EE 584 Machine Vision Term Project Report

Stereo Snakes: Contour Based Consistent Object Extraction for Stereo Images [1]

Introduction

Although the history of stereo imaging goes back to 1850s [2], it has become popular in the 2000s. With the increasing popularity of the stereo images, stereo image editing tools need to be added to software like Adobe Photoshop. Since the computational power needed is almost twice for processing two images, special methods need to be developed in order to process stereo images more efficiently. Moreover, because stereo image pairs consist of two views which are nearly duplicate, one can achieve developing such methods using the consistency between them.

The aim of this project is to both present a good understanding and a successful implementation of [1]. The method proposed in [1] achieves segmenting an object in one of the stereo image pair, given the object contours in the other image. To do this, it uses both stereo correspondence cost and object boundary cost to generate an energy function and tries to minimize this energy function like it is done in the Snakes method [3]. Thus, the method is named “Stereo Snakes”. The authors of [1] claim that

Main Energy Equation

According to [1], in order to segment an object from both of a given stereo image pair, one can first use any segmentation algorithm such as GrabCut for segmentation of one of the images, and use the result of this segmentation for the segmentation of the second image. Using the energy function

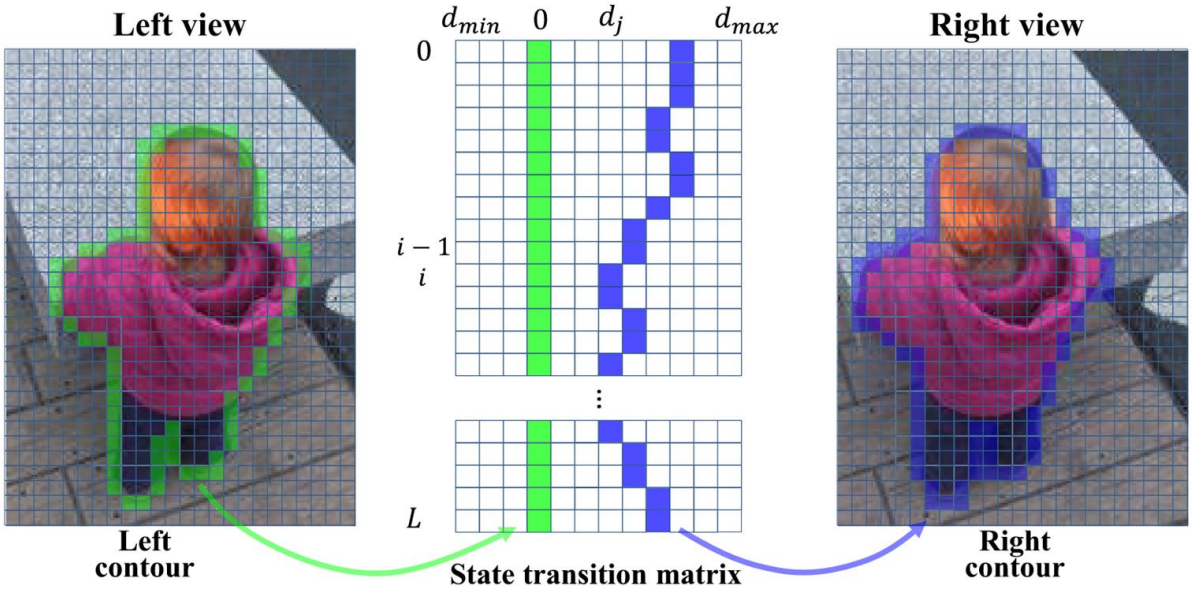

Figure 1: Illustration of contour correspondence algorithm [1].

given in eqn. (1), one can use an algorithm similar to well-known Snakes algorithm which was proposed by Kass et. al. [3] to find the contours that belong to the object.

$$E(d) = \sum_{p_i \in \mathcal{C}_m} C_S(p_i, p_i - d(p_i)) + \lambda_o C_O(p_i, p_i - d(p_i)) + \lambda_s N(p_i, p_{i-1}) \quad (1)$$

The algorithm proposed by [1] simply tries to obtain a smooth contour which simultaneously minimizes the energy function given in eqn. (1). In the eqn. (1), E(d) is the objective energy score, p_i is the contour pixel which is found by any image segmentation algorithm, p_j is the corresponding pixel of p_i in the other image, i.e. $p_j = p_i - d_i$ where d_i stands for the disparity of p_i. Cs function represents the stereo correspondence cost, Co function represents the object boundary cost and N represents the smoothness (or continuity). Note that the disparity d_i is restricted in the horizontal direction because of the epipolar geometry.

Stereo Correspondence Cost

The stereo correspondence term Cs is used for determination of pixel-wise matching cost. Using eqn. (2), one can obtain pixel-wise absolute color difference. In order the algorithm to be more robust, [1] says that aggregation of the eqn. (2) in a local window is needed. So, eqn. (3) is obtained.

$$C_{AD}(p_i, p_j) = \sum_{h=\{R,G,B\}} |c_h(p_i) - c_h(p_j)| \quad (2)$$

$$C_S(p_i, p_j) = \sum_{p_x \in \Phi(p_i) \wedge K(p_x)=1} C_{AD}(p_x, p_y) \quad (3)$$

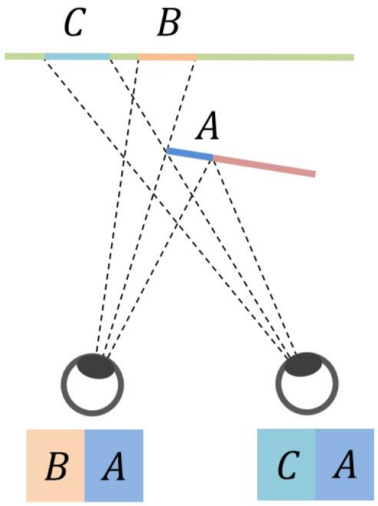

Figure 2 Background Occlusion: There might be local inconsistencies in stereo images due to binocular disparity. For example, in this figure, although the object parts are seen as the same, background seems to be different in different images. The left camera sees background "B" while the right camera sees background "C". [1]

Where $p_y = p_x - d(p_i)$, $\Phi(p_i)$ is a local window where the eqn. (2) is aggregated. $K(p_x)=1$ means that only the object pixels are considered. So basically, a pixel in the right image (or image with the unknown object contour) is compared with the pixels which belong to the known object in a local window. By doing so, Cs overcomes the problem of inconsistency due to binocular disparity as shown on Figure 2. Figure 3 shows step by step application of Cs, the cost image obtained using Cs and the result obtained when only Cs is considered.

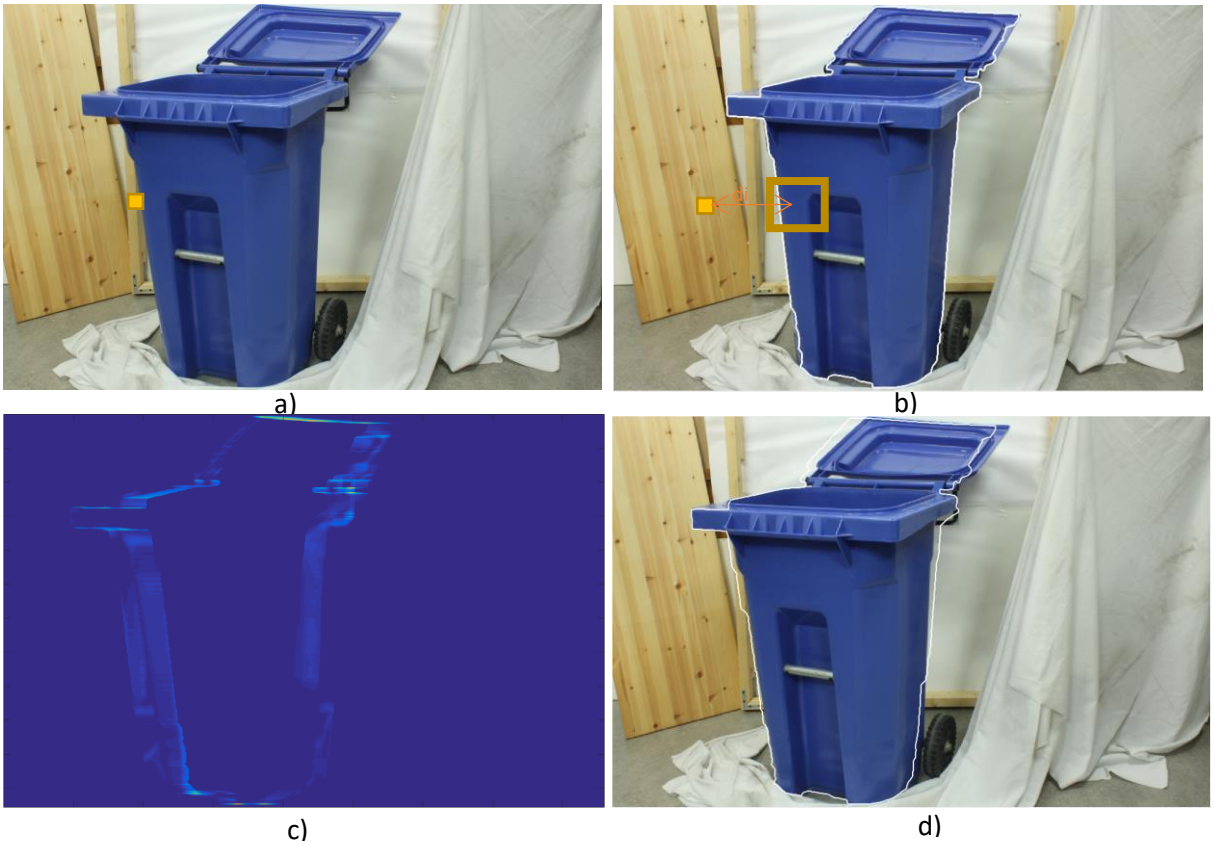

Figure 3 (a) Right image and a pixel of interest py as shown by yellow square. (b) Left image and the px pixels that corresponds to py pixel in (a) which are in a local window shown by bigger yellow square. (c) Stereo Correspondence Cost obtained (c) The resulting contour shown on the right image.

Object Boundary Cost

The object boundary cost is used for overcoming the self-occlusion problem, which is shown in Figure 4. In self occlusion, a part of the object may be invisible to one of the cameras due to the object itself. Therefore, the histograms of both background and object of the image with the object mask provided are calculated. Using these histograms, one can obtain a probability map which determines each color value's probability of belonging to an object or to background.

$$C_O(p_i, p_j) = \sum_{p_x \in \Phi(p_i)} |Pr(O|p_y) - K(p_x)| \tag{4}$$

$$Pr(O|p_y) = \frac{H_O(c(p_y))}{H_O(c(p_y)) + H_B(c(p_y))} \tag{5}$$

For the determination of the posterior probability $P_r(O|p_y)$ of a pixel p_y in the right image to be object, eqn. (5) is used. H_O term in this equation denote the object histogram value for the

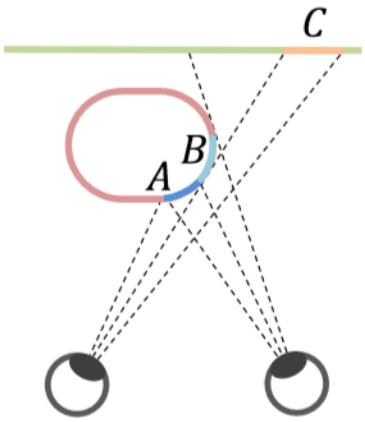

Figure 4 Self Occlusion: A part of an object may not be seen by one of the cameras due to the object itself. So the object seems different in different views. For example in this figure, left camera sees only the "A" part of the object while the right camera sees both "A" and "B" parts. [1]

corresponding color value while H_B term denote the background histogram value for the corresponding color value. It is important to note that 8x8x8 histograms are used for the calculations.

The term inside the summation in eqn. (4), $\left|P_r\left(O\middle|p_y\right) - K(p_x)\right|$ is low when p_x belongs to an object and p_y's possibility of being an object is high. Likewise, it is low when when p_x belongs to background and p_y's possibility of being an object is low. In other words, C_O gets minimum score when both object and background parts in a local window $\Phi(p_i)$ matches.

Figure 5 illustrates step by step application of object boundary cost. It also shows the result obtained when only C_O is considered in the energy function.

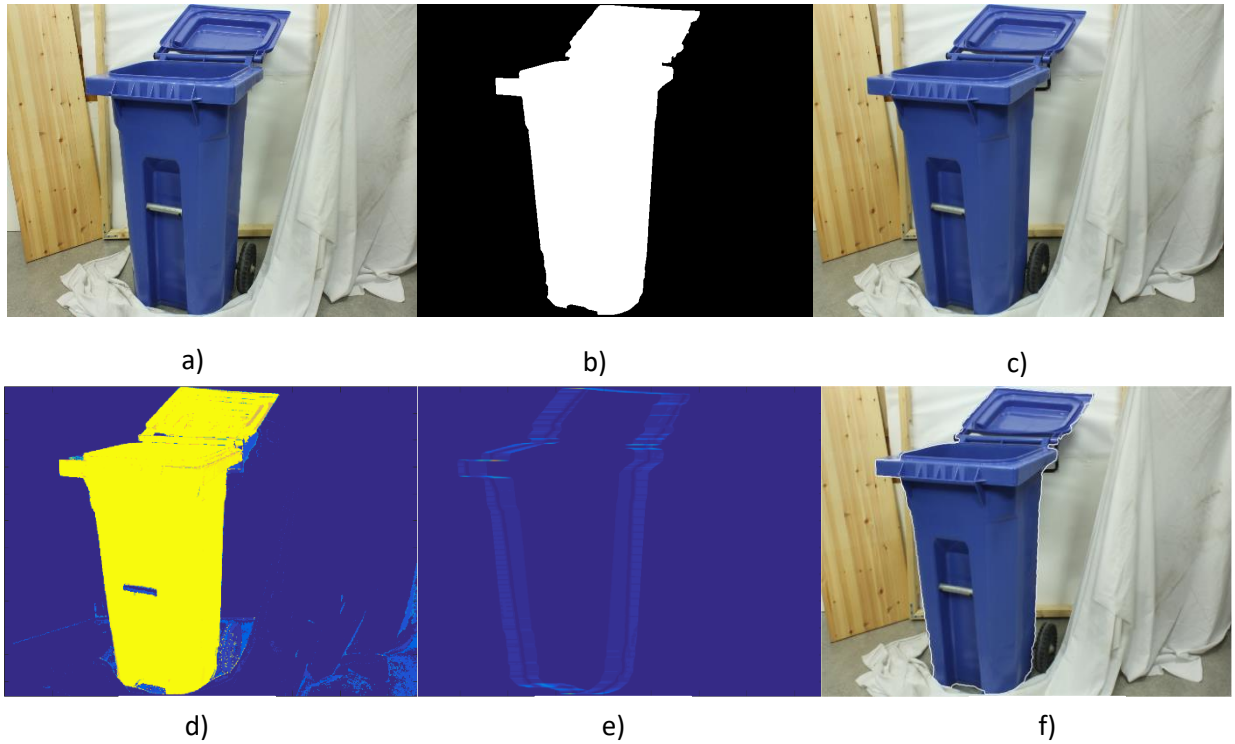

Figure 5 (a) The left image (b) Object mask for the left image (c) Right image (d) Posterior probability map for right image, calculated using eqn (5) (e) Object boundary cost calculated using eqn. (4) (f) Resulting object contours found using only object boundary cost.

Dynamic Optimization

The last energy contributor to the snake cost function, optimization factor (N), aims to smoothen the output contour and preserve the input shape, that is defined by input contour. The optimization is done by a single trace in disparity space.

For a single contour, N factor is defined as follows:

$$N(p_i, p_{i-1}) = \begin{cases} C_p, & \text{if} \quad |d(p_i) - d(p_{i-1})| \leq \tau_d \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

Here in equation (6), p_i and p_{i-1} stands for adjacent contour pixels. Whenever relative disparity (difference in adjacent disparities) is less than or equal to τ_d parameter, which is one of six user defined parameters; this disparity difference is assigned to N. On the other hand, N goes to infinity. Considering the fact that energy function is meant to be minimized, the cases of relative disparities greater than τ_d are ignored, in other words. This τ parameter sets a limit to permitted disparity difference between adjacent contour points.

Complete Algorithm Flow

In this section, how all the energy functions are implemented and dynamic tracing is done will be discussed.

First, state energy definition should be given, as below:

$$E_{stf}(i,j) = \begin{cases} E(p_i, d_j), & i = 1 \\ E(p_i, d_j) + \min\limits_{t \in [j-\tau_d, j+\tau_d]} E_{stf}(i-1, t), & otherwise \end{cases} \tag{7}$$

The initial energies $E(p_i, d_j)$ are only consists of stereo correspondence (Cs) and object boundary (Co) cost functions, the first two cofactor in the main equation (1). Procedure starts with assigning this energy to the state energy for the first contour pixel (i=1). Then, gradually, minimum of previous state energy is added to current energy to get current state energy, in τ-defined interval (will be better explained in the whole algorithm flow).

While doing so, dynamic optimization factor (N) will be added, too. Also, it is multiplied by another user parameter λs. This means that, if λs gets greater, then it is harder to relative disparity is bigger, harder to spoil the shape, harder to encounter discontinuities.

Finally, calculating all the state functions, a counter-clockwise (counter direction to previous steps) traceback is done in order to find minimum state functions, still considering τ-defined regions. The mathematical equal is given as:

$$d(i) = \begin{cases} \text{index}\left(\min\limits_{t \in [1, D]} E_{stf}(i,t)\right), & i = L \\ \text{index}\left(\min\limits_{t \in [d(i+1)-\tau_d, d(i+1)+\tau_d]} E_{stf}(i,t)\right), & otherwise \end{cases} \tag{8}$$

Those resulting indices are indicators of disparities in the $\text{disparity}_{\text{minimum}}$ – $\text{disparity}_{\text{maximum}}$ range. Then, final contour coordinates are found by subtracting corresponding disparities from horizontal coordinate of input contour points.

The complete algorithm flow is given in the code flow below. Obviously, complexity of the algorithm is O(LD), not taking number of contour points into account. In the first for-loop, Co and Cs cofactors are calculated and added, only once. Then in the second for-loop, state energies are found. Then, trace back is executed to get resulting disparities and second stereo image contour coordinates. Please note that, below pseude-code assumes τ to be equal to 1 only, for easier explanation. But, τ is an adjustable parameter in provided GUI and source code.

Algorithm 1 Contour Correspondence

Input: $\mathcal{C} = \{p_1, p_2, ..., p_L\}$
Output: $\mathcal{C}' = \{p_1', p_2', ..., p_L'\}$

// State transition matrix calculation
for each cell (i, j) in $M_{L\times D}$ **do**
  $M_{i,j} = E(p_i, d_j)$
end for
for $i = 2$ to L **do**
  for $j = 1$ to D **do**
    $M_{i,j} = \min(M_{i-1,j-1} + \lambda_s, M_{i-1,j},$
      $M_{i-1,j+1} + \lambda_s) + M_{i,j}$
  end for
end for
// Minimum energy path traceback
$d_L = \text{index}(\min(M_{L,1}, M_{L,2}, ..., M_{L,D}))$
$p_L' = p_L - d_L$
for $i = L - 1$ to 1 **do**
  $d_i = \text{index}(\min(M_{i,d_{i+1}-1}, M_{i,d_{i+1}}, M_{i,d_{i+1}+1}))$
  $p_i' = p_i - d_i$
end for

Flow 1: Complete StereoSnakes algorithm for contour correspondence by energy minimization

Experimental Results

In this part, experimental results on 5 sample stereo image pairs are provided (see Figures 5-10) with relevant parameter selection. The same images are also stored in the submitted folder, and selectable in the GUI.

Figure 6: Bin image. Notice better predicted contour on the right edge of the lid.

Figure 7: Bag image. Notice self-occluded black strap extension on the right.

Figure 8: Child image. Notice the self-occluded hand is recovered. Image has low resolution; as a result, contour is less smooth.

Figure 9: Shelf image. Notice vertical edges are corrected and fit better, while horizontal edges are not (due to disparity change is constrained in horizontal direction)

Figure 20: Lion statue image. This time, image on the left is actually right stereo image, disparities have negative signs.

User Manual

The project may be run by source code directly, or by a graphical user interface (GUI). Please note that, GUI has extra feature which are explained below.

Both implementations are coded on MATLAB© 2015a version platform. Other releases may produce warnings, error or bug issues.

The average processing time is approximately 30 seconds, depends on image size, computer features and segmentation selection (QuantumCut method takes more time).

For more detail, please see above algorithm explanation, original paper, presentation and comments in the source code. All the coding (except for QuantumCuts) are created by B.Baydar and V.Okbay, relying on the understanding original article [1].

Parameters are common for source code and GUI implementation:

τ : Relative disparity condition. As increased, occlusions may be detected better, but usually results in less smooth shapes. (default: 1)

d-min : Minimum disparity allowed (depends on image size and epipolar geometry)

d-max : Maximum disparity allowed (depends on image size and epipolar geometry)

λ_o : Object boundary cost coefficient. Increases priority of Co function, better self-occlusion detection. (default: 0.6)

λ_s : Dynamic optimization coefficient. When increased, harder to get a discontinuous shape. (default: 30)

w : Window size for neighborhoods in Co and Cs functions (default:13, less effective parameter)

a) Source Code Implementation

Open main.m in MATLAB editor. Following comments, adjust six numerical parameters that are explained above and in the relevant report sections. Make sure “Cs.m”, “Co.m”, “histprobmap.m” functions also exist in the scope of main script.

Store stereo images in the following naming convention: “***name***0.extension” and “***name***1.extension”. Please consider correct sign for disparity interval (dmin and dmax).

Store segmented object mask in the following naming convention: “***name***m.extension”. Ensure mask image is in binary (logical) class and saved with the same extension as stereo images.

For segmentation, main.m script in the QuantumCut folder may be used. Specify image location and run the script directly. Then, threshold resulting quantum cut image to create a binary mask. Those steps are automatically followed in the GUI.

When parameters and images & mask are ready, run “main.m” script of StereoSnakes algorithm. If successful, two images would pop-up, one showing input image with initial contour by mask; while other showing resulting contour on the second stereo image.

b) GUI Implementation

GUI is more user friendly and comes with extra features. A sample flow is explained in steps, below. Make sure "Cs.m", "Co.m", "histprobmap.m" functions and GUI figure file also exist in the scope of main GUI script. See Figure 11.

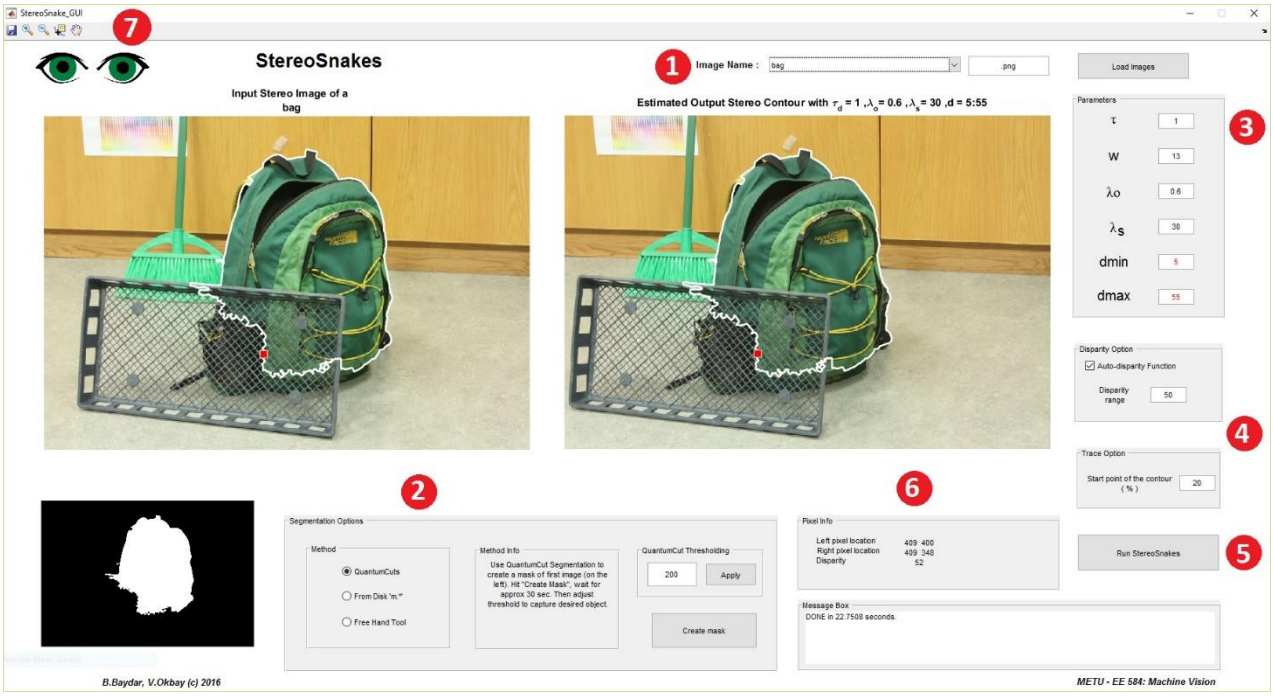

Figure 11: Fully functional GUI view

1-) Select a provided image pair (bin, bag, shelf, lion, child) with extension box ".png" written. Or select "user" option in the drop-down menu, define an image name in the box (to be displayed on the left as input image) in the following convention: "***name***0.*" or "***name***1.*". Make sure both images are stored in the search path of the algorithm (under /Images folder, preferably for QuantumCuts).

2-) Select a segmentation method by following information box next to selection panel.

- For QuantumCut method, specify a threshold value between 0 and 255, hit Create Mask. If desired object is not segmented, change threshold value and apply, without running whole segmentation procedure.
- If appropriate binary image (with the same extension as images) mask is stored in the disk with "***name***m.*" filename, hit Create Mask by selecting "From Disk".
- Free Hand Tool allows user to draw a closed contour on the left image. Please note that, when mouse button is released, contour becomes closed immediately. User may drag it into desired position. Then, double-click on the contour to create a mask.

3-) User parameters explained before.

4-) Two new features coined by us. Auto-disparity feature determines dmin and dmax values by inner calculations. Enter disparity range (dmax-dmin interval). Also user can specify start point of the contour (the red square in the resulting images) that has importance in some cases.

5-) Run algorithm. User may follow message box to get time information after impelementation.

6-) Select a contour point in either of the images to get coordinate and disparity information (zooming in is suggested to select desired points).

7-) Resulting right image can be saved, zoom functions and data tip is available in the toolbar.

Application Area

Stereo images, naturally, has strong correlation with 3-D imaging techniques. Stereo object matching, may be efficient in simultaneous processing on the object located in the left and right images. This processing may include adjusting brightness of both sides in the same amount, resizing stereo objects, or even more complicated tasks such as replacing a stereo object into a stereo scene appropriately. The latter task is called “synthesizing” images, and may lead to depth in 2-D image techniques.

Conclusion and Future Work

Following the complete understanding of [1], a successful implementation has been made. According to our experimental results, a normal segmentation method like QuantumCuts or GrabCut takes almost twice the processing time the method studied in this work. Thus, the studied method can be used for real time tools such as Magnetic Lasso in Adobe Photoshop. While the user is selecting the object contours in one of the stereo image pairs, the object contour in the other image may be automatically determined simultaneously. In future, we are planning to add a function which can determine disparity range (i.e. maximum and minimum disparity values for the contour to be searched for). To do this, we can take the absolute difference of the whole images by shifting them and chose the range at which the absolute difference reaches its minimum (in fact a simple implementation is available in GUI). We can also consider determination of the λ constants automatically, so that stereo correspondence cost and object boundary cost will have the optimal importance in the energy function for the corresponding image. As a final point, although never mentioned in the original article, it is noticed that in several cases, the first contour pixel selection has effect in the result. Trace back procedure, in fact, decreases the significance of the beginning of the contour, but does not erase completely. As a result, we added another facility to shift this beginning point in the GUI.

Submitted Content

- Sample stereo images & masks
- GUI code and figure
- Source code
- QuantumCuts segmentation code
- Original article
- Presentation, report.

References

[1] R. Ju, T. Ren, G. Wu, “StereoSnakes: Contour Based Consistent Object Extraction For Stereo Images”, *ICCV,* 2015

[2] http://www.arts.rpi.ed u/~ruiz/stereo_history/text/historystereog.html

[3] M. Kass, A. Witkin, D. Terzopoulos, “Snakes: Active Contour Models”, *International Journal of Computer Vision*, pp 321-331, 1988