Grape Detection in Vineyards

Final project by

Ishay Levi, Eran Brill

Introduction

In our work we detected areas that are likely to contain grapes as well as areas which don't. We characterized areas based on the given database and came to several conclusions, Areas that contain grapes have:

- High (but not an extreme) gradient magnitude
- Average brightness
- Round edges

On the other hand, areas that don’t contain grapes have:

- Extreme gradient values (either very low or very high)
- Extreme brightness values
- Non green RGB values

Therefore we decided to clean up the picture from unwanted areas and then use Hough transform to detect edges that are circles or have common curvature.

Approach and Method

In every filter we use we don't check individual pixels, instead we scan the picture with a window. We take measures of the window regarding his average gradient size and the average of the RGB values

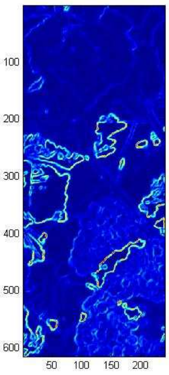
1. Statistical analysis

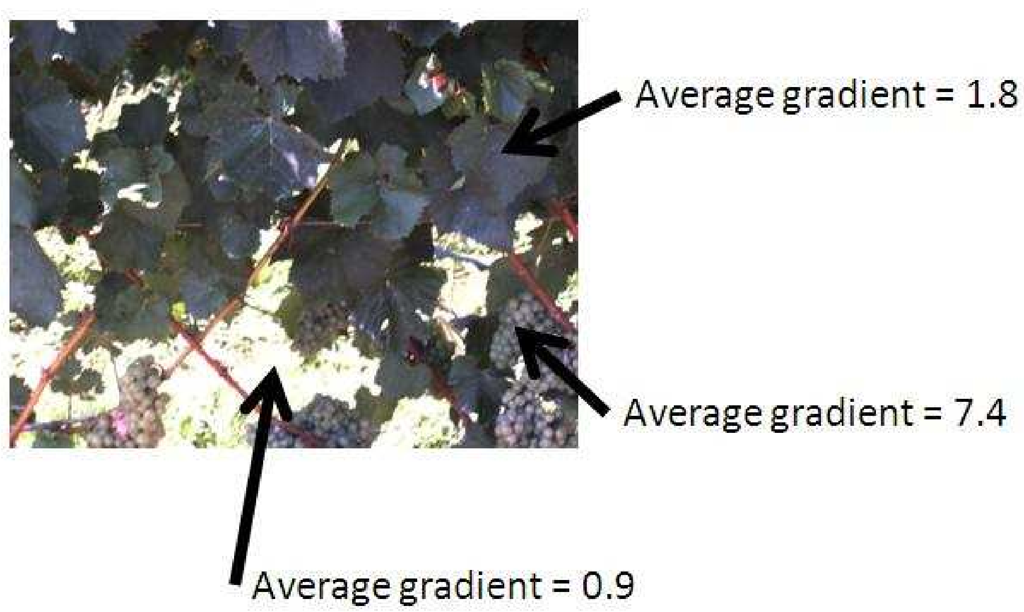
In order to diagnose the specific characters of grapes and non-grapes areas we analyzed many images in order to achieve a statistic overview on various areas. We gathered information regarding:

- Gradient magnitude
- RGB values
- Brightness

1.a Gradient magnitude:

We found that around grape areas the gradient magnitude is high with an average of 9 in the bright pictures and 6 in the dark ones. Non grape areas like leafs have low gradient magnitude, less than 2. So we filtered the picture based on this character leaving only pixels with gradient magnitude between 2 and 35.



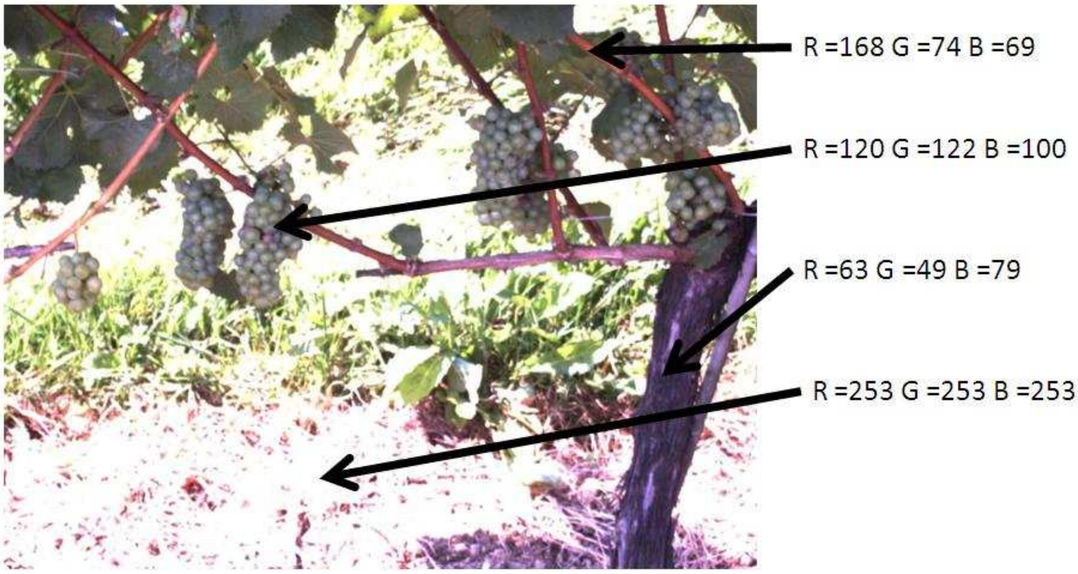


1.b RGB values:

Naturally areas with grapes have a lot of green in them, but the color of a pixel is also determined by the red and blue value, in order for the pixel to be green or in the shade of green the green value has to be higher or very close to the red or the blue value, after analyzing areas such as:

We decided to erase all areas that have one of this terms:

Blue average – Green average > 20
Red average – Green average > 20
Red average – Green average > 10 and Blue average – Green average > 10



1.c Brightness:

After many evaluations we found that generally grapes have below average brightness, so we erased all pixels that have brightness above 120% of the general average of the picture. This is a drastic step, but it was found useful.

2. Hough transform

The final step of the grape recognition function is to distinguish the grape like shapes sections from the other picture objects. After clearing most of the "noise" (=non grape sections) using color and gray level factors, we will be using a method based on the shape of the objects on the picture.

By using Hough transform for circles and ellipses shapes. We can recognize grape like shapes in a picture.

As a result, one sub-goal of our project was to implement a suitable Hough transform function that would take a black & white edge map, and by using the Hough transform will recognize circular and ellipses in the picture where the ultimate goal is to reduce the number of non-circular shapes.

General description of Hough transforms (by icvb091 lectures):

Given:

List of edge points (arbitrary order)

Compute:

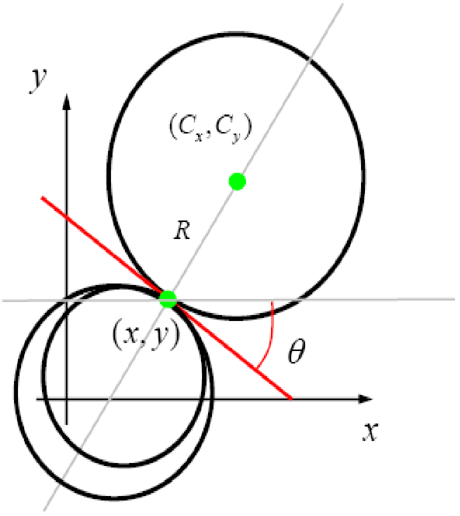
Set of straight lines in the edge map.

Basic idea:

1. Let each edge point vote for all lines it may belong to.
2. Lines with lots of votes "win"

And for circles (algorithm was edited to fit our requirements):

1. Use the parametric model for the geometrical structure:



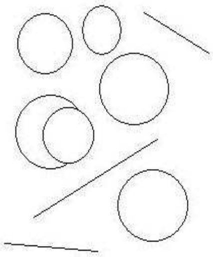
$$C_x = x + R\sin\theta$$
$$C_y = y - R\cos\theta$$

2. Quantize the parameter space appropriately into bins.
3. Initialize each bin to zero.
4. For each point (x,y) in the image space, vote (e.g., add 1) to all parameter
5. Bins that satisfy the model equation. (where R runs in a pre-fixed range)
6. Check for each circle center for most voted radius.
7. Check in edge map and use only edges of circle found in the transform.

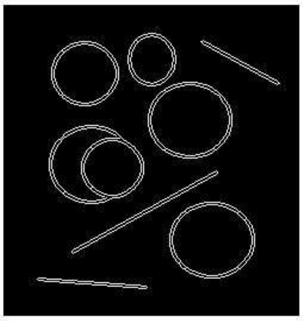
We implemented the above algorithm (**houghcircle.m**). Using statistical analysis of most of the sample pictures we calculated common radius for grapes in the picture.

Running examples of our implementation for reducing non-circular shapes:

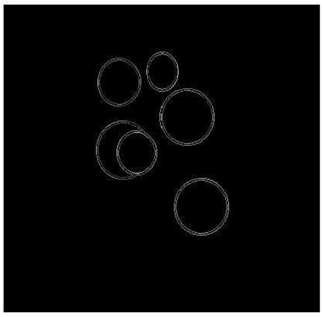
1. Given a Simple shapes picture:



sample picture

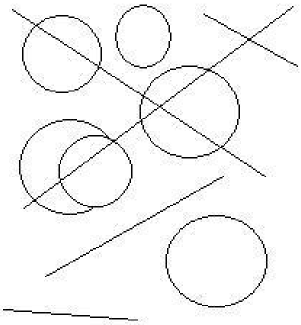


canny edge detector

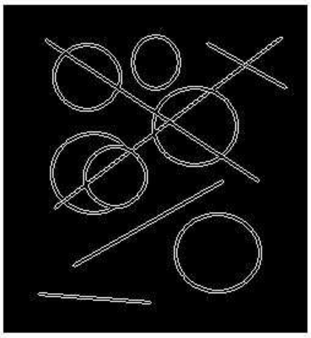


using Hough transform and removing non- circular shapes

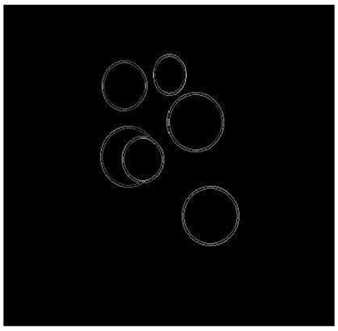
2. Inserting interfiring lines doesn't have a significant effect on the results



sample picture

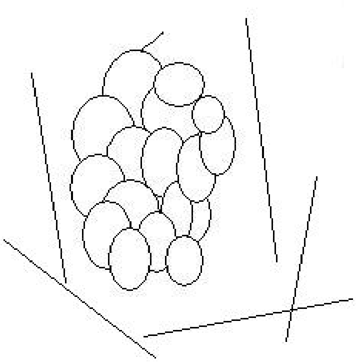


Canny Edge detector

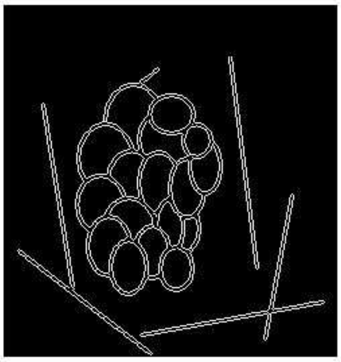


using Hough transform and removing non- circular shapes

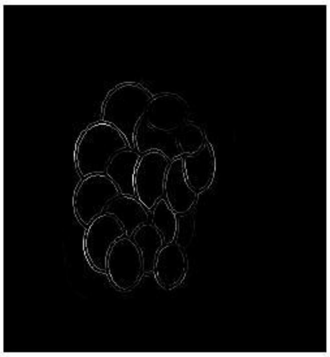
3. Results for Grape like shapes:



sample picture

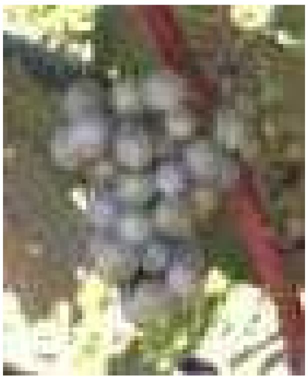


Canny Edge detector

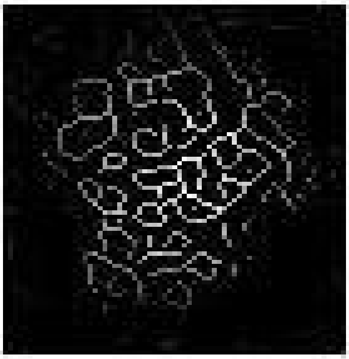


using Hough transform
and removing non- circular shapes

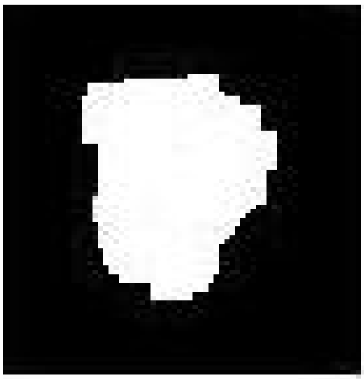
4. Using same algorithm in real-life picture for only grapes:



sample picture



Canny Edge detector +
Hough transform +removing non-
circular edges



Highlighting only "strong" areas

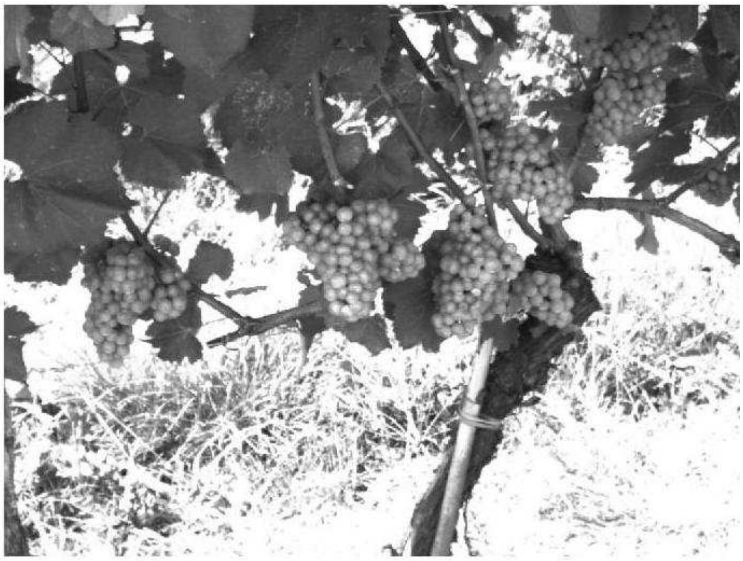
As oppose to the simple usage on sample pictures, the usage in the project pictures is more complex. Due to the existence of many circular shaped objects (leaves, grass, branches exc.), Running a canny edge detector on a full gray-level picture will produce many edges of circular shaped objects. As a result, we use this technique on the picture after non-grape areas were removed by the filters described above. Then we search circular shapes with radiuses that we discovered by statistical analysis and remove the rest.

Selecting proper edge detection

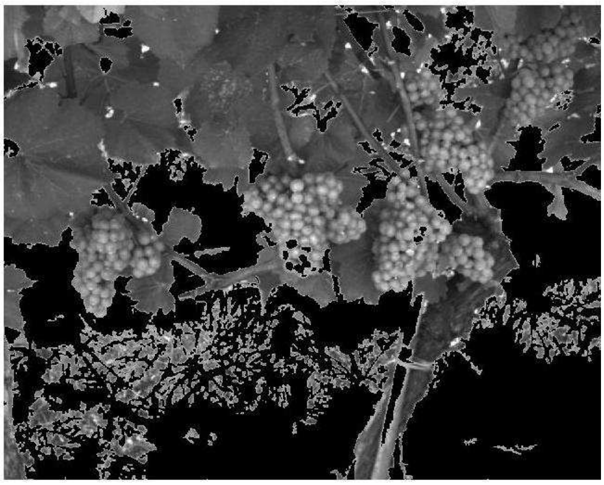
Detecting shapes relay on edge detection, deciding on best edge detector (and its parameters) is by many means trial and error.
This should be depended by input, as dark pictures require edge detector with ability to detect very weak edges and vice versa.
The tradeoff of using too delicate edge detector appears in the final results, as many, circular edge suddenly appear in the most peculiar places. This non-grape areas will than appear to the algorithm as circular objects, and there for will not be deleted.
Of course using too "raw" edge detector will result in not detecting the edges that define the grapes.
We have found that using canny edge detector with delicate threshold (on both strong and weak gradient parameters), produces the best results.

Results

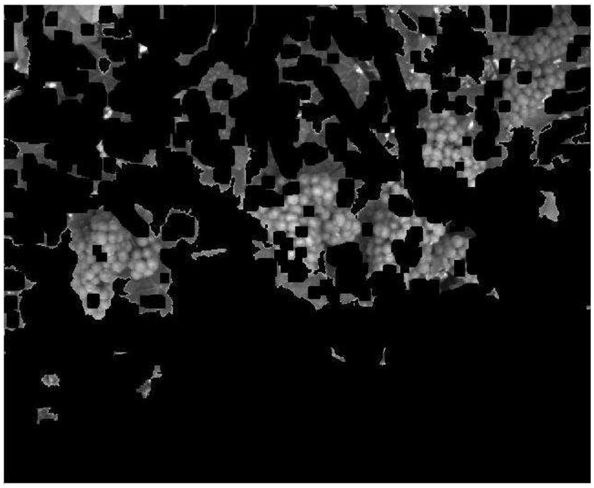
In result section, we will introduce results of our suggested solution over real vineyard pictures.



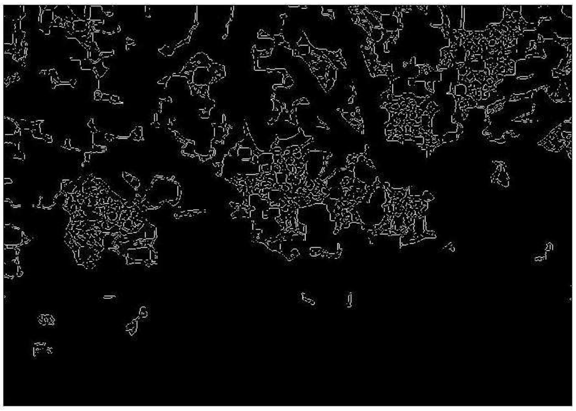
Ex1. Original picture



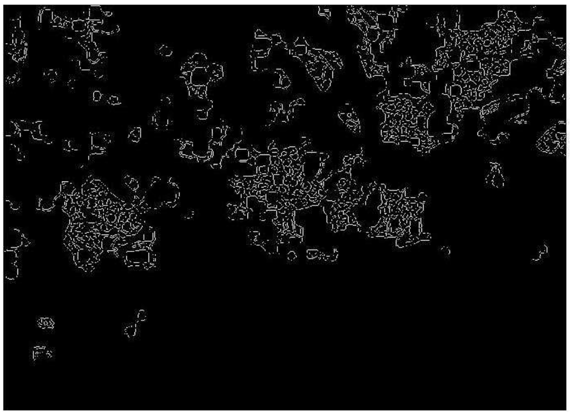
After brightness filter



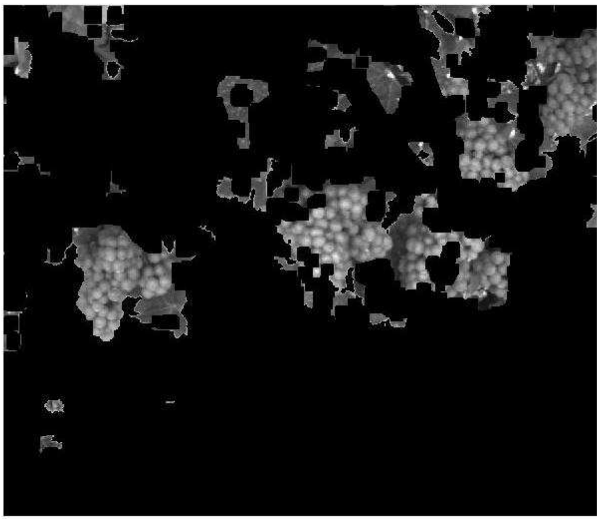
Gradient+ RGB filter



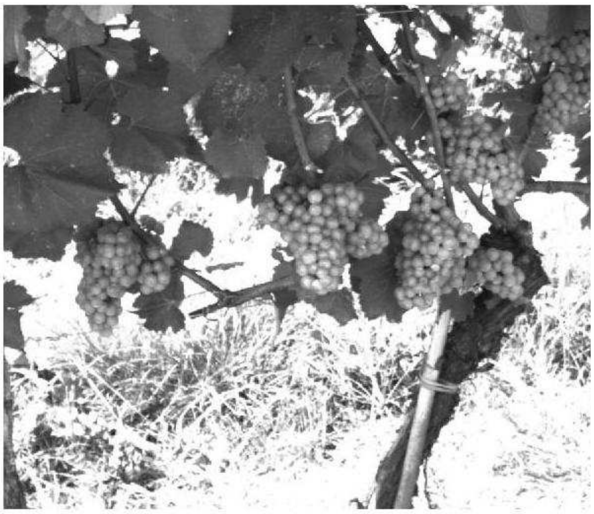
After canny Edge detection



After reducing non-circular shapes



Ex.1 Final solution



Original picture

Conclusions

Our results differ from one image to the other, but in general, using the techniques above, we detected and removed many areas that don't contain grapes.
Large percentage of the grapes "survived" the statistical removal of non-grapes area but there is a large percentage of noise that needs to be removed.
Using statistical analysis proved, in many cases to be quit efficient, and can be supported by large database, that can be produce suitable data according to the conditions, i.e. winter or summer times, time of photographing, and even grape type and age, those matching, of conditions for used values, can improve the results.
In this assignment we tried to produce fit–all variables to be suited for most pictures.

The usage of the Hough transform algorithm has improved results, but its dependency on many variables (i.e. radiuses ranges, edge characteristics) makes it vulnerable, and those only used as the final filter.

References

http://www.cs.bgu.ac.il/~ben-shahar/Teaching/Computational-Vision/

Hough Circle Transform- Harvey Rhody, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, October 11, 2005
http://www.cis.rit.edu/class/simg782/lectures/lecture_10/lec782_05_10.pdf

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT6/node3.html

RGB applet
http://www.lon-capa.org/~mmp/applist/RGBColor/c.htm

AN OBJECT SEGMENTATION APPROACH BASED ON UNITED COLOR MODELS FOR SOCCER VIDEO - YAO-QUAN YANG , NA LIU, WEI GU, XIN-YAN LIANG, Faculty of control science and engineering, North China Electric Power University, China
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04028763