Vediamo/Let's See

Introduction

Vediamo/Let's See is an open source vision system built in JavaScript in order to encourage the spread of this kind of expertise to the public.

1 – Rationale

For many animals eyes are the main organs of sense; the view is the prevailing sense because it not only allows them to characterize the surroundings quickly and with great precision, but permits to do it keeping a safe distance, thus avoiding the risk of alerting a potential prey or getting caught by a predator.

Similar considerations also apply to artificial systems that potentially could face similar problems: such as the cybernetic organisms such as six-legged robot designed for the exploration of Mars, the mechanical arms used to paint car's doors in an assembly line, to finish with applications such as access control to a room or optical character recognition.

Given the importance and the unique charm of this type of application it would seem logical to expect a lively interest of the public to learn and experiment with this technology, especially as the materials required to undertake the experiments are available to everyone (everyone brings constantly with him/her at least a couple of webcam integrated into the smart-phone); but the reality is quite different because a quarter of a century after the formalization of the first API for interfacing cameras the ranks of professional programmers not to mention the DIY fan who engage in this type of development continues to be relatively small.

The delay in providing competences regarding the vision is often justified by noting that the vision is difficult because not only imply the possession of vast knowledge concerning diverse subjects but involve the need to address the complexity of cognitive problems.

As if to say: go and take a few master degrees and then we'll talk!

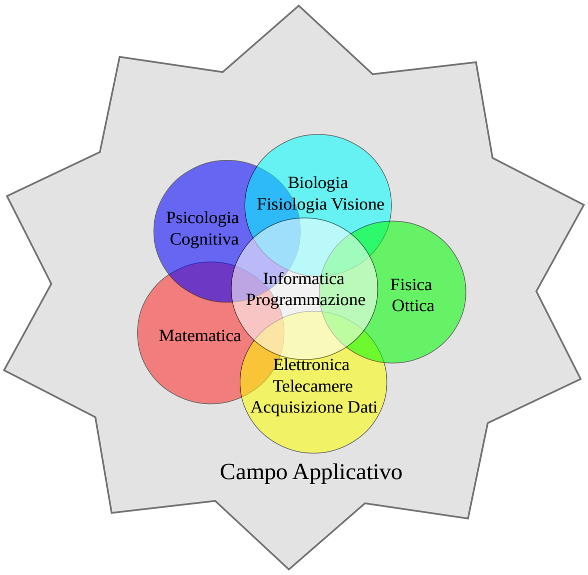


Figure-1) The skills traditionally required for artificial vision development.

2 – Artificial vision is NOT hard.

After that traditionally would follow a dwell explaining why artificial vision system development is difficult, but I will not do it. On the converse I will try to refute the above statement showing that setting correctly the problem it will be possible to "tame" the complexity of the subject in order to make it accessible to an audience of non-specialists.

To accomplish this you must first understand the nature of the difficulties that for a long time have made the artificial vision a topic elitist.

The problems are basically three: the prohibitive cost of professional vision systems, the skills required to use them and the inherent complexity of the cognitive processes.

As you might guess the most difficult problem it is the third since the complexity and the small size of the sensory cells and nerve make it difficult to reconstruct the organization of biological circuits; conversely the human mind operating at a higher level is not aware of the work needed to distill the information detected by the sense organs in order to make it usable for the reasoning, so may need to be a long time before succeeding.

Taking note of this disheartening outcome we could ask: but we really need so much or it could already be possible to obtain useful results with much less?

In my opinion and according to many the correct answer it is the latter. If you expect to have a perfect result you will not do anything, on the contrary it would seem reasonable to assume the possibility to get benefits from the exploitation of the vision abilities of the current machines.

In the following we will assume therefore this working hypothesis proposing to explore the possibility of using the webcam which came with our PC or smart-phone to our personal utility.

This last consideration incidentally provides an answer to the first problem. Given the wide availability of underutilized resources (webcam) we can rule out, at least at first the need to purchase expensive professional tools, so you just have to deal with the issue of competence.

Traditionally the introductory texts to the artificial vision devote at least a couple of chapters on topics of advanced mathematics, explaining the concepts of convolution and Fourier transform, before going into the explanation of the operation of optical devices and shooting. It is assumed that the vision specialist possesses notions of mathematics, physics, optics, electronics and computer science at the university level (in addition to the knowledge of the physiology of vision and cognitive psychology of which is discussed above).

These requirements made sense and were justified if not indispensable at a time when the realization of experiments in artificial vision was time-consuming and required the spending large amounts of money, in similar situation mistakes where not allowed, but today?

The proposal on which is based the system **Vediamo/Let's See** is based precisely on the last consideration that is the assumption that providing resources plentiful and cheap we can afford to reverse the traditional approach to artificial vision-system development addressing the issue in an exploratory way by trial and errors.

But what should be the characteristics of this ideal vision-system?

Based on my experience different users have different needs and what is good for Tom's not good for Harry. So why not put people in a position to go it alone by setting the vision system, or at least to customize its user interface as they please.

Clearly if you are working in C-Language or another "hard" programming language this is not possible but if it were hypothetically possible to reorganize the vision process in order to make it compatible with language closer to end users all maybe it could be feasible.

After a quick examination of the possible alternatives that came to my mind: extension of the functions of a BASIC interpreter, the development of a library for Scratch programming language, Perl, and so on I decided to focus on JavaScript.

3 – Vediamo/Let's See an artificial vision system written in JavaScript

The strength of the proposal lies in the combination of HTML and JavaScript and the ability to use snippets of JavaScript code written by others to introduce "special effects" and/or automate our own web pages.

This practice is true disruptive in nature because it allows virtually anyone to create digital artifacts[1] requiring complex as the only computer skills the ability to use a word processor and make the copy and paste!

The artificial-vision routines developed in JavaScript once made available as open-source code will be immediately available to anyone with the skill to edit text to create new and original products.

In this regard, some skeptics may argue that the proposed solution does not offer even a millionth of the potential of a professional vision-system - no doubt it - but in terms of dissemination of expertise concerning the artificial-vision **Vediamo/Let's See** probably goes beyond any project previously undertaken.

4 – License conditions

Technically **Vediamo/Let's See** consists of a HTML-5 page, which makes reference to the functions contained in *tuisys.js* an artificial-vision library written in JavaScript, because this double nature we provide two licenses.

Please note: both the HTML text that the library is released as open source.

Specifically, the web-automation present in *vediamo.html* and the code of the library *tuisys.js* will be released under the terms of Apache License Version 2.0 while the graphics and the text of the document *vediamo.html* are made available under the conditions of Creative Commons License CC BY-SA.

For licensing details, see:
- http://www.apache.org/licenses/LICENSE-2.0
- https://creativecommons.org/licenses/by-sa/2.0/.

[1] A *digital artifact* is any any digital sequence regardless of his nature as data/information rather than program code (some authors include in the definition also the documentation on paper). This definition is particularly apt for the HTML pages in which the executable code and information are inextricably mixed

5 – Quick start

Because **Vediamo/Let's See** consists of a page of HTML-5 text it is possible to use it simply by opening the corresponding document with a standard HTML browser. In principle you can use any browser as long as it supports HTML-5 and the JavaScript multimedia extensions known as getUserMedia ().

I have tested **Vediamo/Let's See** using the following browser getUserMedia()[2] [3]:
FireFox 17+ (Windows, Linux, Android),
Opera 12 (Windows) and Opera 18+[4] (Windows, Linux),
Chrome 21+ (Windows, Linux),
Midori (Windows).

Microsoft Internet Explorer and Apple's Safari currently supports getUserMedia[5]; Microsoft Edge according to what has been said should be compatible[6].

	Windows	Linux	Android	iOS
FireFox 17+	yes	yes	yes	
Opera 12	yes	yes	Not verified	
Opera 18+	yes	yes	Not verified	
Chrome 21+	yes	yes	Not verified	
Midori	yes	yes	Not verified	
Internet Explorer	no			
Safari				no
Microsoft Edge	yes			

Figure-2) cheat sheet

Eager to try **Vediamo/Let's See**?
Click on the following link:

http://tuisys.com/it/index.html

and then on “try it” button:

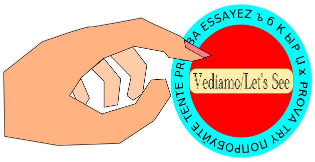
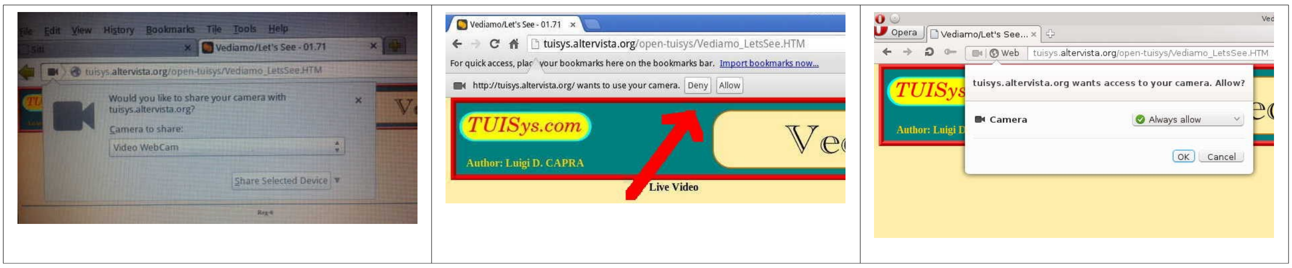


Figure-3) click on “Try it” button

[2] https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia
[3] https://w3c.github.io/mediacapture-main/getusermedia.html
[4] The dynamic of getUserMedia (); in Opera 18+ it differs from what was happening in Opera 12.
[5] See “*Browser Compatibility*” in https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia
[6] https://blogs.windows.com/msedgedev/2015/05/13/announcing-media-capture-functionality-in-microsoft-edge/

Your browser will load the HTML document representing the current release of **Vediamo/Let's See**.

If all goes well the browser should open a window that invites you to allow the use of one of your webcam by the browser.



Would you like to share your camera with tuisys.altervista.org?

Camera to share: Video WebCam

Share Selected Device

http://tuisys.altervista.org/ wants to use your camera. Deny Allow

tuisys.altervista.org wants access to your camera. Allow?

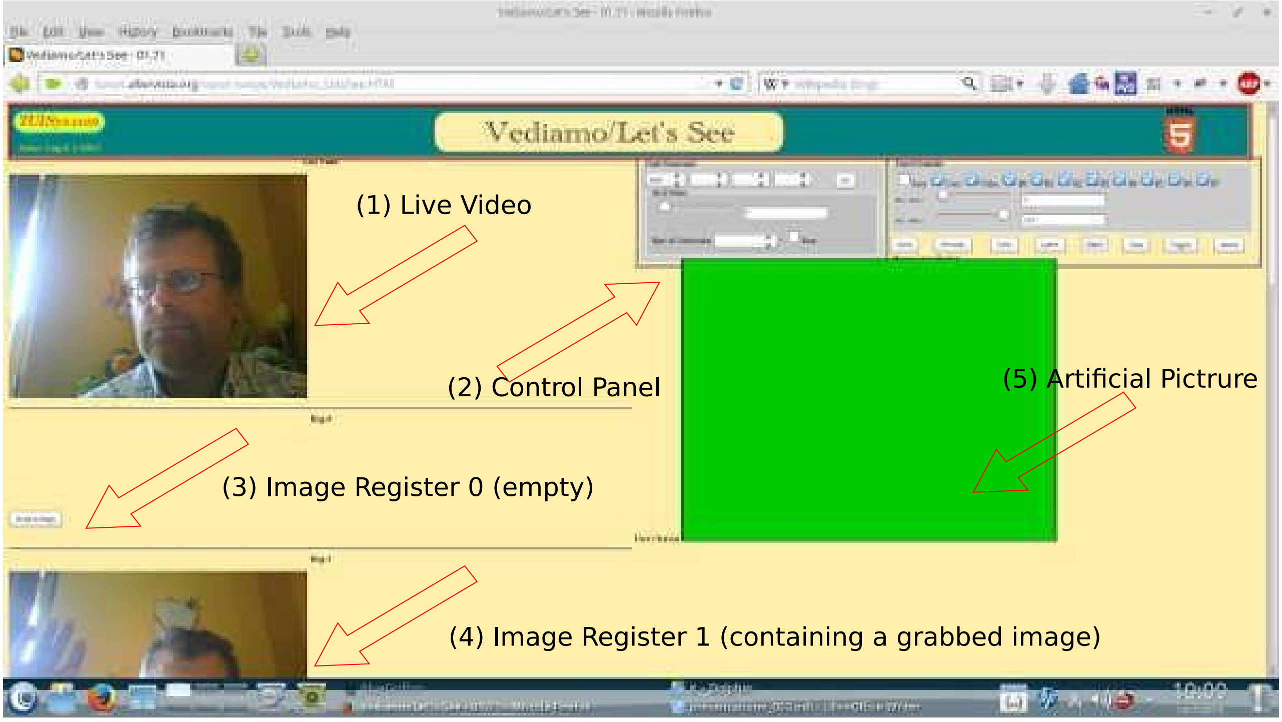
Camera: Always allow

OK Cancel

Figure-4) camera enabling request in FireFox, Chromium and Opera.

In case your computer or smart-phone be equipped with more webcams choose the device you want to use and confirm your consent to the use of the same.

The program should begin immediately to show the acquired pictures in the “LIVE” image window or better the corresponding canvas[7].



Vediamo/Let's See
(1) Live Video
(2) Control Panel
(5) Artificial Pictrure
(3) Image Register 0 (empty)
(4) Image Register 1 (containing a grabbed image)

Figure-5) organization of **Vediamo/Let's See** rel. 1.0 user interface.

In the current version of the program **Vediamo/Let's See** offers the opportunity to choose from among ten different canvas of which: a permanently allocated for displaying "Live" (canvas number 1 in Figure 5), a window permanently reserved for the display of images painted and eight "image-registers" used to temporarily store the processed images.

In figure-5,

-The latest acquired picture is shown in the canvas corresponding to “Live” images(1),

- register 0 is empty then the corresponding *canvas* is closed (3),

- instead *canvas* 1 is open (4) so much so that you can see the image contained in it,

[7] In HTML 5 nomenclature are said *canvas* each of the rectangular display windows used to draw on the screen.

and it is also open the canvas corresponds to the image painted by the program.

For convenience of users to each of the image-registers has a button that allows you to save the scanned image in "Live" in the relevant register (3).

Continue … Work in progress...