the mainframe and consider the case with an empty rectangle and the “shadow” of the tracking object (Figure 11).

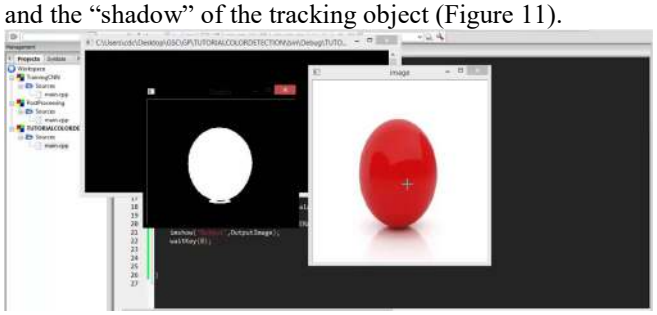

Figure 11. Creating shadow of the red ball.

As you can see, the tracking algorithm (left side) is using the white shadow of the actual red object.

The other way to consider only one object from the whole frame is by using Haar cascades. A Haar cascade is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Let’s discuss an example with a face tracking problem. To train the classifier, the algorithm requires a large number of positive images (images of faces) and negative images (images without faces). After that, we must collect features from it. Haar features, as seen in Figure 12 below, are used for this.

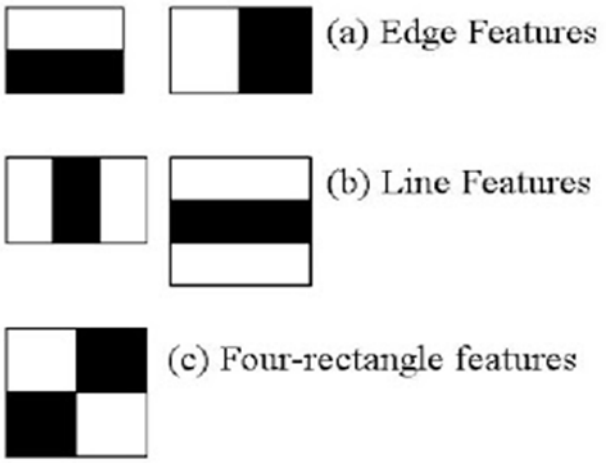

Figure 12. The Haar features.

Each feature is a single value calculated by subtracting the number of pixels beneath the white rectangle from the sum of pixels beneath the black rectangle. (Just imagine how much computation does it need? Even a 24x24 window results in over 160000 features). But usually, the non-face area of an image makes up the majority of the image. As a result, using a simple method to verify whether a window is not a face region is a better idea than analyzing the whole image in order to find the face. If it isn't, throw it out in one go and don't

bother processing it again. Instead, concentrate on areas where a face may appear. We may spend more time testing potential face regions in this way. To avoid this let's use the cascade of classifiers. Instead of applying all 160000 features on a window, the features are grouped into different stages of classifiers and applied one by one. If a window fails the first stage, discard it. So, the window, which passes all stages is a face region [13, 14].

After determining the face region, the tracking algorithm needs to just cut it from the whole frame and use it like it was used in the above example with a ball.

B. *Possible solution 2*

This The main goal of this solution is to generate information about the 3D objects in 2D video footage. For example, the video frame shows the big empty room with a square floor. The camera that captures all is mounted in the nearest top right-hand corner of the room. In the middle of the room, there is a solid box. Also, there is a moving blue ball in Figure 13.

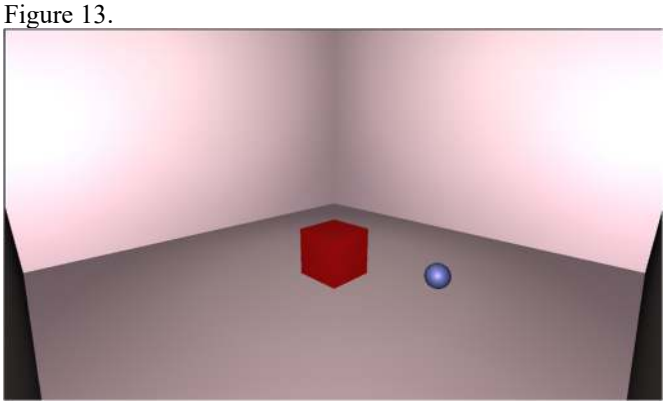

Figure 13. A cube and a ball inside the square room.

When the ball gets behind the cube, the object tracking algorithm will lose the target and it can't be recovered automatically. But what if the algorithm can determine that the obstacle is a cube with certain dimensions and the ball is behind it (Figure 14).

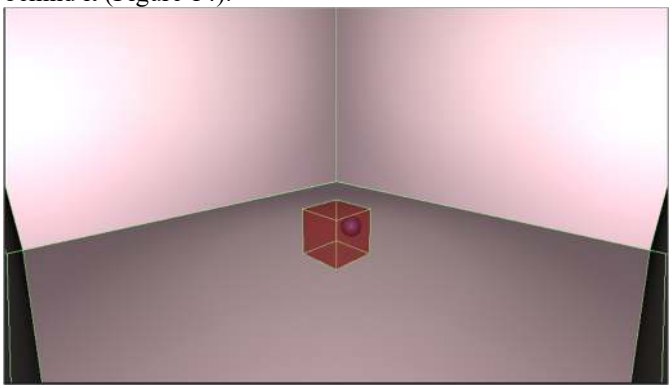

Figure 14. The ball gets behind the cube.

For this task, the algorithm must use several techniques. First of all, it must determine the 2D cube representation as a 3D object to calculate the size of edges. Then the algorithm must use egomotion to determine possible positions of moving the ball after it comes out from the behind of the cube. After that, it must initialize several threads for motion tracking algorithms for each possible position of the frame in which the ball can appear after coming out. When one of the algorithms catches the ball, the other threads must be killed to save the processing resources.

C. *Possible solution 3*

Now let's discuss the case when the ball goes behind the cube but the object tracking algorithm can't determine the 3D sizes of the cube and can't perform an egomotion algorithm. The simplest and obvious solution to continue object tracking is to find another viewpoint so the cube doesn't overlap the ball in the selected frame (Figure 15).

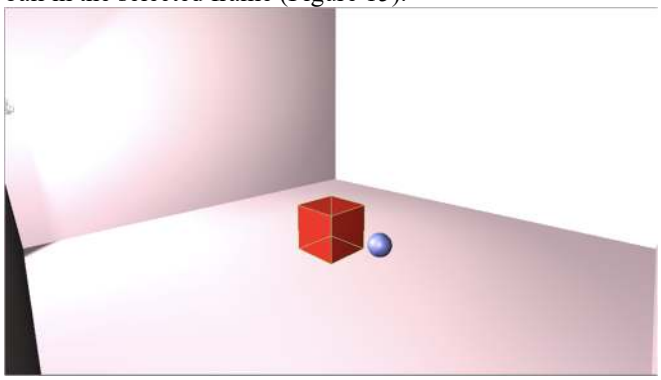

Figure 15. The other viewpoint in the room.

The main problem of this solution is to calculate which viewpoint to choose. For this purpose, the object tracking algorithm must use a very well-trained neural network because the object can have a non-symmetric shape.

VII. Conclusion

Computer vision has many use cases and many algorithms for each possible problem. They are based on human vision and decision techniques.

For object detecting and tracking in an obstacle-full environment, the algorithm must be dynamically adjustable and must combine several solutions. It must handle as many corner cases as it can, but it doesn't need to be very complex in order to save some computing time and resources.

In this article, we have spoken about 3 simple solutions for this problem, but there can be more solutions depending on the selected environment and object of interest.

References

[1] Dana H. Ballard; Christopher M. Brown. *Computer Vision*. Prentice-Hall. ISBN 978-0-13-165316-0, 1982.

[2] Huang, T.Vanoni, Carlo, E (ed.). "Computer Vision: Evolution and Promise" (PDF). *19th CERN School of Computing.* Geneva: CERN. pp. 21–25. doi:10.5170/CERN-1996-008.21. ISBN 978-9290830955. 1996.

[3] Milan Sonka; Vaclav Hlavac; Roger Boyle. *Image Processing, Analysis, and Machine Vision.* Thomson. ISBN 978-0-495-08252-1. 2008.

[4] http://www.bmva.org/visionoverview Archived 2017-02-16 at the Wayback Machine the British Machine Vision Association and Society for Pattern Recognition Retrieved February 20, 2017.

[5] Mike Murphy, *Star Trek's "tricorder" medical scanner just got closer to becoming a reality.*

[6] M. Irani, B. Rousso, S. Peleg (June 1994). "Recovery of Ego-Motion Using Image Stabilization" (PDF). *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*: 21–23. Retrieved 7 June 2010.

[7] W. Burger, B. Bhanu, "Estimating 3D egomotion from perspective image sequence". *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 12 (11): 1040–1058. doi:10.1109/34.61704. S2CID 206418830. Nov 1990.

[8] Peter Mountney, Danail Stoyanov & Guang-Zhong Yang (2010). "Three-Dimensional Tissue Deformation Recovery and Tracking: Introducing techniques based on laparoscopic or endoscopic images." *IEEE Signal Processing Magazine*. 2010 July. Volume: 27" (PDF). IEEE Signal Processing Magazine. 27 (4): 14–24. doi:10.1109/MSP.2010.936728. hdl:10044/1/53740.

[9] Andrew Burton, John Radford, (1978). Thinking in Perspective: Critical Essays in the Study of Thought Processes. Routledge. ISBN 978-0-416-85840-2.

[10] David H. Warren, Edward R. Strelow, (1985). Electronic Spatial Sensing for the Blind: Contributions from Perception. Springer. ISBN 978-90-247-2689-9.

[11] Girshick Ross, (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation" (PDF). Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE: 580–587. arXiv:1311.2524. doi:10.1109/CVPR.2014.81. ISBN 978-1-4799-5118-5. S2CID 215827080.

[12] Christopher M. Brown, (1987). Advances in Computer Vision. Lawrence Erlbaum Associates. ISBN 978-0-89859-648-9.

[13] Paul Viola and Michael J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004.

[14] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900. IEEE, 2002.