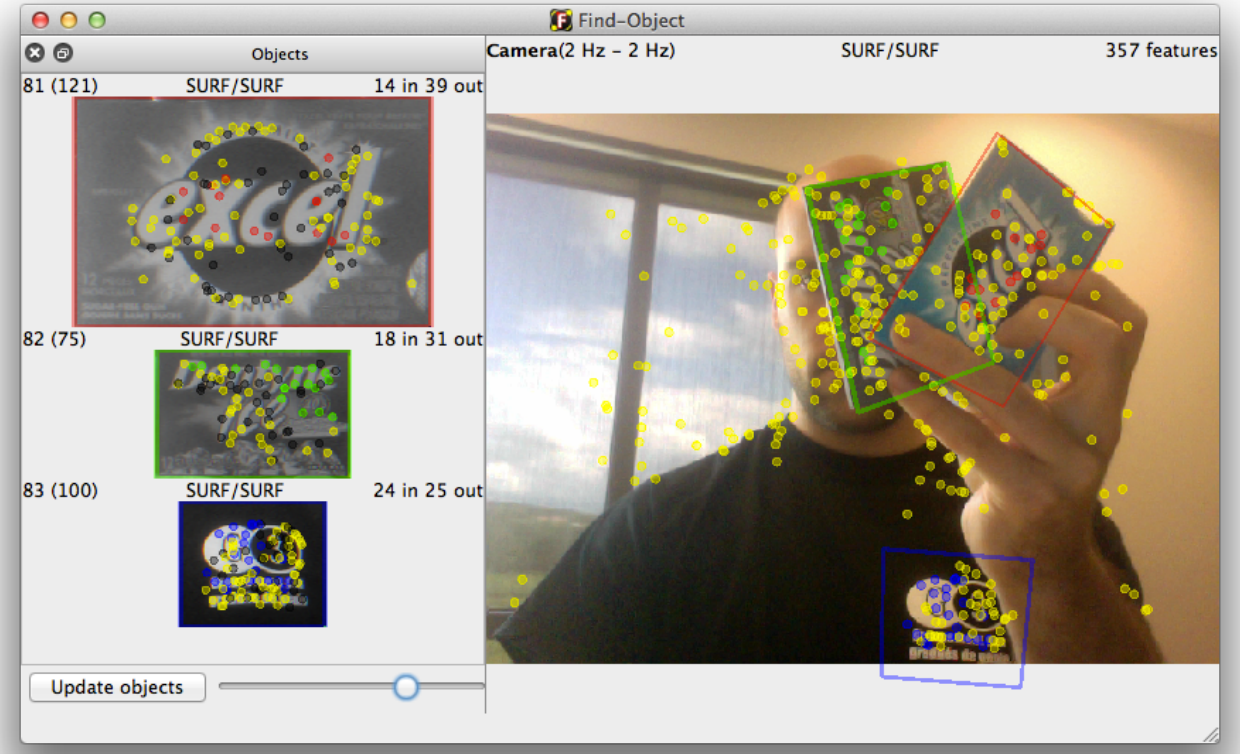
FindObjectsWithWebcam
Find objects with a webcam

Updated Jun 5, 2014 by matla...@gmail.com

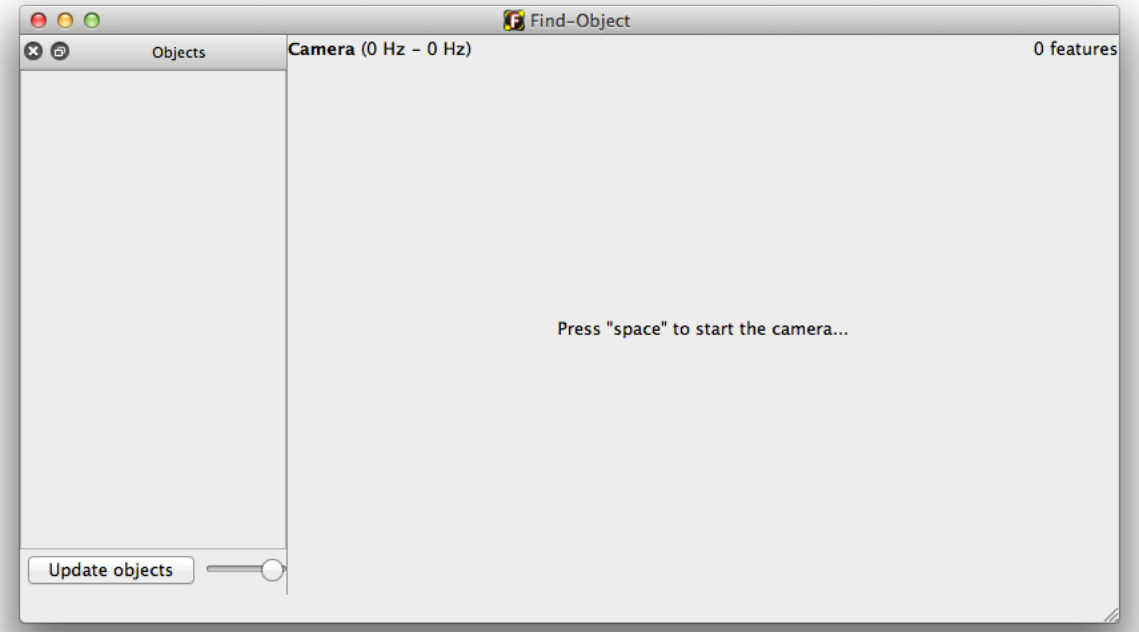
Introduction

This tutorial will show how to add an object using your webcam. Note that version 0.4 is used here.

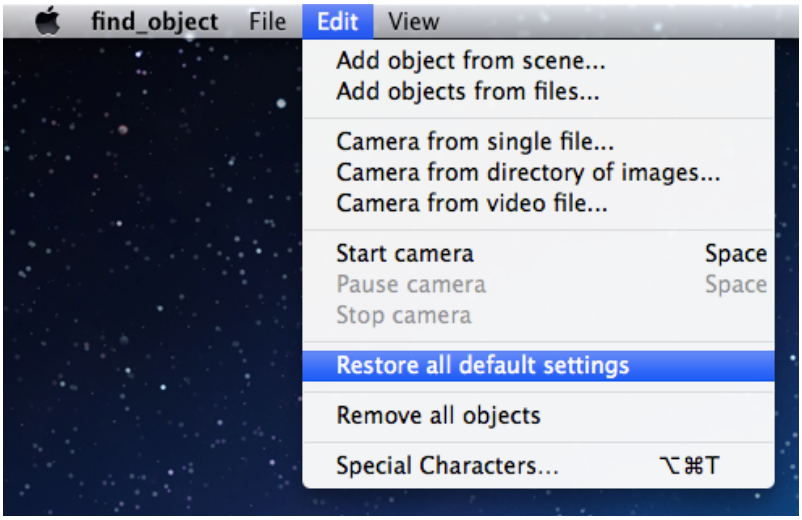


Details

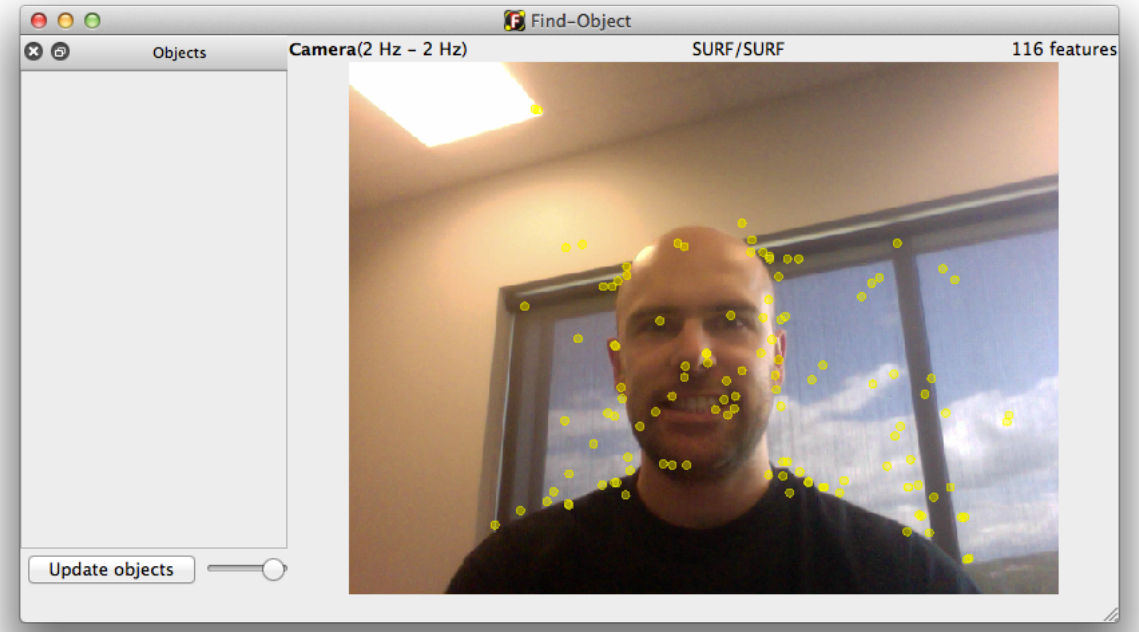
1. Open Find-Object application.



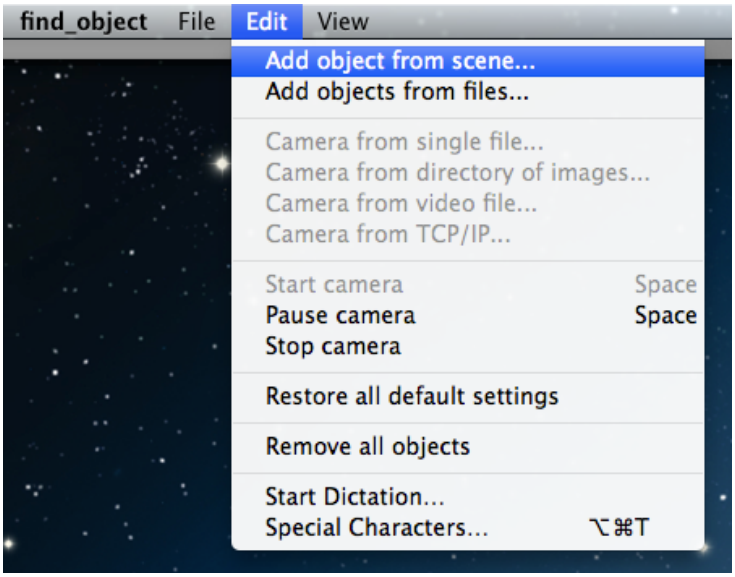
2. If it is the first use of Find-Object, go to next step; otherwise, restore all default settings (menu "Edit->Restore all default settings")



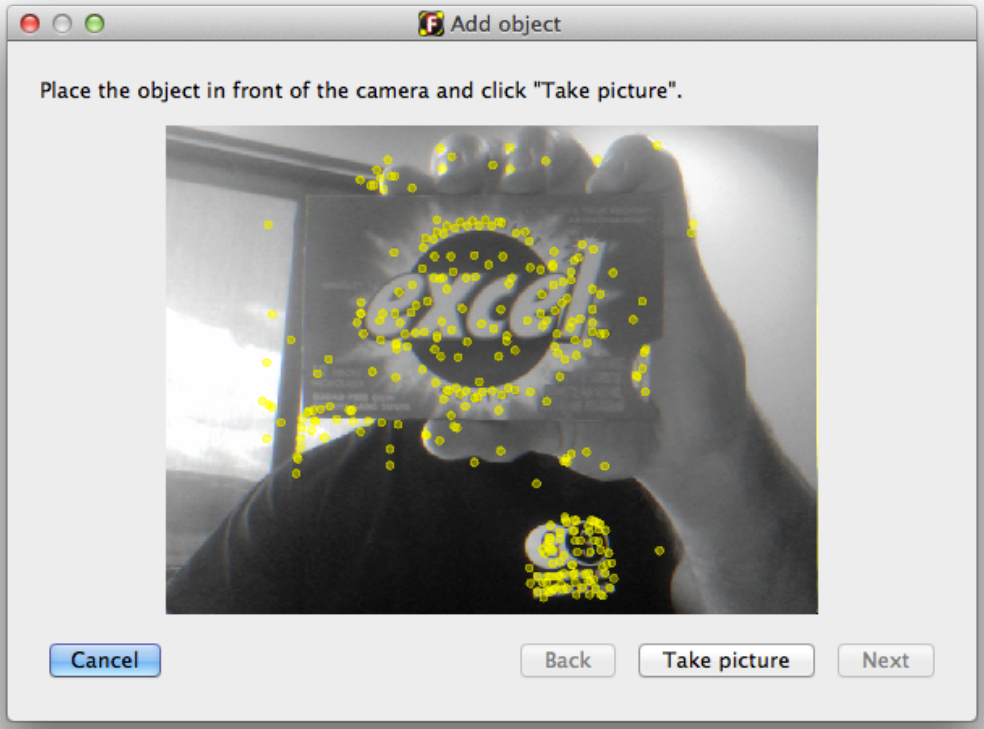
3. Press space to start the webcam. Note that by default, the view is mirrored (you can change this with a right-click on the scene or by changing the parameter "General/mirrorView" using menu "View->parameters").



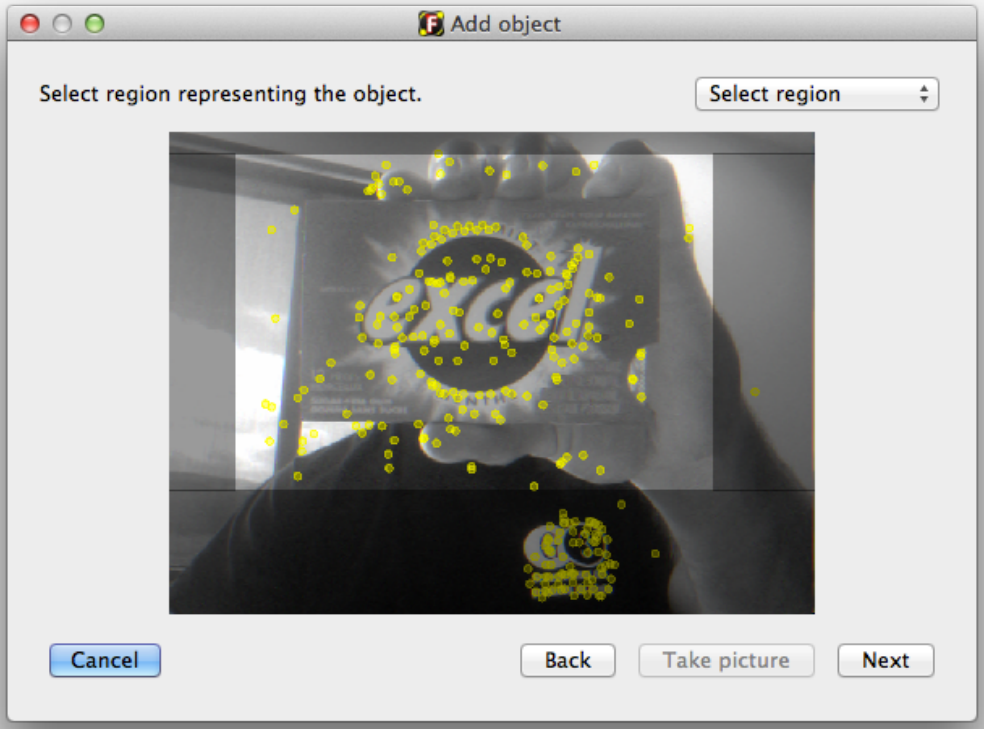
4. Open "Edit" menu and select "Add object from scene...".

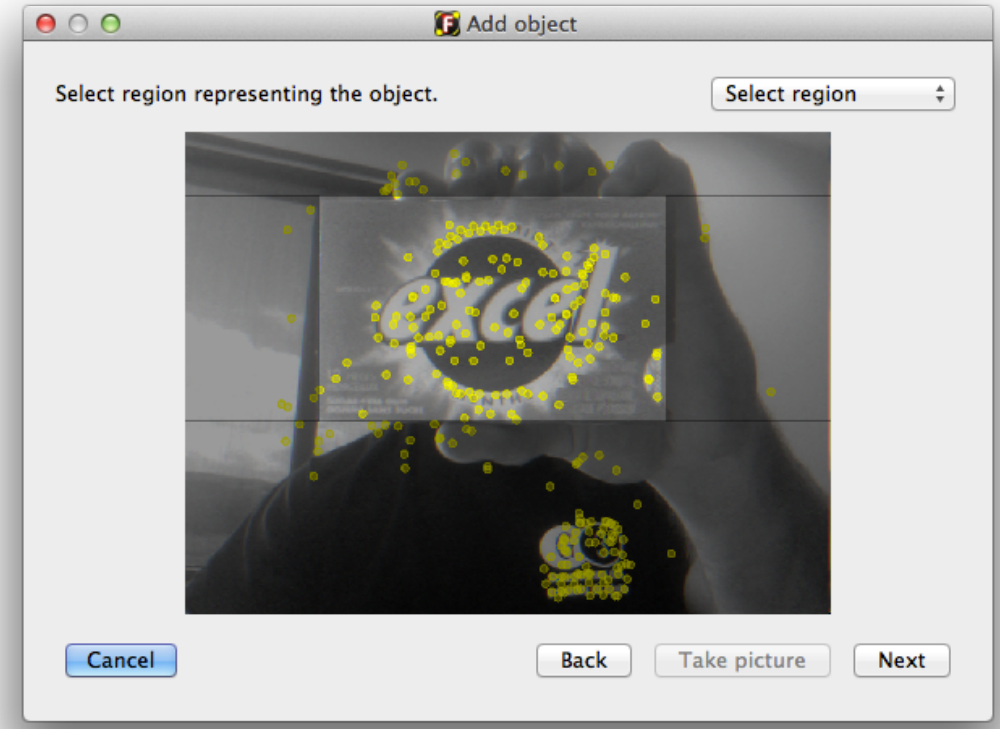


5. Place an object in front of the camera and press "Take picture".



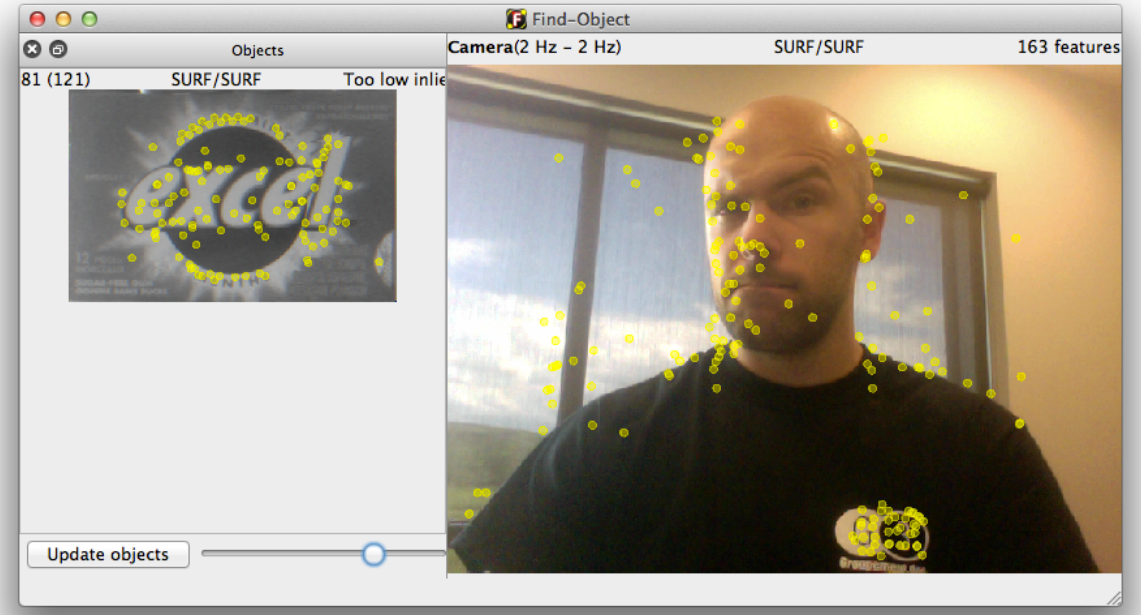
6. Select the region representing the object.



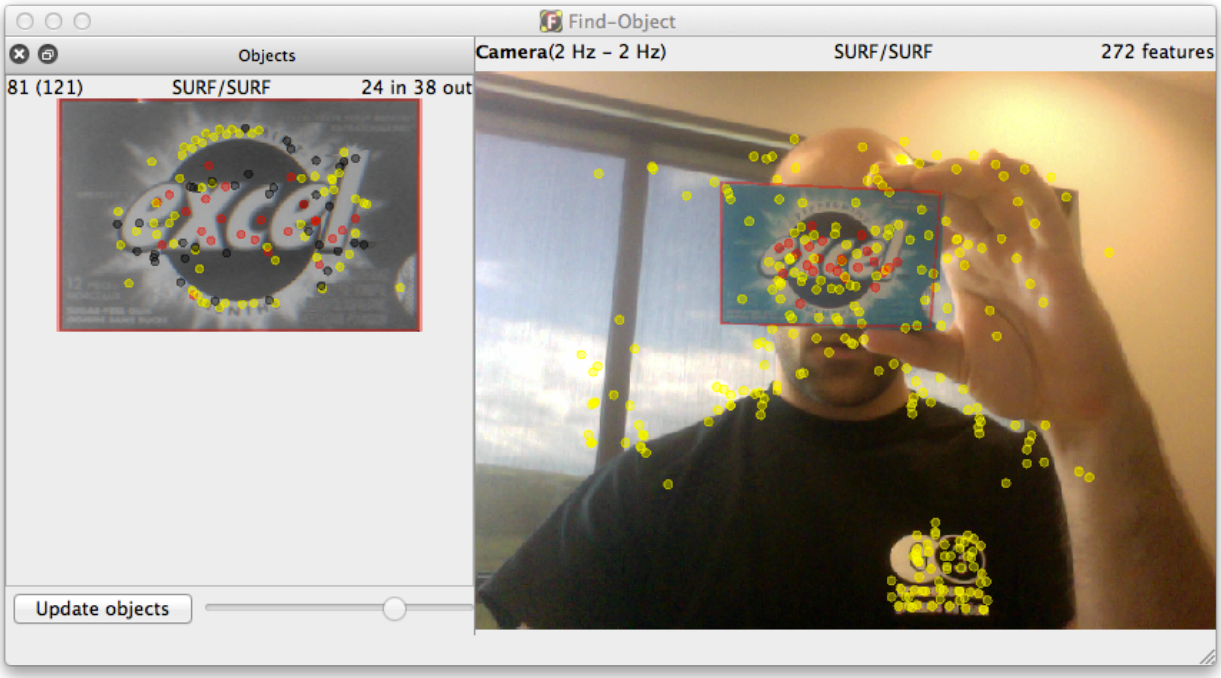


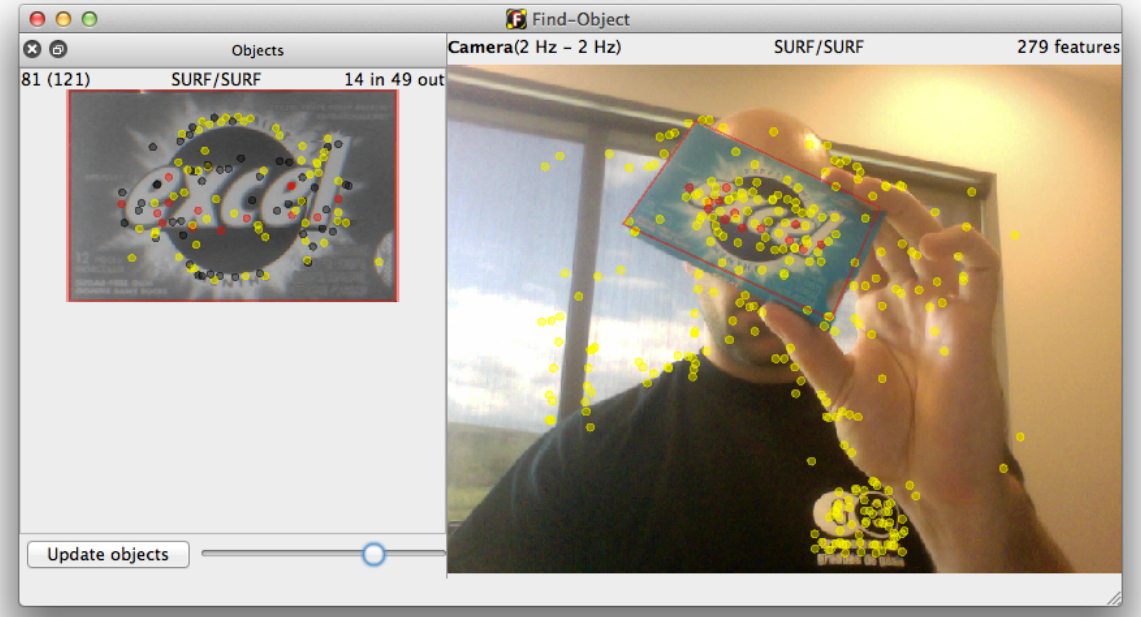
7. Press next to verify your selection. Then press "End".

8. Back to main window, you should see the object in the side pane.

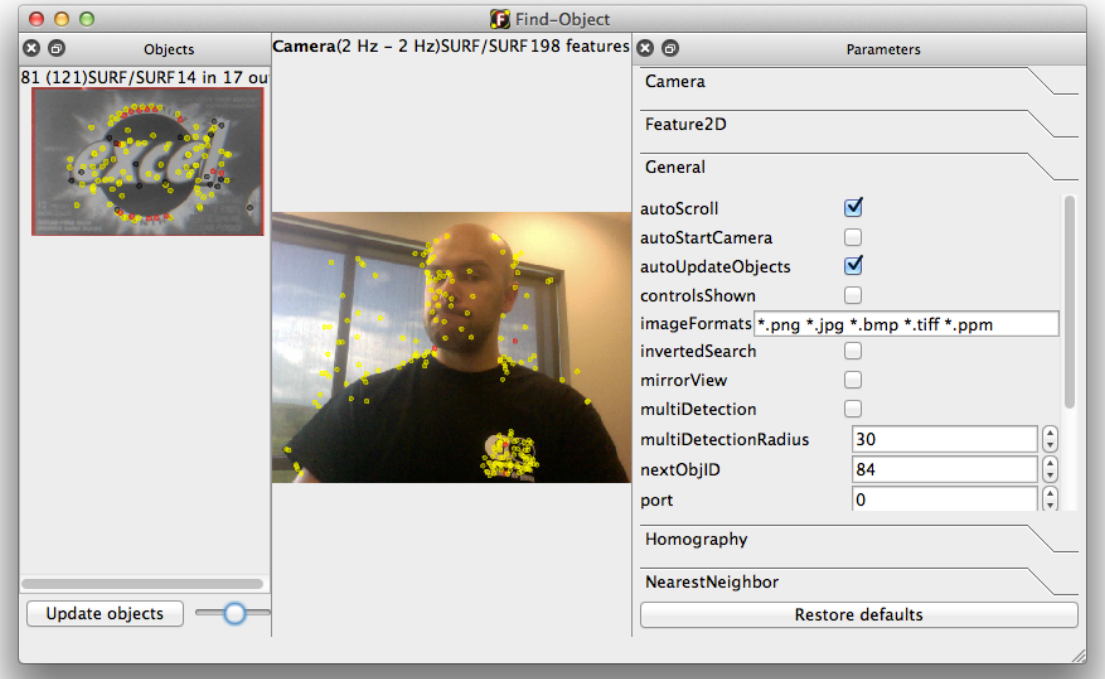


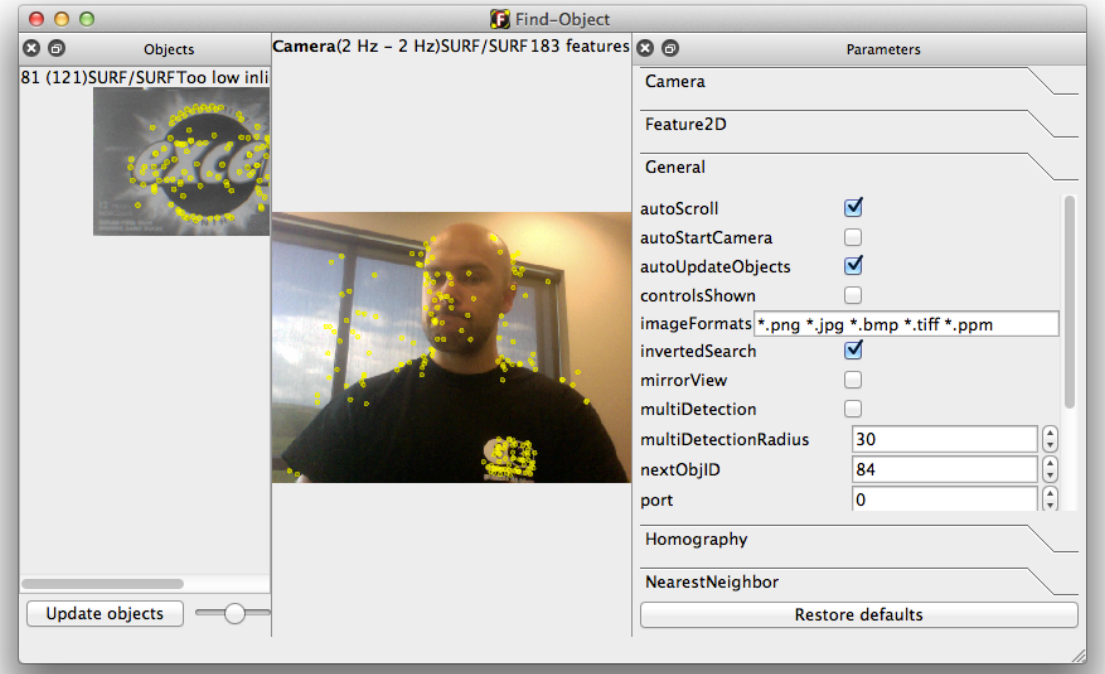
9. Put your object back in front of the camera to see the detection.



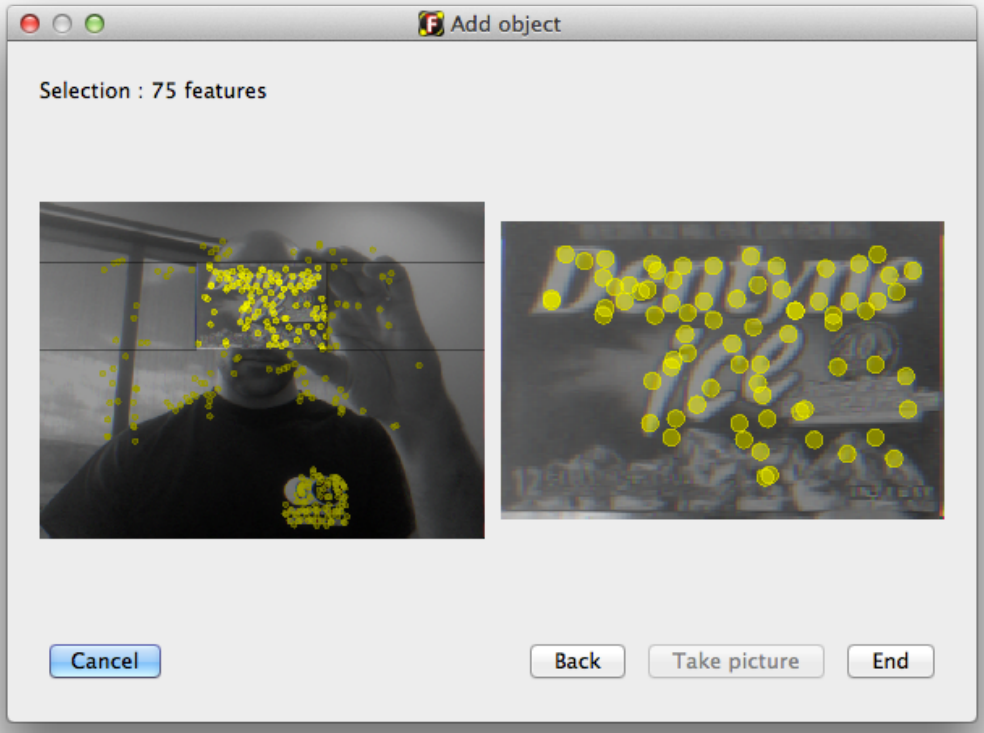


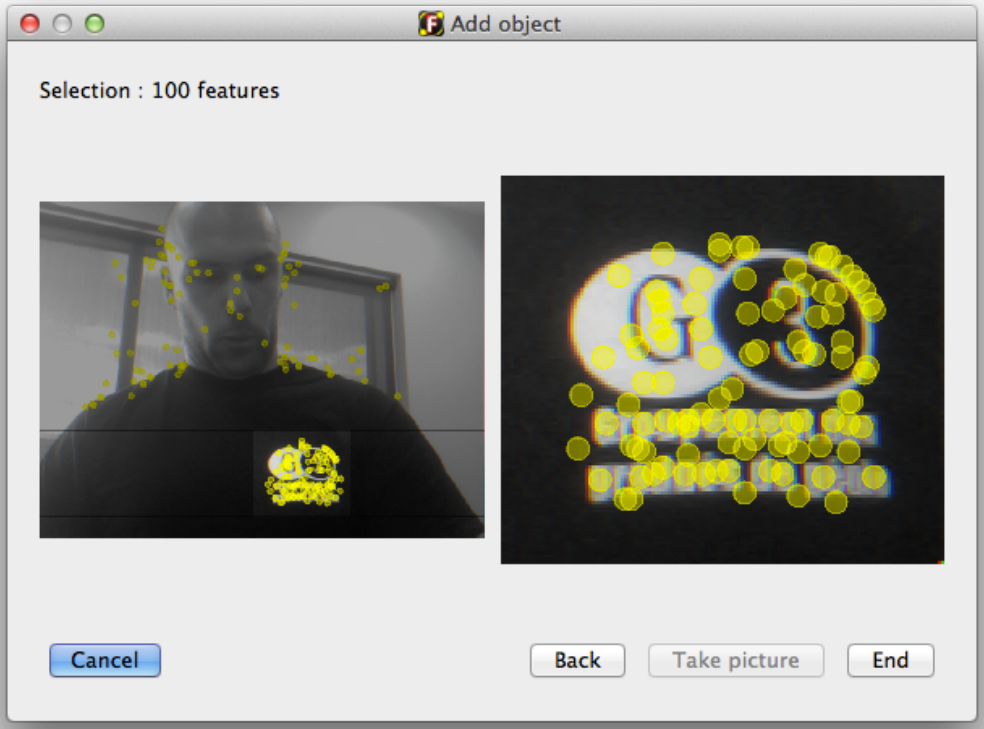
- If the scene doesn't have many features, there may have some bad detections like below. If your scene doesn't contain many features, try General/InvertedSearch mode. For the next, this mode is used.

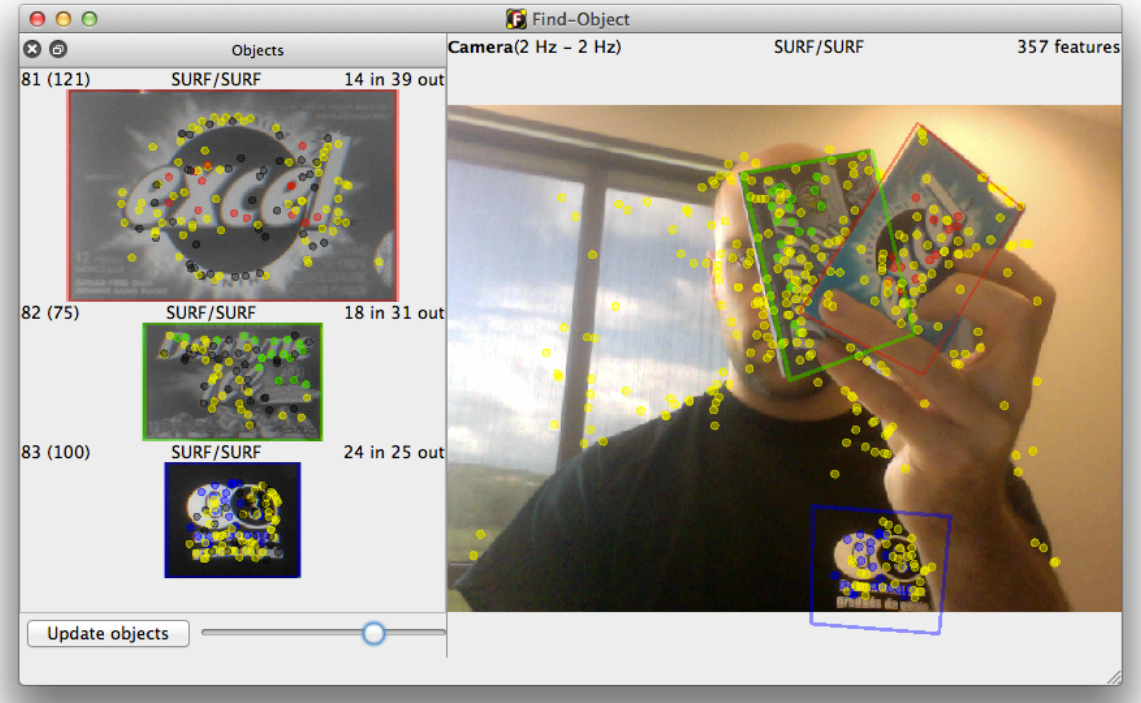




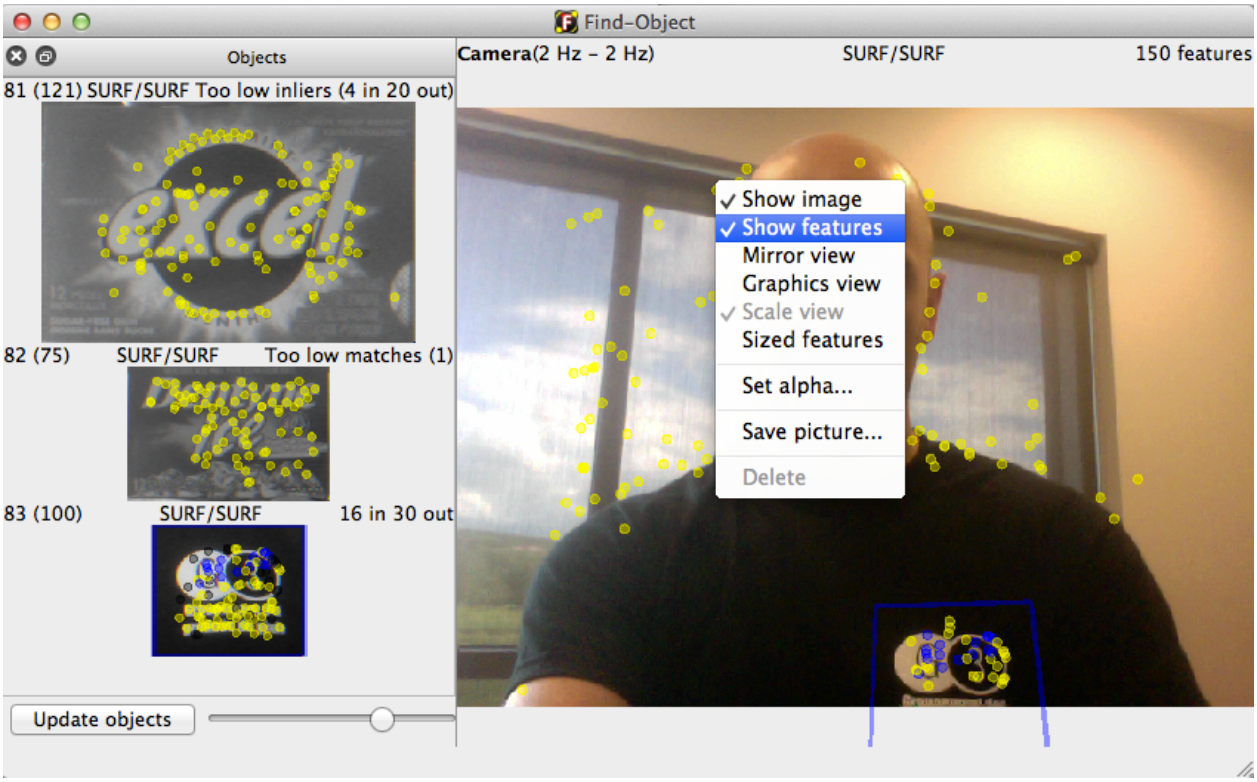
10. You can add more objects if you want.

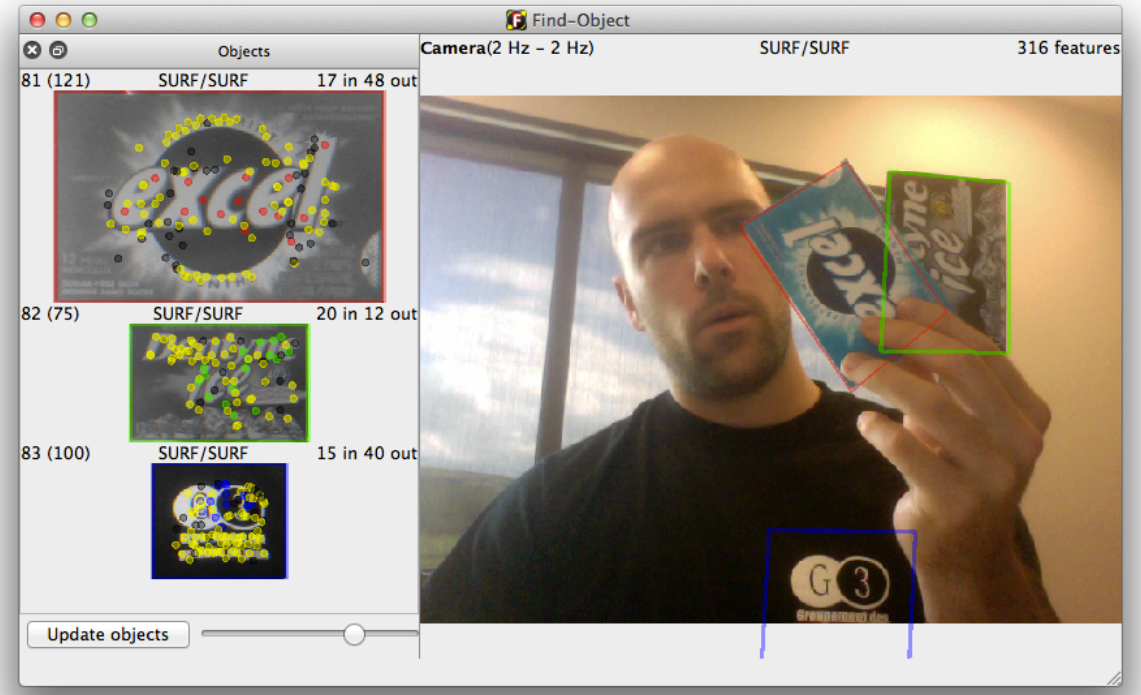






11. You can also right-click on the camera view and uncheck "Show features".





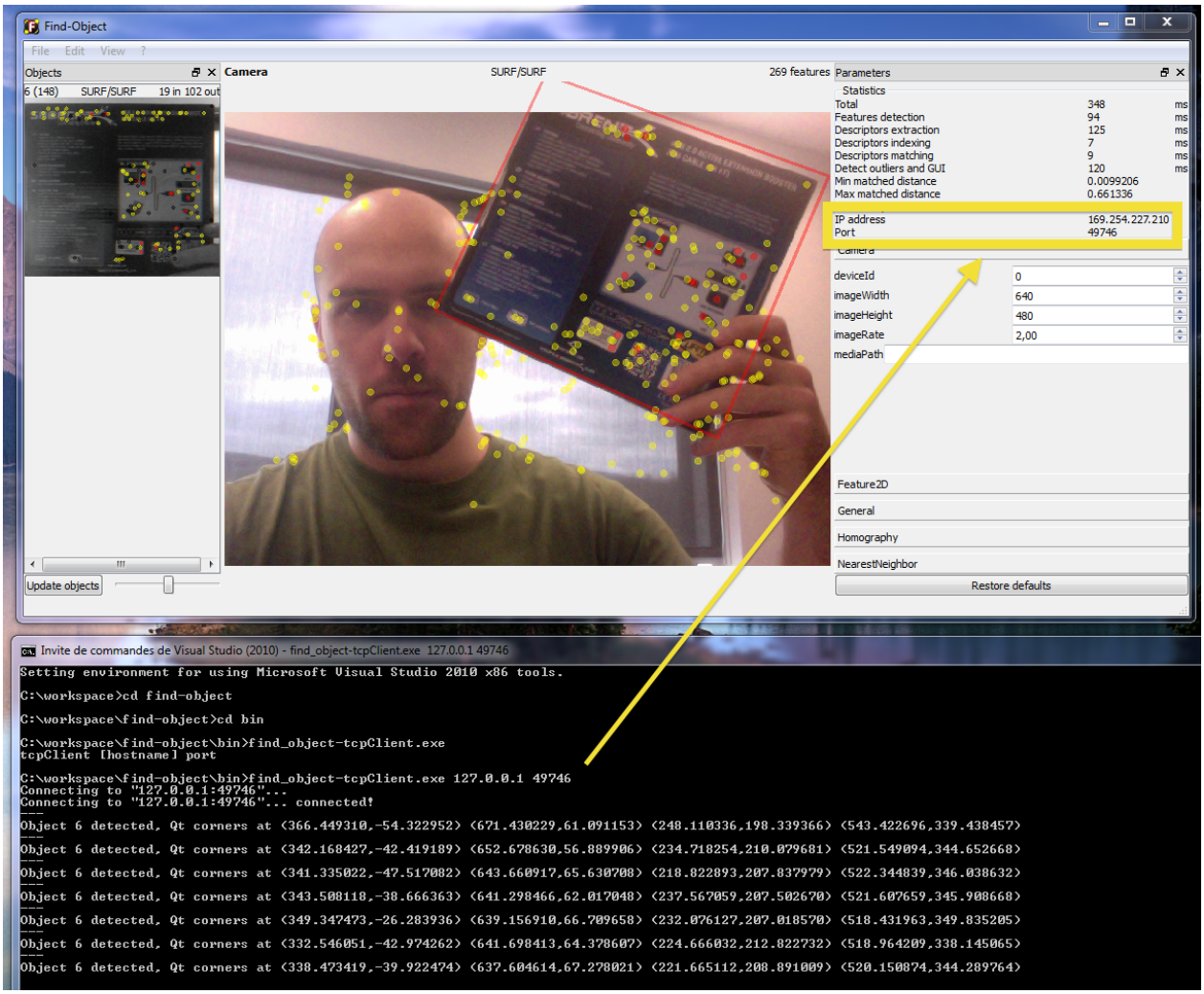
TCP information

Find-Object 0.4.3 required

Some information about objects detected like ID and position are sent over TCP on a port.

- Publish detected objects (with **position**, **rotation**, **scale** and **shear**), formatted as `[objectId1, objectWidth, objectHeight, h11, h12, h13, h21, h22, h23, h31, h32, h33, objectId2...]` where h## is a 3x3 homography matrix (h31 = dx and h32 = dy, see QTransform). Example handling the message (TcpClient.cpp, TcpClient.h and main.cpp).
- Raw format: [quint16 (size in bytes of the following QVector<float>) QVector<float>]. The QVector<float> should have a size of 12 float * number of objects.

- Note that parameter **Homography/homographyComputed** must be true (default true) in order to publish the topic.
- Parameter **General/mirrorView** would be false in order to visualize correctly the homography values returned.
- You can set manually a port, see parameter **General/port**.
- On **ROS**, a topic named "/objects" containing the same data is sent. See here.



Comment by mikezmab...@gmail.com, Oct 5, 2013

can it be applied to opencv using c#?

if so, please do send me tips and steps.

also, can you gave me codes for it..

Comment by AfnanAli...@gmail.com, Jan 26, 2014

hey just a question but how can we access the code so that we may add more of the yellow dots that recognize features?

Comment by kuh...@bluemail.ch, May 5, 2014

Hi! Wow, that's exactly what I need! How do I access the information (picture-number and -position)? Does the program give out the information on a port? Greetings

Comment by project member matla...@gmail.com, May 5, 2014

Sorry for previous comments, I didn't receive notifications when comments were posted, I just added a new rule in my google code settings. So here the answers:

@mikezmabulay If the .NET wrappers of OpenCV (like http://sourceforge.net/projects/opencvnet/ or http://www.emgu.com/wiki/index.php/Tutorial) expose the same methods that are used in the code here, it may be work in C#. For all GUI interface, Qt is used here.

@AfnanAli847? The number of yellow dots depends on how many features are extracted (from the chosen keypoint detector). In the code, ObjWidget?::setKptColor(int index, const QColor & color) is used to set the color of specified features.

@kuhn.f@bluemail.ch For now, only the ROS version of the code provides this output (objects detected with position). See https://code.google.com/p/find-object/#ROS. Though since many persons are using the code on Windows/Mac OS X/ Ubuntu without ROS, I added a new issue to add this feature (https://code.google.com/p/find-object/issues/detail?id=19).

Comment by ii...@yahoo.com, May 13, 2014

thats not at all the given object you are recognizing garbage

Comment by project member matla...@gmail.com, May 14, 2014

If you have problems with false detections, try to increase the parameter Homography/minimumInliers to 20 or more. You can set also Homography/ignoreWhenAllInliers=true.

Comment by Ryan.K.C...@gmail.com, Dec 8, 2014

I'm using somrthing like this for a science experiment at school and need some help. I've got it all working but I'm not very good at codeing :/ Is there a way that some codeing can be incorporated that would force the webcam to take a picture of an object when that specific object is placed in front of the camera? I've been working on this for a little over a month now, and any help would be appreciated!

Comment by project member matla...@gmail.com, Dec 9, 2014

It is possible. In the code, you could add this code in the `if(info.objDetected_.size() > 0)` at this line:

```
 if(info.objDetected_.size() > 0 || Settings::getGeneral_sendNoObjDetectedEvents())
 {
    Q_EMIT objectsFound(info);
+
+   QString path("screenshot.png");
+   ui_->imageView_source->getSceneAsPixmap().save(path);
+   UINFO("Saved %s!", path.toStdString().c_str());
 }
```

If you don't want to edit the code, you may use the TCP detection above to know when to trigger a screenshot on the camera (See "Example handling" above to know how to do a TCP client). An example would be to use in the TCP client a cv::VideoCapture? of OpenCV and to use cv::imwrite() to save the image when the TCP msg is received.

Enter a comment: Hint: You can use Wiki Syntax.

Submit