Combining Monocular and Stereo Depth Cues

Fraser Cameron

December 16, 2005

Abstract

A lot of work has been done extracting depth from image sequences, and relatively less has been done using only single images. Very little has been done merging these together. This paper describes the fusing of depth estimation from two images, with monocular cues. The paper will provide an overview of the stereo algorithm, and the details of fusing the stereo range data with monocular image features.

1 Introduction

Recent work has been done on depth estimation from single images[1]. While this is exciting, it could benefit from the existing stereo image depth algorithms. In a real control system there would afterall be streams of incoming images. This paper will discuss first the stereo algorithm, then how to fuse stereo depths with the monocular features, and finally the results obtained so far.

There has been a lot of work done calculating depth from two stereo images. Here we assume only knowledge of the camera calibration matrix, and no relative rotation. The rest is estimated from correspondencences. We use Feature-based depth estimation techniques due to the availability of MATLAB [2] [3] code, and a perceived speed bonus. This is more fully explained in section 2.

Recent work by Sexena et al. has focused on depth estimation from monocular image cues. They have supervised learning to train a Markov Random Field. This work has also been applied to obstacles depth estimation for use controlling a remote control car at speed [4]. This second application specifically requires fast computation. The intention of this work is to begin combining monocular and stereo image cues into a fast computation, allowing improved depth estimation for dynamic control applications. We lay the groundwork for this in section 3.

Due to time constraints we were only able to generate and test the stereo system. These details are provided in section 4.

Finally the paper close with some conclusions, and notes on work in progress in section 5, and acknowledgements in sections 6.

2 Stereo Depth Estimation

Depth Estimation in this paper using image sequences is broken up into 3 sections: Feature Detection and Correlation, Estimating the Fundamental Matrix, Additional Guided Matching, Depth Estimation, and Error Estimation.

2.1 Feature Detection and Correlation

For correlation we desire a relatively small and even smattering of feature points. This is achieved by corners. In this paper we use Harris corner detection, which uses a threshold on the top two singular values of small image windows. The Harris algorithm uses a spatial edge suppression technique to prevent detecting multiple edges where only one exists.

[1] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng, "Learning Depth from Single Monocular Images"

[2] P. D. Kovesi, "MATLAB and Octave Functions for Computer Vision and Image Processing" http://www.csse.uwa.edu.au/~pk/research/matlabfns/

[3] A Zisserman, "MATLAB Functions for Multiple View Geometry" http://www.robots.ox.ac.uk/ vgg/hzbook/code/

[4] Jeff Michels, Ashutosh Saxena, Andrew Y. Ng, "High Speed Obstacle Avoidance using Monocular Vision and Reinforcement Learning"

Kovesi provides a correlation scripts for matching through monogenic phase or direct intensity correlation. Each searches for high window correlations in a range around each feature. Only matches where each feature selects the other are kept. Kovesi notes a typically better performace for matching through monogenic phase, and a potential comparability of speed. I have used monogenic phase matching here and regular matching later, to get the benefits of both.

2.2 Fundamental Matrix

From the correlated matches we can generate the fundamental Matrix, F, for the two images. F encodes the location in pixels of the projection of the two cameras on the opposite image plane as well as the rotation between the two images. The equation defining F is:

$$\bar{p}_r^T * F * \bar{p}_l = 0 \tag{1}$$

Where $\bar{p}_r$, and $\bar{p}_l$ refer to the pixel coordinates of the matches in the right and left image respectively.

Here we use the Random Sample Consunsus algorithm (RANSAC) which iteratively computes an F based on a random sample of matches and evaluates how many matches agree. The eight point algorithm is used to construct F. To evaluate whether or not a match fits the current estimate RANSAC simply evaluates (1) and applies a pre-determined threshold. Ransac stores the details of the best candidate found. The algorithm terminates when it is 99% sure that it has chosen a random set of matches containing no outliers. The probability of an outlier being picked is set from the fraction of matches were deemed inliers for the best candidate.

Since RANSAC classifies correlations into inliers and outliers, we simply discard the outlying matches.

The RANSAC distance function ignores the direction of the match, and large angular deviations from the epipolar line for matches with very similar pixel coordinates. While these checks could be included in the distance function, they are used afterwards, to avoid high computation costs in the iterative RANSAC algorithm.

This implementation assumes no relative rotation, since most control algorithms using image streams will have small relative angular rotations, and will be concerned primarily with nearby objects. This leads to a faster and more consistent RANSAC convergence.

2.3 Additional Guided Matches

Now that we have determined F we have a lot more information about where matching features should be. Indeed we can reduce it to a one dimensional line search. Further, since we have only translation we can determine whether a matching feature should be closer or farther from the epipole. We use this information to perform another dual correlation search. We perform several rounds of matching, removing succesfully matched points at the end of each round, to obtain as many correlations as possible.

2.4 Depth Estimation

With the camera calibration matrix, M, and F found above, one can obtain the Essential Matrix, E, which encodes information on the relative rotation, and position of the two cameras. Not knowing how far or what direction we have moved, results in a scale factor ambiguity in E. We obtain the normalized E using:

$$E = \frac{MFM}{\sqrt{tr((MFM)^T(MFM))/2}} \tag{2}$$

Given that we have assumed pure translational motion, E takes the form:

$$E = \begin{bmatrix} 0 & -\hat{T}_z & \hat{T}_y \\ \hat{T}_z & 0 & -\hat{T}_x \\ -\hat{T}_y & \hat{T}_x & 0 \end{bmatrix} \tag{3}$$

Where $\hat{T}$ is the normalization of the unknown translation vector. It is worth noting that $\begin{bmatrix} \frac{\hat{T}_1}{\hat{T}_3} & \frac{\hat{T}_2}{\hat{T}_3} & 1 \end{bmatrix}$ is the projection of the epipole on the image plane. Using M and $\hat{T}$ one can estimate the depth of points by triangulating their position using:

$$Z_l = \frac{\hat{T}_1 - x_r\hat{T}_3}{x_l - x_r} \tag{4}$$

Here x_r and x_l refer to the 1st coordinates of the projected matches in the right and left camera frame respectively. This equation follows the convention that third component of the projected matches is 1. They are obtained from $p_r = M\bar{p}_r$. Since we have assumed $R = I_3$ the only ambiguity comes from the sign of $\hat{T}$, which just causes a sign switch in Z_l. In cases where $x_r - x_l$ is too small we substitute y_r and y_l instead.

2.5 Error Estimation

To estimate the error of this stereo algorithm, a series of photographs and laser depth scans were taken around campus with 2 foot separations. Thus the comparison is between the depths calculated from the pictures and the laser range depths.

There are a number of sources of error in this matching:

1. The laser depths are taken slightly misaligned from the camera. This misalignment can very easily cause massive errors due to the features being positioned often on the edge of large stepchanges in depth. An effort has been made to select pictures with many features in consistent depth regions.

2. The relatively fewer corners offered by structured rather than unstructured scenes increases the correlation accuracy, since there are simply fewer wrong choices to confuse the matching algorithm. In extreme cases, there were not enough matches to estimate F, leading to no depth data at all.

3. Since the pictures were taken at different times near sundown, about 5 minutes, some scenes have different lighting conditions, we have removed the scenes with very significant lighting changes, but some lighting differences remain. Also, some objects (people and bikes) have moved between pictures.

4. The laser depths and stereo depths have different ranges. This leads to large errors when stereo depths are well beyond the maximum laser depth.

5. Metal or windows can create spurious features from reflected light. They can also create laser depth readings by not reflectingthe laser back.

3 Fusing with Monocular Cues

Saxena, et al's "Learning Depth from Single Monocular Images" uses a Markov Random Field to model the depth relations among image patches. They then maximize the distance probability over several parameters given the dataset. While, they use both Laplacian and Gaussian distributions, we will only concern ourselves with Gaussian distributions here. Further they use multiple scale leading to several layers of distance paramters. Only base distances are left independent, as the lower resolution copies are restricted to be averages. We extend their gaussian model by adding a gaussian penalty for distance from the stereo depths. The model is:

$$P(d|X;\theta;\sigma) = \frac{1}{Z} exp\left[-\sum_{i\in SFM}\frac{(d_i(1) - d_m - d_{SFMi})^2}{2\sigma_{SFM}^2} - \sum_{i=1}^{M}\frac{(d(1) - x_i^T\theta_r)^2}{2\sigma_{1r}^2}\right] exp\left[-\sum_{s=1}^{3}\sum_{i=1}^{M}\sum_{j\in N_2(i)}\frac{(d_i(s) - d_j(s))^2}{2\sigma_{2rs}^2}\right] \tag{5}$$

$d_i(s)$ refers to distance i at scale s; SFM is the set of point provided by the stereo algorithm; d_{SFMi} is the distance provided by the stereo algorithm for point i;$d_m = \sum_{i\in SFM}\frac{d_i}{|SFM|}$;$\sigma_{SFM}$ is an empirically determined quantity expressing the error in the measurements. All other parameters are as described in Saxena, et al's work. We choose to restrict

If we put this into the standard Gaussian form, $exp(-(d - \tilde{\mu})^T\widetilde{\Sigma}^{-1}(d - \tilde{\mu}))$, the maximum liklihood estimate for d would be μ. Expanding by order of d we see:

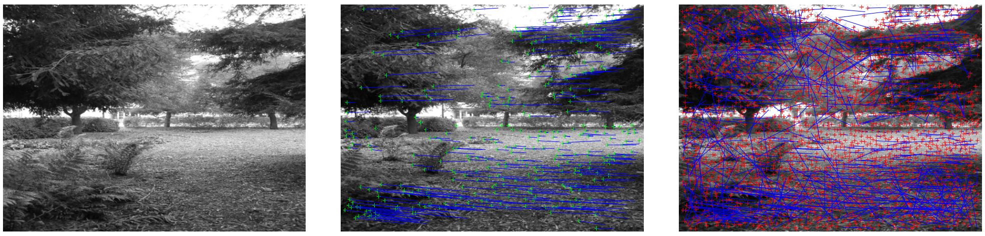
$$\widetilde{\Sigma}^{-1} = \Sigma_{SFM}^{-1} + \Sigma_{1r}^{-1} + \sum_{j\in N_2(i)}\sum_{s}\Sigma_{2rs}^{-1}\forall i \in SFM \tag{6}$$

$$\tag{7}$$

Where $\Sigma_{SFM}, \Sigma_{1r}, and\Sigma_{2rs}$ are known empirically. μ can then be found by iteratively fitting it to minimize the gap between the two term of order 1 in d. This, combined with the current monocular techniques gives us a gaussian models for each d_i, and thus the maximum liklihood estimate.

4 Results

Figure 1 shows an image in various stages of processing. For unstructured environments there are an enormous number of corners leading to many correlations. These are then whittled down by estimating F, and converted into depths, shown below. Stereo depth algorithms will never show the sky as it has no corners.

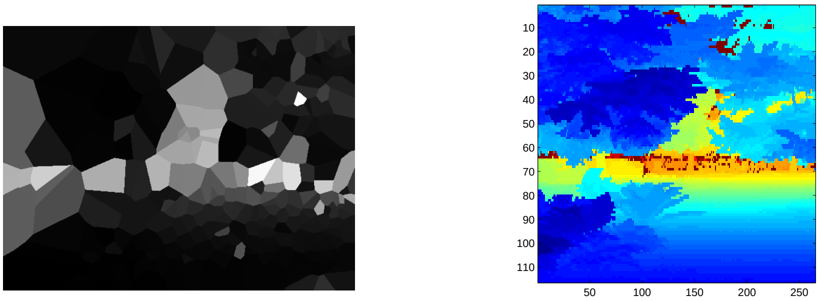


(a)

(b)

(c)

Figure 1: a) Base left image. b) Potential matches from correlation. c)Right camera image with inlying matches, crosses indicate corners on right image.

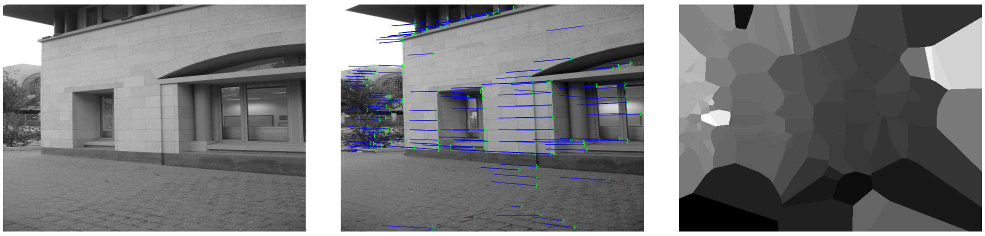


(a)

(b)

Figure 2: a) Estimated depth from features. b) Laser range scan data

Figure 2 shows the estimated depth on the left and the true depth on the right.



(a)

(b)

(c)

Figure 3: a) Base left image. b) Potential matches from correlation. c)Right camera image with inlying matches, crosses indicate corners on right image.

Figure 3 shows a structured scene. Here we see the effect of glass, causing the algorithm to give the depth of a reflection in the top right. Further, we see the error inherent in extremely regular textures.

For a selected set of structured images, for which a large proportion of the features exist in surfaces as opposed to on the edge, this stereo algorithm has a true log depth to calculated log depth correlation of 0.8606, and a standard deviation of 0.4472. For reference, a constant guess of the mean produces a standard deviation of 0.8773. Due to the scale ambiguity in our range estimates, they were shifted to have the same mean.

5 Conclusion

Image manipulation, and 3-D reconstruction are not trivial. Indeed, although some code existed, I needed to code up a substantial amount of the process. Particularily, there were numerous spurious matches that needed to be pruned.

The stereo algorithm that resulted works, but would benefit from some fine tuning, and speed enhancements. It is more accurate, but less informative for structured scenes.

Comparing the calculated range data to laser range scans requires significant processing to ensure that the limited ranges, and misalignments do not destroy the results. I could easily get near zeros correlations through the right choice of scene.

Clearly more work needs to be done. The to do list includes using the camera calibration to undistort the image, adaptive distance limits for the guided matches, automatically selecting features insensitive to laser scanner/camera misalignment for error measures, algorithm speed improvements, and of course complete integration with monocular features.

6 Aknowledgments

I would like to thank Prof. Ng for the loan of his book ”Introductory Techniques for 3-D computer Vision” by Trucco & Verri. Also, Ashutosh Saxena was quite generous with his time. Further, I used a large number of online resources, mainly the sample chapter from Andrew Zisserman’s ”Multiple View Geometry in Computer Vision” entitled “Epipolar Geometry and the Fundamental Matrix”.