Conversion of 2D to 3D video using interactive co-segmentation

Adarsh P Kowdle	Anandram S	Course Guide: Ashutosh Saxena
Cornell University	Cornell University	Cornell University
apk64@cornell.edu	as2454@cornell.edu	asaxena@cs.cornell.edu

Abstract:

The purpose of this project is to successfully convert a 2D video like a movie to a 3D one by co-segmenting a set of images into multiple classes using user input and then reconstruct a 3D scene using these segments by stereo pair creation. Here the user marks the various segments in an image using scribbles. The program then uses these scribbles and the calculated optical flow to do the segmentation. It then fits planes to each class and then finds the depth map of the image to create a stereo pair output of the 2D video.

Introduction:

There is a growing trend in the movie making industry and that is the creation of entire movies or clips of a movie in 3D and the consumers are also caught in this hype. There is a large database of 2D videos just waiting for the development of an algorithm which can convert them all to 3D. Co-segmentation of clips in a movie is the main task of this project. All the user does is provides scribbles for multiple classes, like say, ground, sky, foreground, background etc,.

Contributions:

The contributions of this project are that

1) It introduces a multi-class segmentation as opposed to the generally prevalent 2-class segmentation.

2) We add features like optical flow to the data term of the model which gives us better results.

3) Introduces sift feature matching and optical flow data to propagate scribbles and segment each image individually on a new model learnt on the image.

4) Gets plane parameters based on the segmentation and some assumptions.

5) Combine the segmentation and plane parameters with existing algorithms to get depth map and create stereo pair output for a simple 3 class (bg, fg, ground) input.

Technique:

1) Get user scribbles using a scribble guidance system.
2) Propagate scribbles across all the images in the sequence.
3) Interactive co-segmentation to get the various classes in the image on all the images in the sequence (This is the part where machine learning comes in).
4) Calculate plane parameters for various classes based on some assumptions and the segmentation.
5) Find the depth map using the plane parameters.
6) Create a stereo pair from the image and its depth map.

Related work:

Interactive image segmentation:

iCoseg: Interactive Co-segmentation with Intelligent Scribble Guidance by Tsuhan Chen,Adarsh, Dhruv, Parikh and Jiebo Luo have developed an interactive co-segmentation algorithm and a user friendly interface which takes a group of related images and segments them by taking in user scribbles for 2 classes.[1]

There are works on active learning where algorithms are able to choose the data they learn from by querying the labeling oracle. This is a vast sub-field of machine learning and we refer the reader to Settles [4] for a detailed survey. In computer vision, active learning has been used for object categorization [7], classifying videos [5], ranking images by information content [6] and creating large datasets [8]. More recently, Kolhi et al. [9] showed how to measure uncertainties from graph cut solutions and suggested that these may be helpful in interactive image segmentation applications.

Approach:

1) The first step is to get the scribbles as input from the user. This is done by just using a mouse to scribble on the displayed image.
2) The next step is to propagate the scribbles from one image to another. This was done by 2 methods:
    a) Use SIFT feature matching to match features between each image in the sequence and the scribbled image. Now consider each triangle formed by the matched feature points and look at all the scribble points in this triangle and generate same number of random scribble points inside the triangle formed by the matched points in the other image.
    b) Use the optical flow data to see where the scribble has moved in the next image.
3) The next step is the co-segmentaion of the images in the sequence. This is done by energy minimization

Energy minimization:

We define the labeling problem as the minimization of the overall energy (Gibbs energy) of a graph constructed over each image. In the preprocessing stage each image is first rescaled and a set of super-pixels are formed over these images. We construct and do the segmentation on these super-pixels. The graph is also constructed on these super-pixels. The vertices of the graphs are the super-pixels and the edges connect the adjacent super-pixels. Using the scribbles provided by the user, we label these super-pixels and using these labels we learn a model consisting of a mixture of Gaussians to define the unary (data) appearance model, say A1, and another model to define the pair-wise(smoothness) appearance model, say A2. This A={A1,A2} is common to all the images in a group and using these models we define the energy of a node(k) as

$$E^k(X^k{:}\,A) = \sum_{i \in V^k} E_i\left(X_i^k{:}\,A1\right) + \alpha \sum_{(i,j)\in \varepsilon^k} E_{ij}(X_i^k, X_j^k{:}\,A2)$$

The first term corresponds to the cost of saying the pixel belongs to a particular class and the second term is used for penalizing any label discontinuity between adjacent pixels.

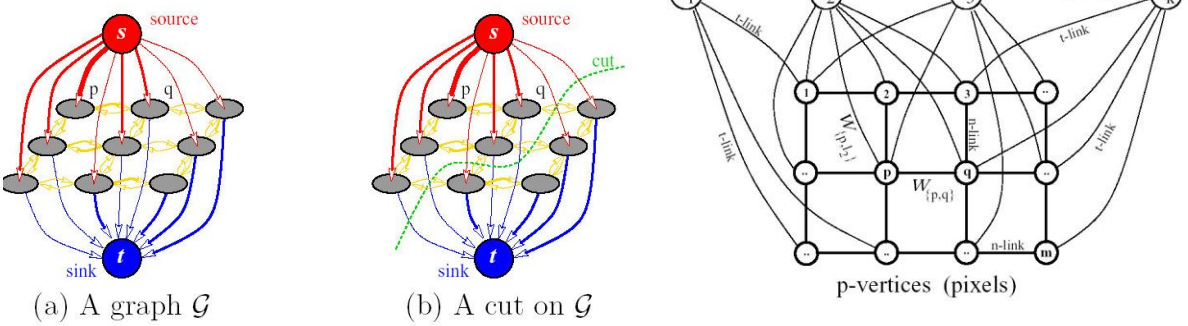
Data (Unary) Term:

Our unary appearance model is a multi-class Gaussian Mixture Model, i.e., A1 = (GMM1 ;GMM2;GMM3….). We extract color features extracted from super-pixels. We use features from labeled sites in all images to fit the various classes in the GMM. We then use these learnt GMMs to compute the data terms for all sites, which is the negative log-likelihood of the features given the class model. To this we add an optical flow term calculated from the flow mats calculated in the pre-processing using a the optical flow developed by C.Liu[3]

Smoothness (pairwise term):

This is a simple indicator function I(Xi≠Xj)exp(-β*dij)
The dij term is also learnt every time a scribble is provided.

The labeling is then done by minimizing the energy for each pixel in the image by assigning it to a particular class. Using the updated values this segmentation is done on each image by doing a graph-cut algorithm. The fact that a particular pixel label depends on the labels of its neighbors allows modeling the optimization problem as a Markov Random Field. The graph contains 2 types of vertices, the p-vertices which are all the pixels in the MRF and the l-vertices which are the labels. The figure [11] on the left below shows how to do a cut on the graph for a binary labeling case and the one on the right shows the graph for a multi-label case.



(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

4) The next step is to calculate the plane parameters a, b, c and d for each of the classes in the image.

   ax+by+cz=d

   This is the equation of a plane. So for each class we fit a plane and specify the plane parameters. This is done by assuming the position of the camera to be at the center and at 1 unit above the ground plane (in a transformed space). We then look at where the segmentation (using scribbles) of various classes meet the ground plane and assuming they are all perpendicular to the ground we find the plane parameters for the other classes. We also set the plane for a class at the first pixel in the y-direction.
5) Once the segmentation and the plane parameters have been successfully calculated the depth-map is obtained using the code written by Adarsh[10].
6) We then feed the depth map and the image to another program which creates stereo pair by calculating the inverse of the depth to give the distance a pixel has moved in the second image in the stereo pair. So in this case the pixels which are farther away move less compared to the pixels which are closer.

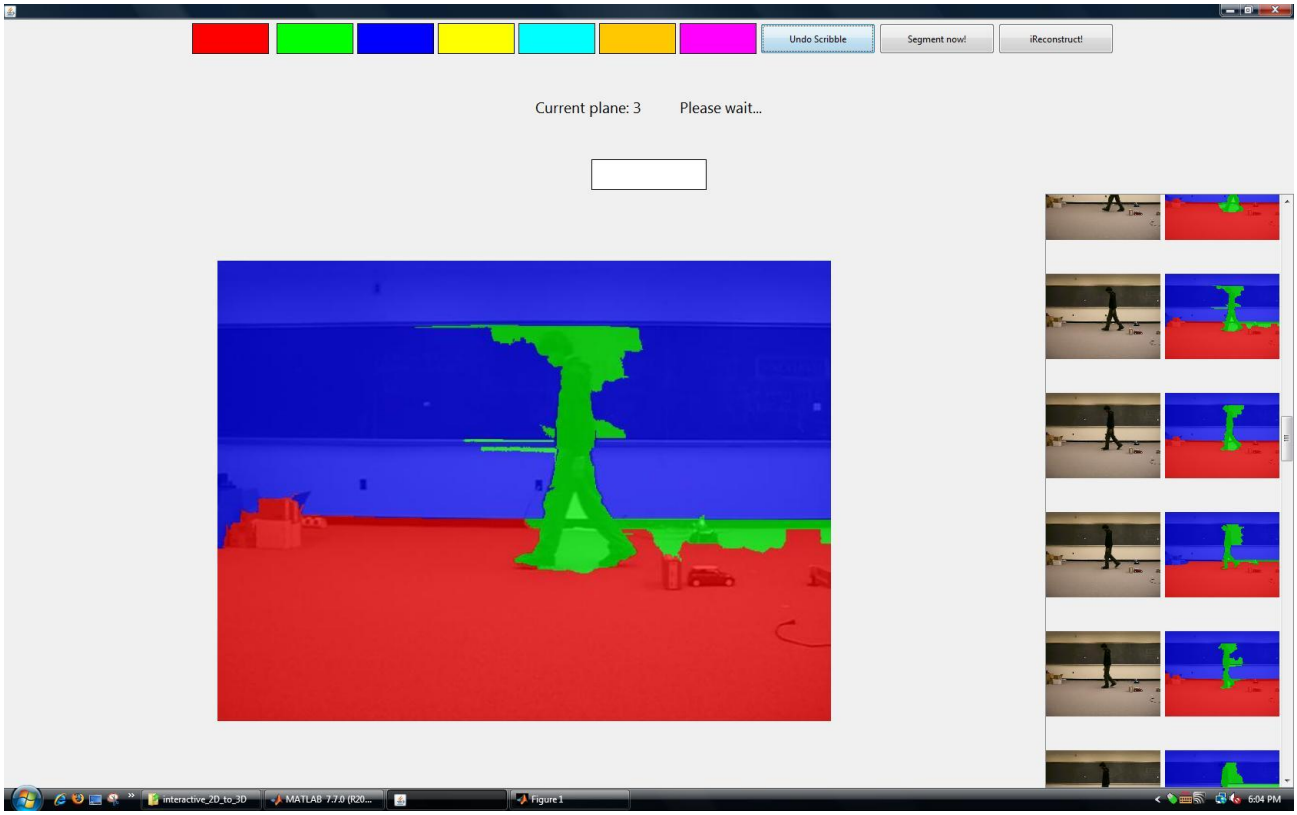
Dataset:

To evaluate the developed algorithm we captured movies around the campus of buildings, people, and arbitrary objects. The movies were then sampled at 10 fps to get a set of images. The movies were all with the camera static and with the foreground object moving. All the movies had the ground, foreground and background clearly distinct.

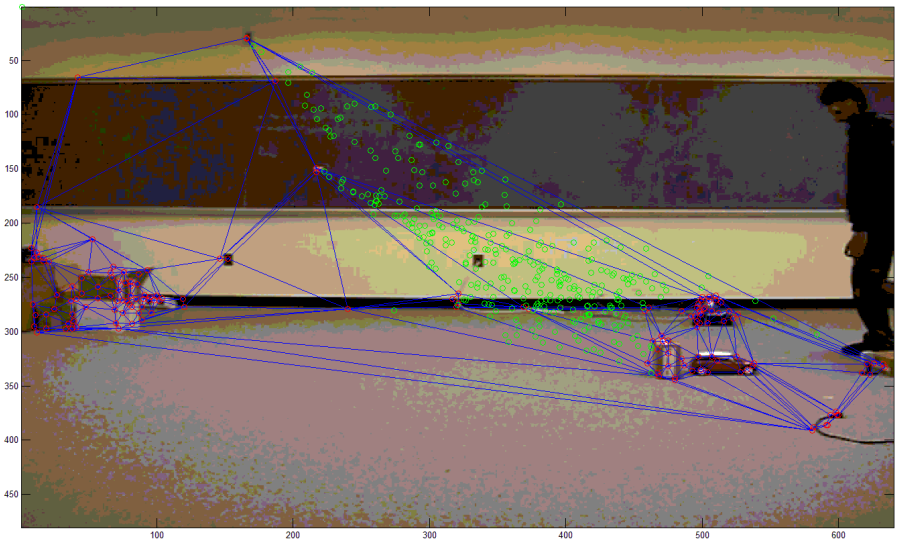
Experiments and Results:

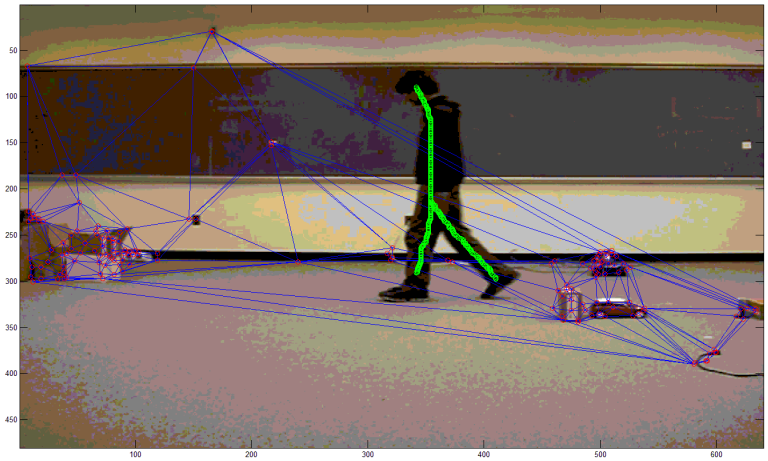
We have run the multiclass on the data sets and so far the multiclass segmentation has given satisfactory results and after adding the optical flow there is an excellent segmentation happening. We have also successfully propagated the scribbles from one image to another using both sift feature matching and optical flow data. The former was very ineffective in all the test cases mainly because of the lack of enough features in the image. The latter was giving a problem when there was any occlusion happening. Thus the proposed idea is to combine both of these to give optimum results. This is done by looking at only points on the scribbles which are matched and then using optical flow data on only those points. This removes any errors due to occlusion as the matching will not happen in such cases. The presence of blur due to motion made both the methods slightly less effective. The calculation of plane parameters were done based on the above mentioned assumptions and using the pre-existing code for depth map calculation and stereo pair production, we were able to create a 3D version of the movie captured using a static camera. Though the foreground is clearly seen in 3D, some of the other features in the background also pop up. This is because of improper segmentation and due to slight errors (human) in the camera position assumption.

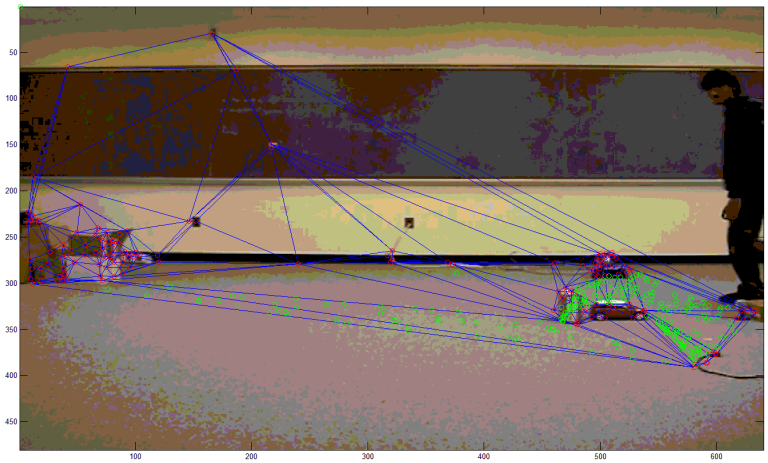
The multiclass segmentation on an image set using optical flow

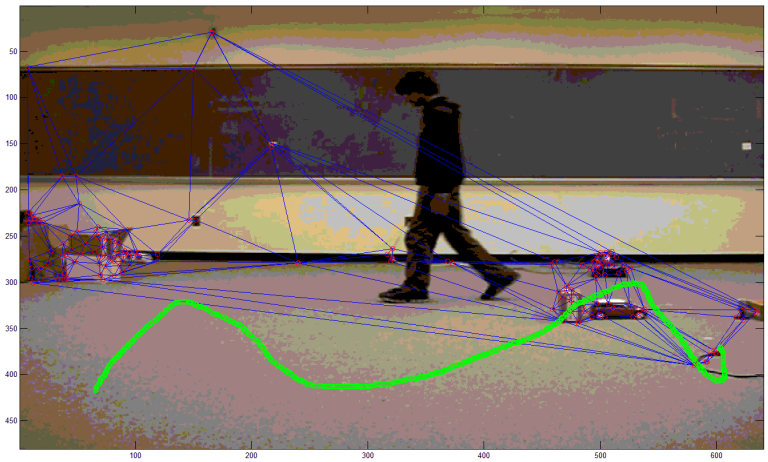


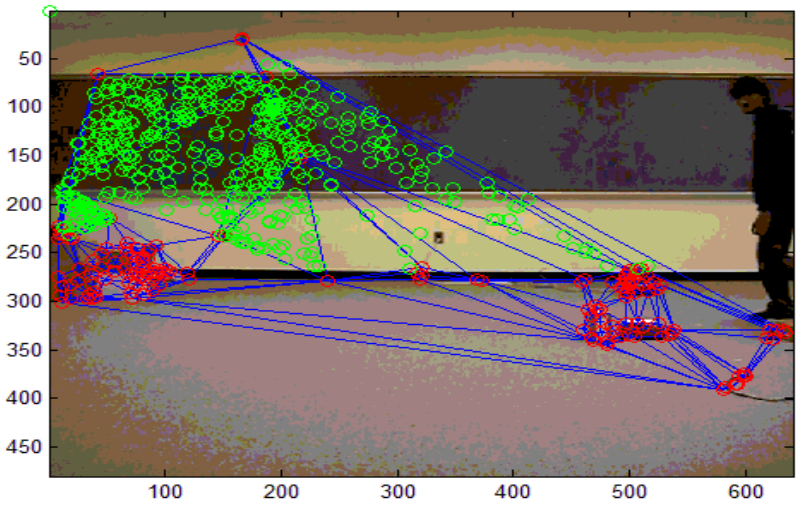
Scribble propagation using SIFT feature matching

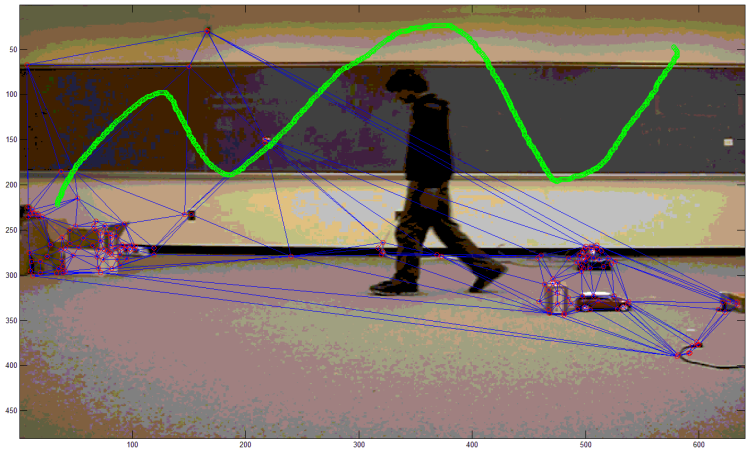



50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

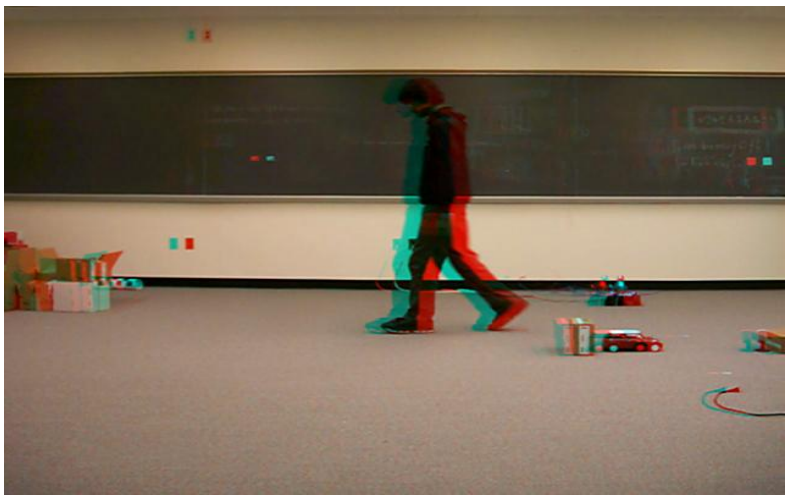

50
100
150
200
250
300
350
400
450
100
200
300
400
500
600


50
100
150
200
250
300
350
400
450
100
200
300
400
500
600





Output



Evaluation:

We ran the entire algorithm on a movie sequence which contained 59 images and saw how well the output 3D version is. We checked the intermediate portions of the algorithm to see how well the segmentation is happening. The effectiveness of the algorithm is also checked by looking at how many scribbles are required for the said segmentation. This is done by comparing each pixel to a ground truth image marked on one of the images in the sequence.

Future Work:

1) We can try a different interaction scheme like probably providing a scribble over the entire cuboid of frames rather than one image. This gives scribble points for all classes on all images in the sequence in one go, thus making it a much more user friendly and effective algorithm.
2) We can modify the plane parameter estimation to a piecewise planar or some other 3D fit model for the classes so as to give a better and realistic feel to the scene.
3) Try the algorithm for multi-class input as this will be the case in most real life situations. There might be more than one object in the foreground or the background could be a cylinder.
4) We can also try and modify it to give results on other datasets like for a static scene, moving camera and for a moving object, moving camera.
5) Implement the combination of optical flow and SIFT feature matching to get better results even in the case of occlusions. Also try for a better feature matching algorithm to extract more features from the image.

References:

1) D. Batra, A. Kowdle, D. Parikh, J. Luo, and Chen. T. icoseg: Interactive co-segmentation with intelligent scribble guidance. In CVPR,2010

2) Make3D: Learning 3-D Scene Structure from a Single Still Image, Ashutosh Saxena, Min Sun, Andrew Y. Ng, To appear in*IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2008.

3) C. Liu. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis**.** Doctoral Thesis. Massachusetts Institute of Technology. May 2009.

4) B. Settles. Active learning literature survey. Computer Sciences
Technical Report 1648, University of Wisconsin–Madison, 2009.

5) R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In ICCV, 2003.

6) S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In CVPR, 2009.

7) A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In ICCV, 2007

8) B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In ECCV, 2008.

9) P. Kohli and P. H. S. Torr. Measuring uncertainty in graph cut solutions. CVIU, 112(1):30–38, 2008.

10) Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman, Cambridge University Press, June 2000.

11) Graph Cut Algorithms in Vision, Graphics and Machine Learning - An Integrative Paper by Sudipta N. Sinha f ssinha@cs.unc.edug, University of North Carolina at Chapel Hill.