RECOVERY OF A MISSING COLOR COMPONENT IN STEREO IMAGES

Serdar Ince

May 2004

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2004-02

BOSTON
UNIVERSITY

RECOVERY OF A MISSING COLOR COMPONENT IN STEREO IMAGES
(or helping NASA find little green Martians)

Serdar Ince

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
`www.bu.edu/ece`

May 2004

Technical Report No. ECE-2004-02

Summary

Recent project of National Aeronautics and Space Administration (NASA) to Mars promised to show 3D images of Mars surface. Spirit and Opportunity, identical Mars rovers of NASA, are equipped with a high-resolution stereo camera pair, called PanCam. Most cameras used by people are sensitive to three color bands, namely red (R), green (G) and blue (B). The PanCam is equipped with more color bands since it was designed to deliver more information to geologists. NASA scientists use special filters to create an RGB image from PanCam's outputs. However, while these filters can reconstruct RGB components of left image properly, they can reconstruct only R and B components of the right image. Therefore, since G component of the right image is missing, currently it is not possible to view a color 3D image of Mars. Considering this problem, the aim of this research is to reconstruct the third color component of an image given its two color components and the other image from stereo.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thanks to Hollywood and science fiction authors, the last few generations have been raised with the thoughts of life on planet Mars. Recent mission of identical NASA rovers Spirit and Opportunity to Mars gave a lot of information about this planet. Images taken by Spirit and Opportunity are published in newspapers almost daily. In order to be able to understand the surface of Mars, Spirit is equipped with a cutting-edge stereo camera pair called PanCam. PanCam is specially designed to take stereo pictures to help the scientists in their research. Ordinary cameras that we use daily are designed to capture three color bands, namely red (R), green (G) and blue (B). PanCam is sensitive to more color bands, since this information is valuable for geologists. Converting these additional color bands to RGB is a problem that has to be addressed. Using the spectral range of colors, NASA scientists use filters to reconstruct RGB images. They can easily construct RGB components of left image, however this is not the case for right camera; It turns out that G component of right image cannot be created using these filters. Since G component of the right image is missing, it is not possible to see color Mars images in 3D. We would like to address this problem in this project.

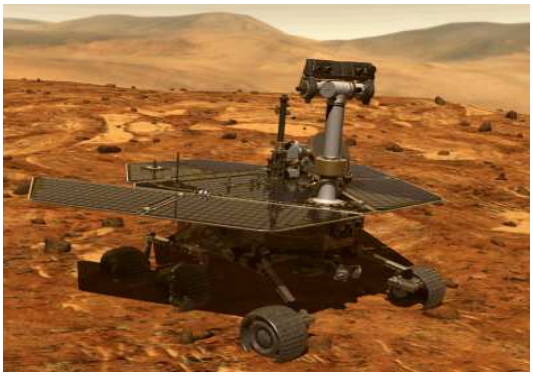


Figure 1.1: Mars rover of NASA

1.1 Problem statement

The goal of this project is to create the third color component of an image given its two color components *and* one image of a stereo pair. Without loss of generality, we will focus on the G component of the images as the 'lost' component. However, the work can be extended to other color components as well.

1.2 Outline of the report

Since we are going to deal with color, it is appropriate to start with basic information on color and human visual system. The next chapter, following color vision, will elaborate my solution method to solve the problem. Specifically, block-based and optical flow based disparity estimators with multiple color components will be discussed. Objective results using stereo pairs will be presented. Tests using real Mars images will follow. A section that discusses the problems encountered during the work will be followed by suggestions for future work and report will end with concluding marks.

Chapter 2

Color vision

This section briefly describes human color vision system, more details about this topic can be found in [5].

2.1 How does the human eye work?

The human eye can perceive a limited range of light spectrum. It can detect from about 400 nanometers which is violet to about 700 nanometers which is red. Wavelengths of colors are given in Table 2.1.

The light passes through the lens of the eye and falls on the *retina*. Two types of receptors are located on the retina: *cones* and *rods*. Cones work under bright light conditions and are able to perceive color, while rods work under low light condition and can only extract luminance information [9]. There are three types of cones, each of which has a different passband in the visible spectrum. The sensitivity curves of cones are given Fig. 2.1. It is easy to notice that curves reach their maximum values near blue, green and red color wavelengths, respectively. Prior to this finding, Thomas Young (1802) stated that any color can be reproduced by mixing an appropriate set of three primary colors [5]. Based on this three-color theory, spectral energy of a colored light $C(\lambda)$ can be described by spectral responses as

$$\alpha_i\,(C) = \int_{\lambda_{min}}^{\lambda_{max}} S_i(\lambda)\, C(\lambda)\, d\lambda \qquad i = 1, 2, 3 \tag{2.1}$$

2.2 Color reproduction

The basic problem in color vision is to reproduce a specific color using a set of light sources. This is usually achieved by using three sources due to the three-receptor model of eye, and they are called *primary sources*. The primary sources standardized by *Commission Internationale de l'Eclairage* (CIE) are three monochromatic sources $\lambda_1 = 700$ nm (red), $\lambda_2 = 546.1$ nm (green), $\lambda_3 = 435.8$ nm (blue).

Color	Wavelength
Ultraviolet	≤380 nm
Violet	380-450
Blue	450-500
Blue-green	500-520
Green	520-550
Yellow-green	550-570
Yellow	570-600
Orange	600-630
Red	630-680
Infra-red	≥680

Table 2.1: Wavelengths of colors

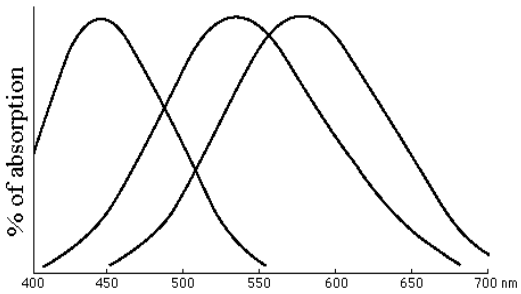


Figure 2.1: Sensitivity curve of cones in human retina

Using this idea, digital color images can be represented using three color components, R-G-B. Fig. 2.2 shows three color components of the famous Lena image as gray scale images. Color distribution of the image can be easily seen from these images. Lena has a dominant color of red, therefore R component has bright values.

(a) R component

(a) G component

(a) B component

Figure 2.2: Color components of the famous Lena image. Each component is presented in gray scale.

Chapter 3

Literature review

The problem that is considered here is a very specific concept, therefore it is not easy to find related literature. However, the problem is closely related to multi-constraint estimation of motion (which is disparity in our case).

Ordinary motion estimation (ME) algorithms work on a single component of the image. This is usually the luminance (Y) component. Information in chrominance components are usually ignored because entropy of these components is usually small. Therefore, ME algorithms estimate the motion on Y component and apply the field to other components as well. The cost function of single-constraint methods is

$$E = \Psi \left[I_{1,Y}(\boldsymbol{x}) - I_{2,Y}(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})) \right], \tag{3.1}$$

where Ψ is an error function, for example mean square error, and $\boldsymbol{d}$ is the displacement.

This approach can be extended to multi-constraint estimation where other image components are also considered in the cost function as follows

$$\sum_{i=1}^{3} \Psi \left[{}_{1,i}(\boldsymbol{x}) - I_{2,i}(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})) \right]. \tag{3.2}$$

The idea is that if there exists a local feature, a corner for example, in the area of interest, this feature would probably exist in other components too. In this way, estimation process becomes robust to intensity mismatches in a single color component.

Our algorithm can be basically summarized as estimation of disparity between left and right views using two color components and then warping the third color component of *left* image using the calculated field to create the *right* image. Since information in the right image does not exist, pixel values of left image will be used during reconstruction therefore occlusion areas will be reconstructed incorrectly. Details of the algorithm will be presented in the next section. Brief summaries of relevant papers will be given here.

Golland and Bruckstein discuss estimation of motion from color images in [3]. They compare three different methods, each utilizing a different color representation, for estimation of optical flow. The first method computes the optical flow using RGB

components as separate gray scale images. Therefore there are three optical flow constraint equations:

$$\frac{\partial R}{\partial x}u + \frac{\partial R}{\partial y}v + \frac{\partial R}{\partial t} = 0, \tag{3.3.a}$$

$$\frac{\partial G}{\partial x}u + \frac{\partial G}{\partial y}v + \frac{\partial G}{\partial t} = 0, \tag{3.3.b}$$

$$\frac{\partial B}{\partial x}u + \frac{\partial B}{\partial y}v + \frac{\partial B}{\partial t} = 0. \tag{3.3.c}$$

This method assumes the brightness is conserved between images. Solution of this overdetermined system of linear equations will yield the result.

Second method uses *chromaticities*[1]. In the chromaticities are defined as follows:

$$r = \frac{R}{R+G+B} \tag{3.4.a}$$

$$g = \frac{G}{R+G+B} \tag{3.4.b}$$

$$b = \frac{B}{R+G+B} \tag{3.4.c}$$

This method assumes color conservation as opposed to brightness conservation. It is obvious that there are two independent chromaticities. Therefore using two components, for example r and g, the following flow constraint equations can be defined.

$$\frac{\partial r}{\partial x}u + \frac{\partial r}{\partial y}v + \frac{\partial r}{\partial t} = 0, \tag{3.5.a}$$

$$\frac{\partial g}{\partial x}u + \frac{\partial g}{\partial y}v + \frac{\partial g}{\partial t} = 0, \tag{3.5.b}$$

Since there are two unknowns, u and v, and two linearly independent linear equations, the system is well-posed and solution of the system will yield optical flow.

Third approach uses *Hue-Saturation-Value (HSV)* representation. HSV values are defined as follows

$$V = Max(R,G,B), \tag{3.6.a}$$

$$S = \frac{Max(R,G,B) - Min(R,G,B)}{Max(R,G,B)} \tag{3.6.b}$$

[1]Golland and Bruckstein use the term 'normalized rgb' for this representation

$$
\begin{aligned}
H = \quad & \text{if} \quad R = Max(R,G,B) \quad \text{then} \quad \tfrac{G-B}{Max(R,G,B)-Min(R,G,B)} \\
& \text{if} \quad G = Max(R,G,B) \quad \text{then} \quad 2 + \tfrac{B-R}{Max(R,G,B)-Min(R,G,B)} \\
& \text{if} \quad B = Max(R,G,B) \quad \text{then} \quad 4 + \tfrac{R-G}{Max(R,G,B)-Min(R,G,B)}
\end{aligned}
$$

Similar to chromaticities, there are two independent quantities and two linear independent equations to be solved. The equations are

$$\frac{\partial H}{\partial x}u + \frac{\partial H}{\partial y}v + \frac{\partial H}{\partial t} = 0, \tag{3.7.a}$$

$$\frac{\partial S}{\partial x}u + \frac{\partial S}{\partial y}v + \frac{\partial S}{\partial t} = 0. \tag{3.7.b}$$

Experimental results using synthetic and real images show that estimation based on two color representation outperforms the brightness representation. Performance of chromaticity representation and HSV is compatible and depends on the data.

In an earlier work, Konrad also discusses the similar problem in [6]. He generalizes the work of Horn and Schunck [4] to multiple image components. He utilizes both chrominance and luminance components in the cost function. The departure from smoothness is defined by a vector multiplication since there are three components. The difference in his solution between Horn and Schunck work is that he uses Taylor expansion around previous iteration, i.e $d = d_{prev}$, of motion vectors while Horn and Schunck used the expansion around $d = 0$. This leads to a more accurate estimate of derivatives due to the displacement compensation. He solves the problem using deterministic relaxation. The experimental results in the paper indicate that multiple constraint based algorithm yields better results than luminance-based algorithm. However, as one expects, computational complexity increases.

Another work that is relevant to this project is by Alvarez and Sanchez [2]. Actually this work is a generalization of [1], where they used a PDE-based method for disparity estimation, to color image model. They modify the cost function such that it includes all three color components. They use the RGB components as separate gray scale images.

Final work on color images that I would like to mention is by Mühlmann *et al.* [8]. Although they don't provide any theoretical work on estimation, they demonstrate an efficient implementation of a disparity estimator for color images. They use RGB components as separate gray scale images.

If we make summary of papers we presented, the usual approach is to use each color components as a separate gray scale image and estimation of disparity jointly using all color components.

Chapter 4

Solution method

The main step in the reconstruction algorithm is disparity estimation. There are many ways to estimate disparity. The easiest way is to use block based estimation technique.

4.1 Block matching with multiple constraints

The easiest way to estimate disparity in stereo images is block matching. The usual approach is converting the RGB image to YUV space and then working on Y component only. This approach can be extended to multiple color components. For three color components, the cost function becomes

$$E = \sum_{i \in R,G,B} \sum_{m \in M} \sum_{\boldsymbol{x} \in B_m} \lambda_i \Psi \left[I_{1,i}(\boldsymbol{x}) - I_{2,i}(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})) \right], \tag{4.1}$$

where M is the domain of the image, m is a block, B_m is the domain of the block, $\boldsymbol{x}$ is the pixel position, $\boldsymbol{d}(\boldsymbol{x})$ is the disparity value at corresponding pixel position, λ_i's are weights of each component and Ψ is some error function.

Minimizing this cost function will yield the minimum energy solution, however this is not necessarily the best solution. The prior information about disparity is that disparity field should be smooth. Neighboring blocks will have very similar disparities since the objects are rigid in real world. Using this prior information we can update the cost function by adding a regularization factor so that resulting disparity field is smooth within the image.

$$E = \sum_{i \in R,G,B} \sum_{m \in M} \sum_{\boldsymbol{x} \in B_m} \lambda_i \Psi \left[I_{1,i}(\boldsymbol{x}) - I_{2,i}(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})) \right] + \lambda_{reg} \sum_{y \in \eta(x)} \Psi_{reg} \left[\boldsymbol{d}(\boldsymbol{x}) - \boldsymbol{d}(\boldsymbol{y}) \right] \tag{4.2}$$

where $\eta(x)$ is the neighborhood of the block of interest and Ψ_{reg} is another cost function.

In the test results of the project proposal, I demonstrated that estimation of vectors jointly using color components gives better results in terms of prediction error. As one can see there are four λs in the cost function and establishing optimal values for all of them is not trivial. In proposal, I have also tested parameter assignment method that depends on the variance of image intensities, however it did not give promising results, therefore I abandoned this method.

Since in our case we don't have the corresponding G component in the right image, I will use two color components, namely R and B, in order to estimate the disparity field.

Block matching is a very simple and efficient way of computing disparities however, it has various limitations. First of all, disparity field is not very well defined due to the assumption that all points in a block should have identical disparity values. Due to this assumption resulting disparity field of a block based estimator is piecewise -constant. This constancy creates a very big problem, especially near the edges.

Moreover, a dense disparity field[1] will be much more useful for reconstruction part of this project. Another problem is that precision of block matching is limited. Although one can use half or quarter or even less pixel precision disparity vectors, ideally infinite precision vectors would be more accurate.

Addressing all the problems, I decided to refine the disparity field using optical flow methods. Optical flow methods result in better described dense disparity fields with infinite precision. However, they cannot capture displacements of more than 2-3 pixels due to the Taylor expansion used in derivation. Therefore, it is appropriate to use the results of block-based estimation as a starting point.

4.2 Optical flow

The optical flow that I have utilized is the famous Lucas-Kanade method [7]. Similar to Horn-Schunck method [4], this method also works on optical flow constraint equation.

The difference is that the Lucas-Kanade method minimizes an error function in a local window with the assumption that pixels in this local window experience similar motion (which is disparity in our case). For every point in the image, a window is created around the point and the error function is minimized. Following this step, the motion vector calculated for the area is assigned only to the point of interest and window is shifted to the next point. This approach is akin to sliding block matching, but is superior in the sense of infinite precision of vectors. Although detailed derivations will be skipped in this report, it is appropriate to give at least the cost function to be minimized and the solution.

For a specific point x and a window W around x we would like to compute a d[2] such that the following function is minimized:

[1] Dense disparity field has a distinct disparity vector for every point in the image, as opposed to a single disparity vector for a block of points.

[2] Although it is a vector, we consider d as a scalar for this demonstrative derivation

$$E = \sum_{x \in W} |I_1(x + d) - I_2(x)|^2 \tag{4.3}$$

Expanding this equation around d using Taylor series we get

$$E \approx \sum_{x \in W} |I_1(x) + I_1'(x)d - I_2(x)|^2. \tag{4.4}$$

Since we are trying to minimize this equation with respect to d, we take a derivative with respect to d and make it equal to zero

$$\sum_{x \in W} 2I_1'(x) \left[I_1(x) + I_1'(x)d - I_2(x)\right] = 0. \tag{4.5}$$

Arranging this equation, the solution for d is obtained as follows:

$$d = \frac{\sum_{x \in W} I_1'(x) \left(I_2(x) - I_1(x)\right)}{\sum_{x \in W} I_1'(x)^2}. \tag{4.6}$$

Iterative solution of equation (4.6) will yield one disparity vector for each x position. The smoothness constraint that has been added in block matching is inherently included in this algorithm, therefore no additional smoothness term is required in the cost function. The smoothness can be controlled by the window size.

4.3 Reconstruction of the G component using the disparity field

The last step in the method is using the computed disparity field to reconstruct the G component. This is achieved by warping the G component of left image by the disparity field using the following formula

$$I_{G,right}(\boldsymbol{x}) = I_{G,left}(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})). \tag{4.7}$$

where $\boldsymbol{d}(\boldsymbol{x})$ is the disparity vector for the pixel position $\boldsymbol{x}$.

Chapter 5

Experimental results

5.1 Objective results

In order to evaluate the performance of the algorithm, tests were conducted using stereo pairs. For each stereo pair, G component of right image is dropped in order to simulate the problem and then G component is reconstructed using the algorithm proposed here. Following the reconstruction, reconstructed color image is compared with the original color image. Table 5.1 shows the PSNR values of the reconstructed images.

Along with the results, some ad-hoc solutions to the problem is also compared. The first row indicated by 'G component is lost' is the case when there is no G component and values in G component are padded with zeros. R$\rightarrow$ G and B$\rightarrow$ G are the tests using duplicating R and B components respectively. For $\frac{R+B}{2} \rightarrow G$ row, R and B components are averaged and assigned to G component. Although these components have comparable PSNR values to our results, the resulting pictures are totally inconsistent with the original picture in terms of color. In 'tsukuba' one of the adhoc solution gives the best results but this is incidental because tsukuba lacks color diversity, the background is almost black. Black color is represented with small values in R, G, B components therefore, by chance, average of R and B is close to original G value.

Figures 5.1-4 show the reconstructions for subjective comparison. Figure 5.5 shows the prediction errors for G components of reconstructions. Figure 5.6 shows the results of adhoc solutions for Teddy. One can easily notice that colors are inconsistent with the original image.

Results show that PSNR values of resulting reconstructions are greater than 30 dB. Also, it can be easily seen that optical flow based estimation yields 3-6 dB gain.

5.2 Tests with real data

I applied the proposed algorithm to many stereo pairs from Mars. In each pair, all components of the left image, and the R and B components of right image were avail-

Name of	Teddy	Tsukuba	Venus	Sawtooth
G = 0	9.91	14.33	11.60	11.58
R→ G	18.33	24.92	22.53	25.31
B→ G	17.65	27.93	16.09	24.22
$\frac{R+B}{2}$→ G	19.19	33.91	30.22	22.62
Block-based estimation	30.70	27.27	29.17	27.96
Optical flow based estimation	36.40	32.96	35.15	30.90

Table 5.1: PSNR values (in dB) of reconstructed images

able and I have reconstructed the missing G component. Since original data are not available, PSNR evaluations are impossible. Reconstructed images and corresponding color left images are shown in Figures 5.7-10 for subjective evaluations. Structures are well matched, though there are few color mismatches present in objects. These are due to presence of occlusions and incorrect estimation of disparity.

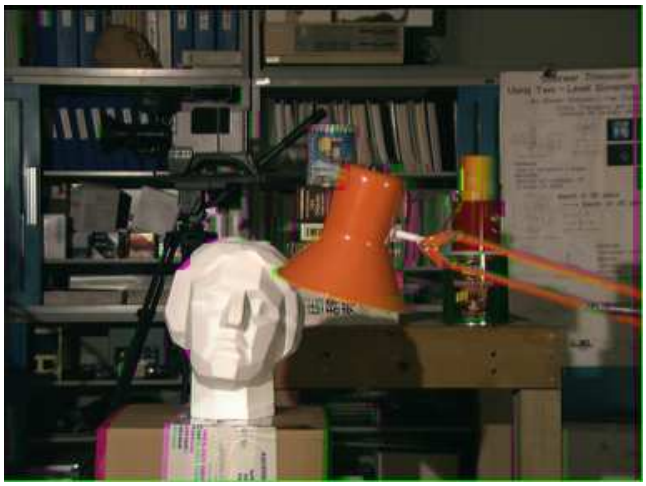
(a) Original right image

(b) Block based reconstructed image

(c) Optical flow based reconstructed image

Figure 5.1: Subjective results for 'Teddy' stereo pair

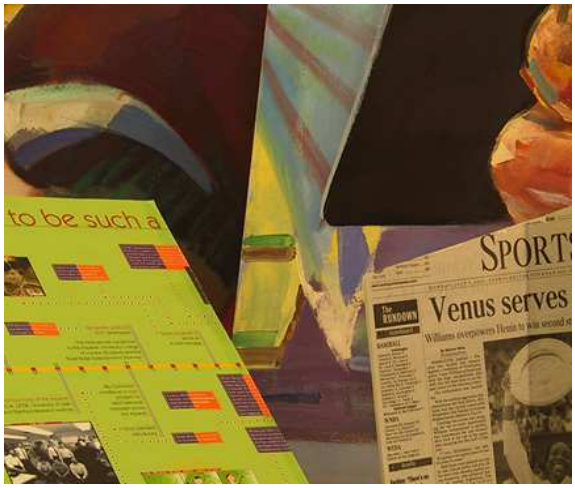
(a) Original right image



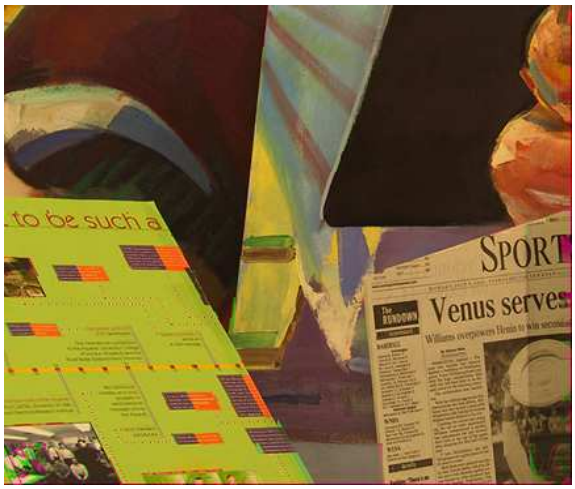
(b) Block based reconstructed image

(c) Optical flow based reconstructed image

Figure 5.2: Subjective results for 'Tsukuba' stereo pair



(a) Original right image



(b) Block based reconstructed image

(c) Optical flow based reconstructed image

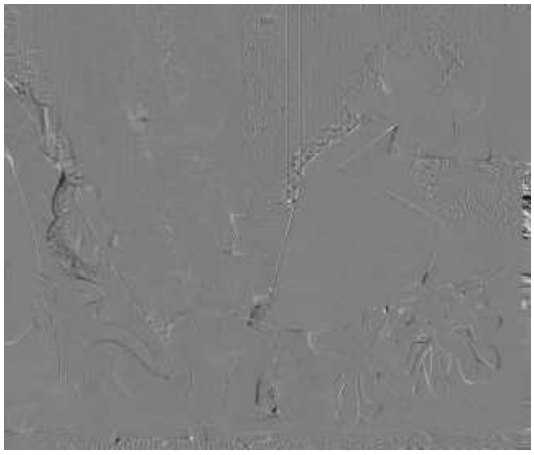
Figure 5.3: Subjective results for 'Venus' stereo pair

(a) Original right image

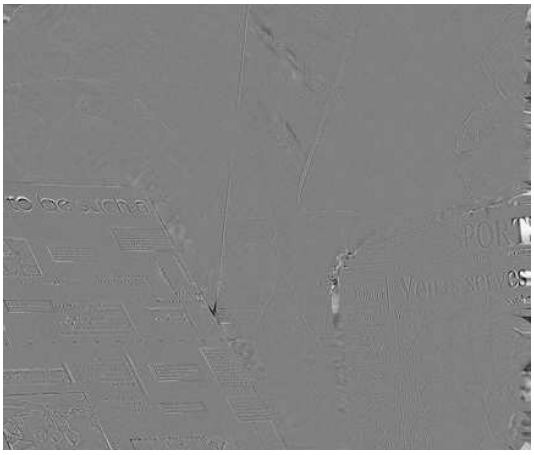
(b) Block based reconstructed image

(c) Optical flow based reconstructed image

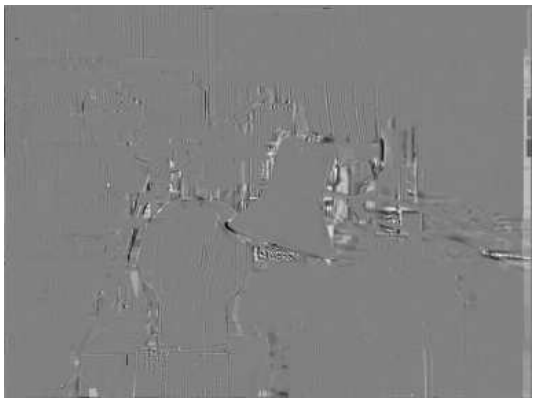
Figure 5.4: Subjective results for 'Sawtooth' stereo pair



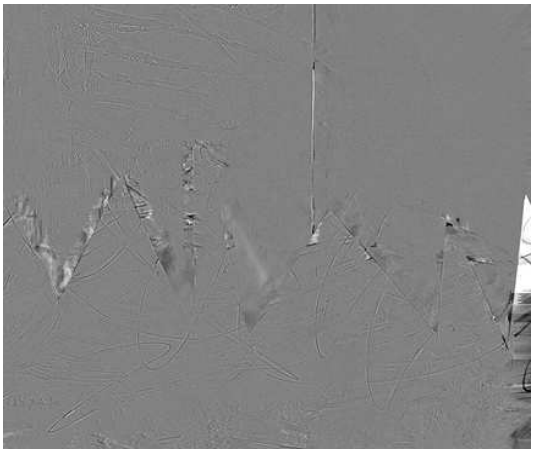
(a) Teddy



(b) Venus

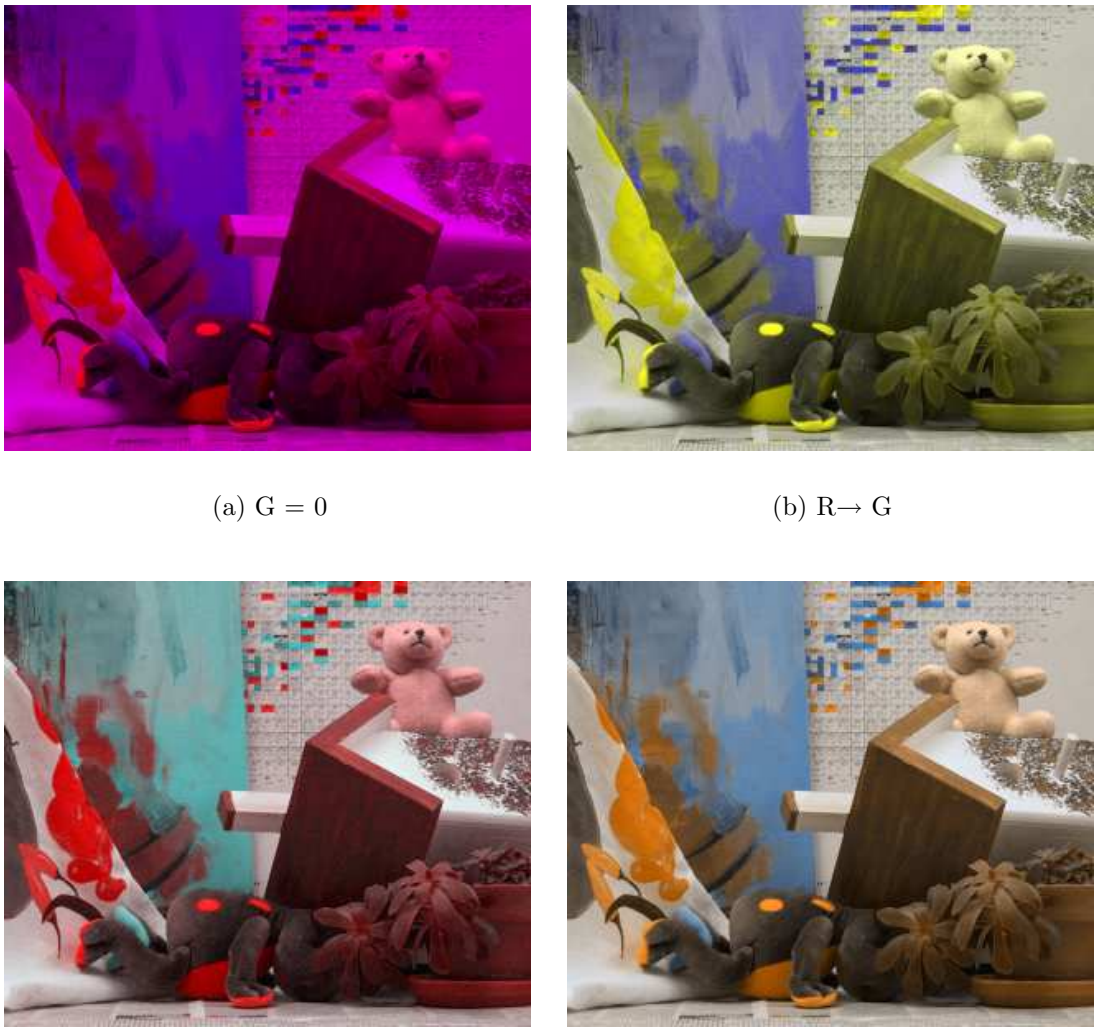


(c) Tsukuba



(d) Sawtooth

Figure 5.5: Prediction errors of G components of stereo pairs



(a) G = 0

(b) R→ G

(c) B→ G

(d) $\frac{R+B}{2}$→ G

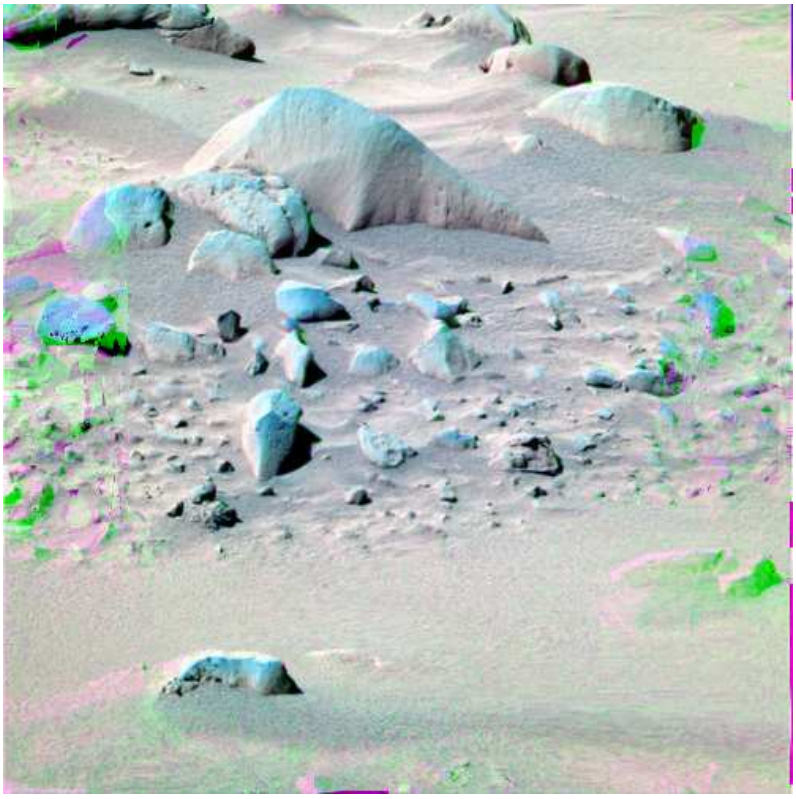
Figure 5.6: Resulting images using adhoc solutions for Teddy pair

(a) Left image

(b) Reconstructed right image

Figure 5.7: Subjective results for 'mars image7'

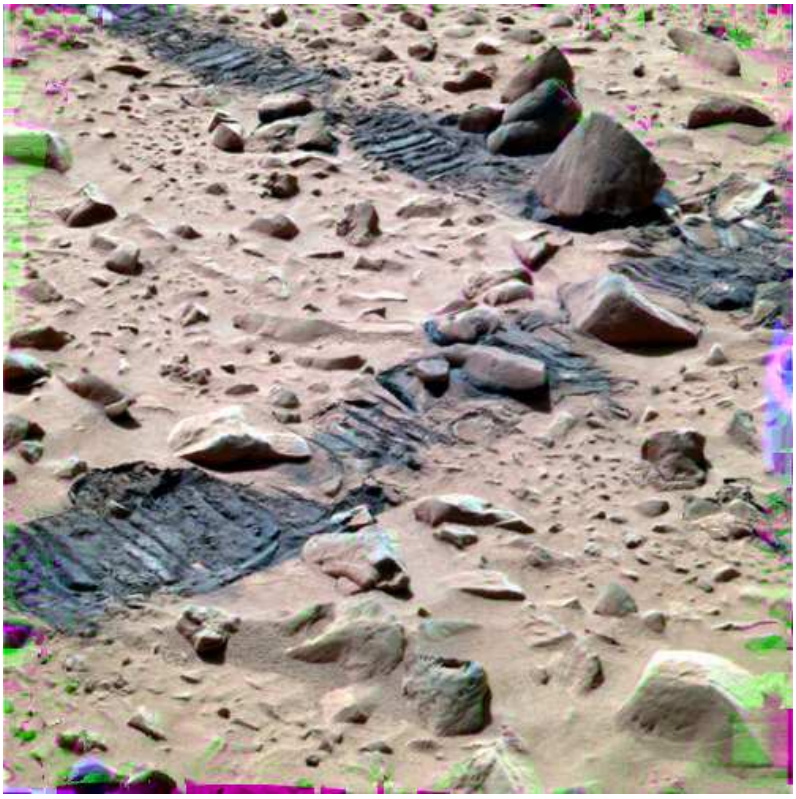
(a) Left image



(b) Reconstructed right image

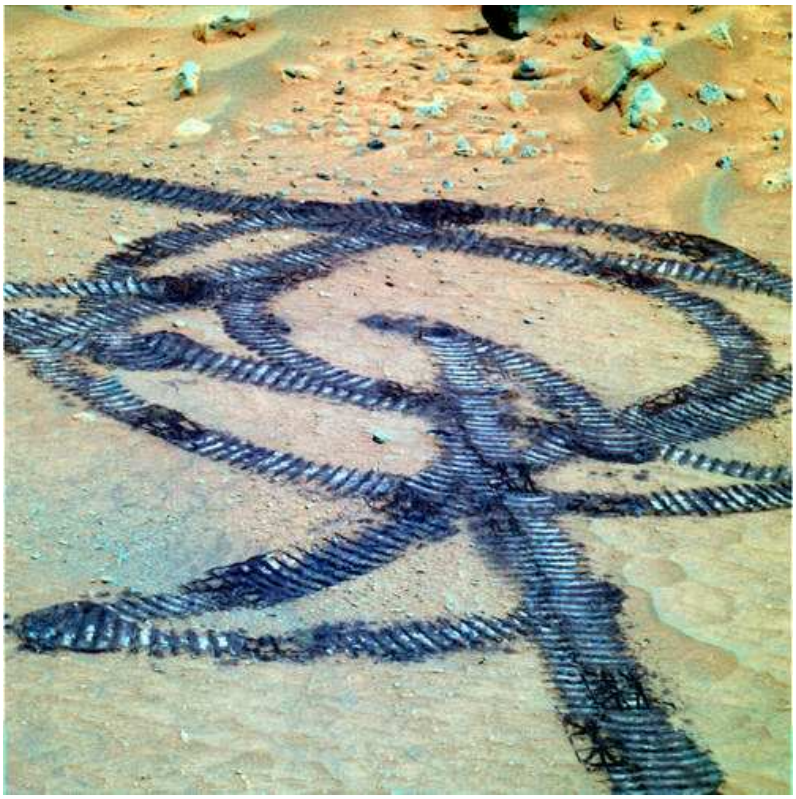
Figure 5.8: Subjective results for 'mars image6'

(a) Left image



(b) Reconstructed right image

Figure 5.9: Subjective results for 'mars image9'



(a) Left image

(b) Reconstructed right image

Figure 5.10: Subjective results for 'mars image10'

Chapter 6

Problems encountered

The first problem experienced was at image boundaries. As in all image processing tasks, image boundaries create additional difficulty. The main issue is that pixels near the boundaries cannot be matched during estimation process.

However, this problem can be solved to some degree. Since the disparity field is smooth, it can be assumed that disparity field will be similar along the border of right image. Therefore, one can easily copy motion vectors from inner pixels to the boundary pixels. If the cameras are perfectly parallel, like in the images in objective results, this solution will solve problems along the top, bottom and left border of images. However, again considering images used in objective tests, the right side of the right image will still be a problem. Figure 6.1 illustrates this issue.

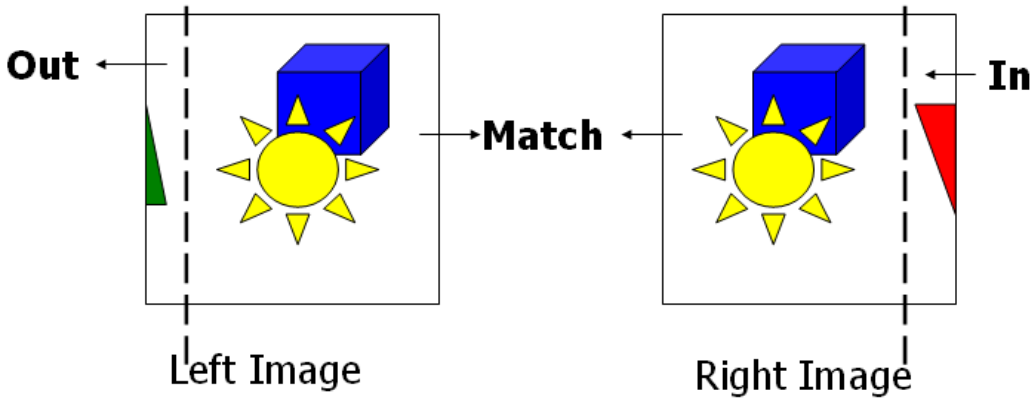


Figure 6.1: Illustration of boundary problems

Let's jointly divide the stereo pair images in the figure with dashed lines. The leftmost region in left image will disappear in the right image, due to the distance between cameras. This area will not cause any problem since it is not present in the right image. Therefore, the green object can safely be ignored. The common views in the images should not cause any problem since the disparity estimation part can match pixels in this area. The rightmost area in the right image is the main

problematic area. Since this part does not exist in the left image, there is no way to match the red triangle during estimation part. Since there is no additional disparity information available in this area, the problem simplifies to another problem: Given R and B component of a part of the image, find G component of the image. R, G, B are uncorrelated in intensity values therefore solution is not trivial. However, edge information in R and B components can be used at least to preserve edge information. The assumption here is that if there is an edge in R and B components, it should also be present in G component. However, if there is special formation such that R and B did not have any edges, but G had an edge, this assumption will also fail, but this should be rarely happening in real-world images.

The problem mentioned above occurs only on one boundary of the right image in parallel camera setup. These artifacts can be seen in Figure 10 on the right side of the images. There is a significant number of dark and bright pixels indicating mismatches.

However working with Mars images is more complicated. The PanCam is said to be a toed in camera. As far as it can be seen from available components of stereo images, it is not perfectly toed-in. In a toed-in camera setup, optical axes of cameras intersect at a point, which is called convergence point. This point has a zero disparity value. After a careful examination of available components of Mars images, we could not find a zero disparity point anywhere. Since PanCam is neither parallel nor toed-in the problem mentioned above occurs in all boundaries of the image. Therefore there are significant artifacts near the boundaries.

Another problem are large disparity values. The objects in the images are very close to cameras, therefore disparity values sometimes reach 220 pixels in x direction and 30 pixels in y direction in a 500x500 image. This results in a small field of common view. Even if we were able to reconstruct a perfect right image, it would be impossible for a user to view it, because human brain cannot fuse large disparity values. Therefore, it is appropriate to crop the images such that disparities are reduced and the field of common view is maximized.

I have cropped the reconstructed images as mentioned above and subjectively tested them on a liquid-crystal shutter glass stereo system. 3D experience was comfortable and colors were well matched.

Chapter 7

Future work

The results in this work are encouraging in terms of image quality and PSNR values, however there is still room for improvement.

First of all, the solution of the current method may be improved by adjusting some of parameters in the system. In two-step estimation part, there are many parameters such as block size, regularization factor, weights of color components, number of iterations in Lucas-Kanade algorithm etc. Combining all of these parameters and finding an optimal set for each image is almost impossible. However, one can try to test different values for better results.

Additional work can be done such that edges (or other features) of all color components are aligned as they were aligned in left image. Although it would be the case, we don't explicitly enforce alignment of the same features in all color components, because in general case, there may be areas where only two components have aligned features. Therefore, we can force points in the right image to have a similar alignment with the alignment in left image.

In other words, if R and G have corresponding features in the left image, they should have a corresponding feature in the right image too. In a similar way if B and G have corresponding features in the left image, they should have corresponding feature in the right image. Therefore the new cost function should force the alignment of features if there is an alignment in the right image.

We can write the new cost function as follows. Let left image be image 1 and right image be image 2;

$$E = \sum_{i \in R,B} ||I_{1,i}(x) - I_{2,i}(x+d)|| + \Psi \left[||\nabla I_{1,i}(x) - \nabla I_{1,G}(x)|| \right] . ||\nabla I_{2,i}(x+d) - \nabla I_{2,G}(x+d)||. \tag{7.1}$$

The selection of Ψ is essential here. I would prefer to use the Lorentzian function, due to its characteristics. It gives higher output values for small input values, and smaller outputs for larger inputs, and the function 'fades' quickly. A plot of Lorentzian function is given Figure 7.

The first term in the cost function is the usual data fidelity term. The interesting part is the second term. Let's examine this part closely with various conditions:

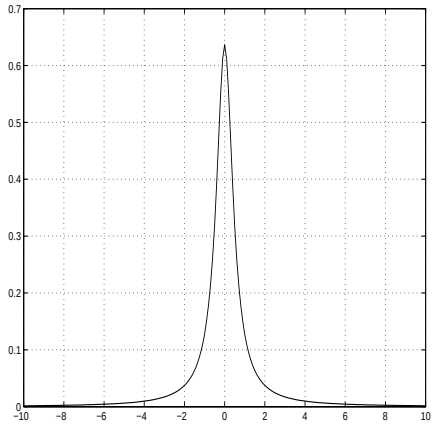


Figure 7.1: Lorentzian function

In case of a feature alignment of R with G and B with G, the weights of feature aligning terms will increase because input value to Lorentzian will be small. In case of a feature alignment in only one component, for example R with G, weight of alignment in B with G will decrease but weight of alignment in R with G will increase. Therefore, as a result, this cost function is expected to not only match the data (first term) but also align the features (second term).

I have started implementing this part, however I was unable to complete by the time this report is written.

Chapter 8

Conclusions

In this report I have demonstrated a method to create the G component of right image in stereo pairs. The approach can be extended to any color components. The objective results show a PSNR value of at least 30 dB. Tests with Mars images show proper reconstruction that can be visualized in a 3D system. Images shown in this report can be downloaded from http://iss.bu.edu/ince/Research/Mars/.

Acknowledgments: I would like to thank Prof. Janusz Konrad for his help in all stages of this work. Stereo pairs used in objective tests are property of Middlebury College Stereo Vision Research Group (http://www.middlebury.edu/stereo). Mars images can be downloaded from NASA Jet Propulsion Laboratory website at http://origin.mars5.jpl.nasa.gov/gallery/all/. I would like to thank Prof. Andrew Woods for his help about Mars images.

Bibliography

[1] L. Alvarez, R. Deriche, J. Sánchez, and J.Weickert. Dense disparity map estimation respecting image discontinuities : A PDE and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13:3–21, 2002.

[2] L. Alvarez and J. Sánchez. 3-d geometry reconstruction using a color image stereo pair and partial differential equations. Technical Report 917, Cuadernos del Instituto Universitario de Ciencias y Tecnologas Cibernticas, 2000.

[3] P. Golland and A. M. Bruckstein. Motion from color. *Computer Vision and Image Understanding: CVIU*, 68(3):346–362, 1997.

[4] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artif. Intell.*, 17:185–203, 1981.

[5] A.K. Jain. *Fundamentals of Digital Image Processing*. Information and System Sciences Series. Prentice Hall, New York, NY, 1989.

[6] J. Konrad. Use of colour in gradient-based estimation of dense two-dimensional motion. In *Proc. Conf. Vision Interface VI'92*, pages 103–109, May 1992.

[7] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.

[8] K. Mühlmann, D. Maier, J. Hesser, and R. Männer. Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation. *IJCV*, 47(1):79–88, APR 2002.

[9] Y. Wang, J. Ostermann, and Ya-Qin Zhang. *Video Processing and Communications*. Prentice Hall, 2001.