Application of Belief Propagation Algorithm in Early Vision System

Chung Lee, Ph.D.
Department of Information Technology
Mongolia International University
Ulaanbaatar, Mongolia
chunglee@miu.edu.mn

***Abstract*— The Belief Propagation (BP) algorithm presents a unique approach to solve vision problems. By iteratively generating and passing messages between immediate neighbors, the entire set of pixels in an image achieve the minimum energy configuration. Based on the solid mathematical basis and continued research led by Tappen and Freeman [1], the algorithm has been adopted to the solution of many computer vision problems.**

This study further expands the algorithm. Firstly, the original algorithm was modified to overcome over-smoothening in image restoration process. By modifying parameters and embedding a recorder that tracks the magnitude of messages generated, new algorithm shortens the restoration steps at the optimum point where image boundaries are preserved. This made the algorithm more useful as a practical restoration tool.

Secondly, various improvements were integrated to the method to speed up the restoration in stereoscopy(Fast Belief Propagation). Different message update schemes, such as an alternate message coordinate system and deletion of bad messages based on statistics have been implemented. Furthermore, detailed studies on parameters, cutoffs and functions used in the algorithm have been made. Consequently, two new modes of BP algorithm have been proposed and tested, which showed time saving of 30 to 60% compared to the algorithm suggested by Felzenszwalb and Huttenlocher[2].

Keywords—Early Computer Vision; Belief propagation; Image Restoration;Stereopsys

I. INTRODUCTION

Modern society demands intelligent machines that can mimic human behavior. Since machines cannot reason by themselves, Artificial intelligence (AI) techniques are applied to teach complex machines how to think like human beings. Many AI algorithms exist to solve different problems with varying degree of success. This machine-human gap is very acute in the computer vision arena that includes study of image processing, stereoscopy, segmentation and image recognition among other topics.

One promising approach in AI field is the Belief Propagation (BP) algorithm. The algorithm has been successfully used in statistical physics, error-correcting coding theory, and signal processing. The use of this technique in the vision field came only recently by the efforts of Freeman and Yedidia, Weiss and Freeman [3] and Scherer-Negenborn [4].

While the belief propagation methods are known to be notoriously slow and inefficient for vision problems, the recent paper by Felzenszwalb and Huttenlocher [2] presents mathematical simplifications that expedite the execution of BP algorithm. They shows the BP algorithm can be used in low level vision processing including the restoration of a noisy image. The only weakness is that their restoration performed poorly due to overly smoothing the image. With a right combination of parameters, this method can yield more acceptable performance.

In this research, efforts are made to find the optimal combination of parameters so as to use BP algorithm as a practical restoration tool by identifying the approximate end point and stops the iteration. Determining the optimum termination point of a restoration operation is critical not to lose the image detail. The insight gained by this study will aid in making a BP-based restoration tool that extends existing restoration methods such as spatial domain kernels or frequency domain filters. In addition, two BP algorithms faster than the original form are suggested and tested. These new algorithms, when further refined, will be useful in time-critical applications.

II. LITERATURE REVIEW

A. *Computer Vision process*

The computer vision is a very active development and diversified field. It includes general image formation, noise removal, edge finding, image compression, texture determination, segmentation, stereoscopy, motion tracking, algorithms and other topics. A typical Computer Vision process undergoes the following stages;

1) Low level (Early Vision) : Image capture $\rightarrow$ Preprocessing $\rightarrow$ Region extraction $\rightarrow$ Region labeling
2) Mid-Level (Medium Vision): Region construction and iteration
3) High level (Late Vision): Parameterization $\rightarrow$ Identification $\rightarrow$ Conclusion $\rightarrow$ Iteration $\rightarrow$ Storage

In early vision system, several functions such as noise reduction, filtering, stereoscopy construction are important *as explained in Gonzales [5] and Forsyth [6]* and this research focuses on these steps.

B. Stereoscopy

Human sees real world objects in 3-D stereo but typical display is no two-dimensional flat surface and therefore the fidelity of image is not reached. Different algorithms such as Simple Color Matching, Dynamic Programming, Correlation Windows, Cooperative Methods and Graph Cut Method are used. [5][7]

C. Belief Propagation (BP)

The BP algorithm aims to solve an "inference" problem. To get the final (marginal) probability of a particular outcome, one has to sum up all probabilities for each node that gives the outcome. A vision problem makes inference from visual images. The two-camera stereo problem, for example, aims to figure out the three-dimensional surface structure from two pictures shot in an angle between them.

In general, we infer a set of characteristics $\{f_p\}$ from the observed set of data $\{i_p\}$ where p denotes a typical data point(a node). The p may be a pixel in aforementioned stereo problem. The solution is attempted when the relationship between f_p and i_p is known or estimated and their relationship is expressed as; [8][9]

Probability $\{f \mid i\}=(1/N)*\prod D(f_p, i_p) * \prod W(f_p, f_q)$ where,

N = the normalization factor

$D(f_p, i_p)$ = data cost at the pixel p

$W(f_p, f_q)$ = discontinuity cost between the pixel p and q

The same equation can be derived from Markov Random Field (MRF) models. For observation i,

Prob (f|i) = k *Prob(i|f) *Prob(f) where,

Prob(f|i) = conditional probability of f when i is given

k= normalization constant

Prob (i|f) = probability of observing i when f is given

Prob (f) = probability of f occurring.

The MRF model is a good framework for expressing tradeoff between spatial coherence and fidelity to data. This equation is known to be a NP problem. Even with modern super computers, this equation cannot be solved in reasonable time. Efficient algorithms such as Graphcut [10] and BP by Felzenswalb and Huttenlocher [2] and Sun et al [11] are used in the stereo vision problem to generate an approximate solution.

D. Message Generation

In most vision problems the observed data $\{i_p\}$ are given at the beginning and fixed for the duration of the attempted solution. According to Veksler [12], this pixel intensity shows the probability as follows;

$P(\{f_p\})=(1/N)* \prod D(f_p) *\prod W(f_p\ f_q)$ where,

$P(\{f_p\})$ = final probability of entire set of pixels

N = a normalization factor

$\prod D(f_p)$ = product of data costs of all pixels

$\prod W(f_p\ f_q)$ = product of discontinuity costs of all neighboring pairs.

As given Fig. 2.1, at each nodes, these messages are received from four immediate neighbors (up,down,left,right) and the probability of each state is updated. Weiss and Freeman[12] expanded as,

$m_{pq}(f_q) = c*\max_p(W(f_p, f_q)* D(f_p)* \prod^{\backslash} m_{qp}(f_p))$, where,

$m_{pq}(f_q)$= message from p to a neighbor, q, about the probability of q being f

c=normalization constant

$\max_p$= find the maximum using f_p as the variable

$W(f_p, f_q)$ = the discontinuity cost

$D(f_p)$ = the data cost

$\prod^{\backslash} m_{qp}(f_p)$ = multiply messages into the p with the exception of one from q.

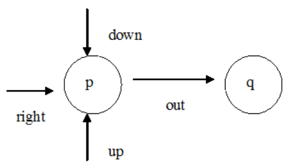


Fig 2.1 : Message Generation from p to q.

E. BP on Restoration

Numerous researches were published to show the application of BP on image filtering and image restoration. Major algorithms are listed here.

a. Geman and Geman algorithm

Geman and Geman[9] used simulate annealing techniques on MRF equations. Using Gibbs distribution, the probability is expressed as,

Probability(w) = $1/Z * e^{-U(w)/T}$ where,

(1/Z)= normalization factor

U = potential energy of w configuration

T = annealing temperature

w = one possible configuration=$\{x_1, x_2, \ldots, x_n\}$.

b. Felzenszwalb and Huttenlocher algorithm

Felzenszwalb and Huttenlocher[2] developed several mathematical techniques to accelerate convergence of BP without sacrificing mathematical accuracy

$Dp(fp) = \lambda\min((I(p) - fp)^2, d)$ where,

fp = possible intensity value at pixel p

Dp(fp) = data cost

λ = ratio, data cost/discontinuity cost(set to 0.05)

d = data cost cutoff = 10000

F. *Need for further research on Restoration*

Geman and Geman [9] took 1000 iteration cycles to restore an image corrupted with noise. This rate is too slow to be useful in image restoration. Felzenszwalb and Huttenlocher, on the other hand, sped up the process but the restored images showed considerable loss of details. The BP iteration had proceeded too fast and too far resulting in destruction of fine features. Firstly, it was done by finding optimum stopping rule. In this study, this aspect was investigated in depth.

G. *Belief Propagation on Stereoscopic Images*

The BP algorithm works well in stereo problems due to its powerful message passing capability [12]. Tappen and Freeman [1] compared the Graph Cuts algorithm [3] and the Belief Propagation algorithm on stereo disparity calculations. Veksler[12] reported the discontinuity cost between two pixel values following Potts model and demonstrated satisfactory result. Sun et. al [11] used the product form of the Belief Propagation equation. Felzenszwalb and Huttenlocher [2] used BP algorithm on stereo problem. Further research on stereoscopy was conducted by Kaneda [13] and Zitnick[14] and more systematic research was done by Scharstein and Szeliski [15].

III. RESEARCH GOALS AND METHODS

The main purpose of this study was to improve the efficiency of BP algorithms so that early processing of computer vision system can be completed in real time. Due to the prevalence of computer vision system in various aspects of daily life, such processing demands speedy processing. To achieve this goal, we needed to find optimal number of iterations in BP process and apply statistical technique. Further algorithmic improvement was attempted. Further elaboration of these goals are given in section IV. A common computing environment with 2G RAM and 2Ghz processor was sufficient to develop this research. The original source program comes from the article by Felzenszwalb and Huttenlocher[2].

IV. RESULTS

A. Termination point selection

The original restoration process reported by Felzenszwalb and Huttenlocher [2] shows significant loss in detail as shown by Fig 4.1 mainly due to over-smoothing. The main reason for this over-smoothing is due to their algorithm, which runs five iteration cycles on the five-level multi-grid (pyramid). Figure 4.2 shows the transition of images by iteration steps. This research is seeking optimal stopping point where image fidelity is reached but not over-smoothed. This point can clearly as seen from Fig 4.3, the restoration did not improve much after about 5th iteration. identified by calculating and plotting the change rate of message magnitude as shown in Figure 4.3.

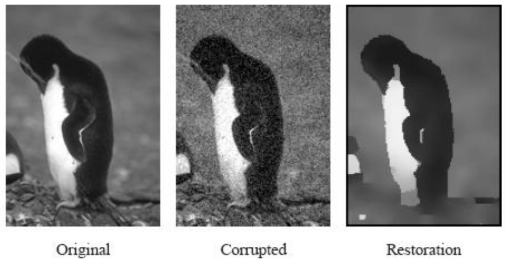


Fig 4.1 Restoration of Penguin image [2]

This corresponds approximately 1% of change in message magnitude. Most noises are treated and the significant details remain. Two improvements are made to the algorithm by Felzenszwalb and Huttenlocher[2] . The first is to use linear discontinuity function to slow down the smoothing process. The second is to stop the iteration cycle when 1% change in message magnitude is reached.

Another example test using a car image also showed 5th or 6th iteration being sufficient for BP. This result is shown in Figure 4.4 and 4.5.

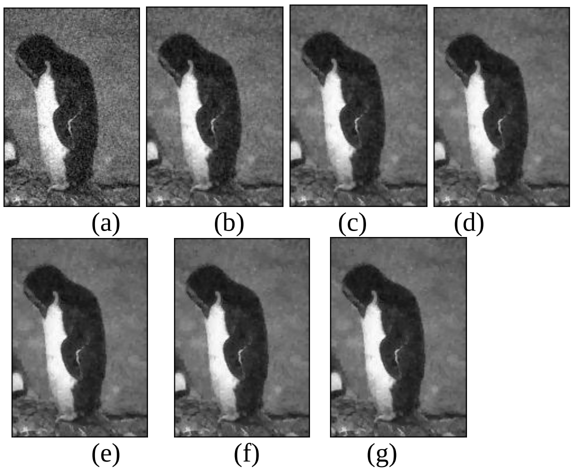
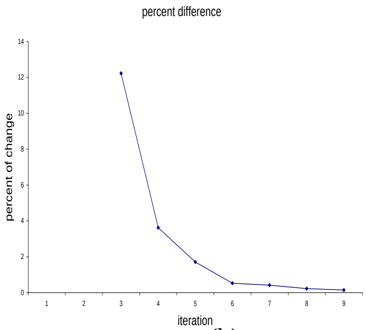


Fig 4.2 BP with the linear discontinuity function (a) After 1 iteration, (b) 2 iterations, (c) 3 iterations, (d) 4 iterations, (e) 5 iteration, (f) 6 iterations, (g) 7 iterations

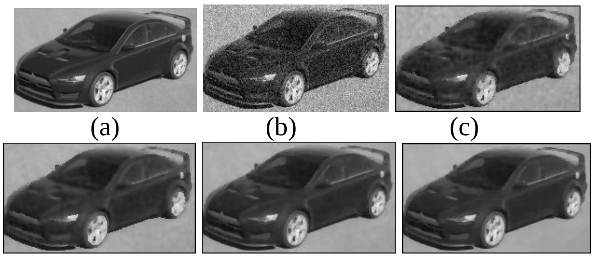
iteration	magnitude	perc diff
1	3066	
2	-4091	-233.4
3	-4591	12.2
4	-4757	3.6
5	-4838	1.7
6	-4863	0.5
7	-4883	0.4
8	-4894	0.2
9	-4901	0.1

(a)



(b)

Fig 4.3 Table (a) and plot (b) for percent change of message (linear discontinuity function)



(a) (b) (c)

(d) (e) (f)

Figure 4.4 Original (a), Noise added(b), 3 iterations(c), 5 iterations(d), 6 iterations(e) and after 10 (f)

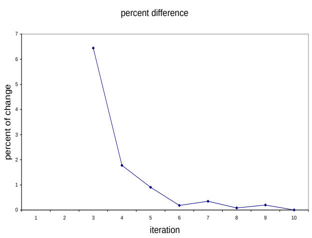
B. Speeding up BP algorithm running

To speed up the BP algorithm, several different approaches were attempted as follows—

a) Immediate Update Scheme (one dimensional)

In this acceleration scheme, among the three messages come into the pixel, only the message in the same direction as the output message is chosen to interact with the data cost at the pixel as modeled in Figure 4.6.

iteration	magnitude	perc diff
1	3373	
2	-5509	-263.3
3	-5864	6.4
4	-5968	1.8
5	-6022	0.9
6	-6033	0.2
7	-6054	0.3
8	-6059	0.1
9	-6071	0.2
10	-6071	0.0



(a) (b)

Fig 4.5 Percent change of message (car – linear discontinuity)

The resulting probability vector is transformed through the linear discontinuity cost function and the resulting message is immediately applied to the next pixel. The forward sweep, the reverse sweep, the up sweep and the down sweep are made in turn. The result is shown in Figure 4.7 and was not very satisfactory.

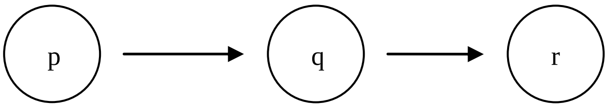


Fig 4.6 One dimensional message update from p to q to r node

b) Eight Directional Sweep

The original work by Felzenszwalb and Huttenlocher[2] used four coordinate system as with other literature on Belief Propagation Algorithm.

(a) (b) (c)

Fig 4.7 One dimensional message update (one sweep to right direction) showing Original (a), after 10 sweeps (b) and after 100 sweeps (c)

The eight coordinate system, or the seven message update scheme, is studied. There are seven incoming messages; for the case of generating the right message as shown in the figure 4.8, the messages from the positions, going clockwise, “up”,”up-right”, ”down-right”, “down”, “down-left”, “left” and “up-left”. It generates eight messages per pixel from these seven incoming messages and the data cost of the pixel. New quadratic sweeping scheme is made, where only four out of sixteen pixels are updated so as to prevent any interactions between messages. This translates into the following pseudocode given as,

```
for (int t = 0; t < iter; t++) {
    st=t%4;
    switch(st){ //new checkerboard for 8 coord
        case 0: xstart=1; ystart=1; break;
    case 1: xstart=2; ystart=2; break;
    case 2: xstart=1; ystart=2; break;
    case 3: xstart=2; ystart=1; break;
  }
  for (int y = ystart; y < height-1; y+=2) {
     for (int x = xstart; x < width-1; x+=2) {
              ….messages here……
        }
    }
```

(a) (b) (c)

Fig 4.8 Penguin image after (a) 4 iteration, (b) 8 iterations and (c) 40 iterations

c) Statistical BP

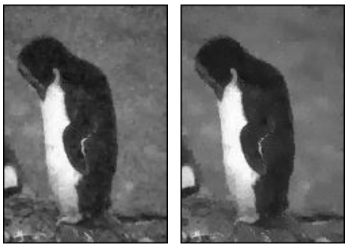
To speed up the process with optimal result, statistical treatments were applied to the algorithm for detecting and removing two-tailed outliers.

Since we have three incoming messages and the data cost at the pixel, four different vectors can be tested with this criterion to identify outliers and delete them. The minimum of each vector is chosen to represent the vector.

The standard deviation formula, $Sd = \text{sqrt}([\Sigma x^2-(\Sigma x)^2/n]/(n-1))$ is substituted to the rejection criteria, $(x-\Sigma x/n)>2*Sd$ which is transformed as $d^2 > 4* [n*(\Sigma x^2)-(\Sigma x)^2]/[n*(n-1)]$, where $d= (x- \Sigma x/n)$. The initial messages start at zero probability at all possible states; the initial mean at the start is zero. Therefore, all message changes are rejected. To overcome this problem, the regular BP algorithm is run through one full iteration cycle before the statistical rejection is applied. This technique was used for restoration and the outcome is shown in Figure 4.9 for penguin and Figure 4.10 for car image.

The third example for the restoration is a dog seen in Fig 4.11. This image is rather difficult to restore due to heavy interactions between the dark and the light boundaries present in the original picture. Yet the improved algorithm performed reasonably well. The message magnitude change is shown in Table 4.1 for Dog and Car images.

The 1% change occurred again between the 5^{th} and the 6^{th} cycle. This, together with previous examples, may suggest that the rule of thumb is to take the image after 5^{th} iteration as the best image. Further validation may be necessary to prove this claim.



(a) (b)

Fig 4.9 Penguin image (a) after 5 iterations, (b) after 10 iteration

(a) (b)

Fig 4.10 Car images after the (a) 5th and (b) 10^{th} iterations

C. Fast Belief Propagation (FBP)

Further speeding up of the BP algorithm was attempted. After trying various changes in the parameter to speed up the algorithm, three facts are observed. The first observation is that either the discontinuity function or the discontinuity cutoff can be deleted. The algorithm still gives a reasonable result as seen in above description. The second observation is that even though the lambda, the ratio between the data cost and the messages, is very sensitive but can be deleted and the discontinuity cutoff can be adjusted to compensate for this deletion. The third observation is that one may be able to run less number of iterations in each layer of the pyramid as long as more layers added to the pyramid.

It is given that the each change suggested above does degrade the final disparity map. However, the loss of fidelity is compensated by the gain in speed. To further maximize the speed gain, float variables inside the code are changed to integer values. The entire algorithm is altered to run only in integer operations.

It is given that the each change suggested above does degrade the final disparity map. However, the loss of fidelity is compensated by the gain in speed. To further maximize the speed gain, float variables inside the code are changed to integer values. The entire algorithm is altered to run only in integer operations.

There are two Fast BP algorithms that are tested for speed and the quality of final solution. The first one, Fast BP (a), is made of no discontinuity cost function and no lambda. It runs five iteration cycles on each of five layers of pyramid. The second algorithm, Fast BP (b), is made of no discontinuity cost function transform and no lambda as in (a). However, it runs only two iteration cycles per layer, that is equivalent to full update on each pixel in this bipartite graph (checkerboard) mode. It compensates this reduction in iteration by increasing the number of pyramid layers to ten. From the running time given in the Table 4.2, the Fast BP (a) algorithm runs at 69% of the original algorithm and the Fast BP (b) at 33% of the original algorithm. The time saving at Fast BP(b) is considerable.

Table 4.1 Message magnitude change at each cycle for (a) dog and (b) Car images

iteration	magnitude	perc diff	iteration	magnitude	perc diff
1	3726		1	8404	
2	-4796	-228.7	2	-11584	-237.8
3	-5562	16.0	3	-12837	10.8
4	-5822	4.7	4	-13178	2.7
5	-5934	1.9	5	-13362	1.4
6	-5976	0.7	6	-13379	0.1
7	-6003	0.5	7	-13443	0.5
8	-6024	0.3	8	-13441	0.0
9	-6033	0.1	9	-13481	0.3
10	-6046	0.2	10	-13467	-0.1

V. CONCLUSION AND DISCUSSION

A. Summary of work

The Belief Propagation (BP) algorithm is basically an application of relaxation approach. It starts with observed data and then pixel data are continuously improved by receiving messages generated by surrounding pixels.
The message is not a scalar, but is a vector of probability that has been transformed by a function that represents discontinuous nature of the environment.
After receiving messages, a pixel generates messages toward its neighbors. This receiving and generating process (“gossip”) continues with multiple iteration cycles to the end point. This result matches with that of Freeman and his coworkers. [1], [8], [16]. After the final

iteration, an optimum value is suggested for each pixel and the pixel value is replaced with this new value. Since the BP algorithm goes through many iteration cycles to account for the entire image, the final result should be better than a localized technique which only considers values of the immediate neighbors.

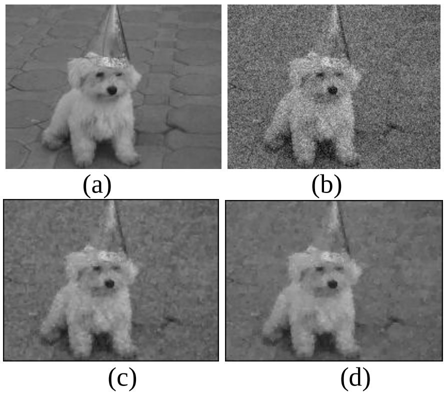


Fig. 4.11 Dog Image : Original (a), 20% Gaussian noise added(b), After 5 iterations(c) and after 10 iterations (d)

Table 4.2 Running time (sec) of algorithms

algorithm	Tsukuba	fraction	2xTsukuba	fraction
original	2.22	1	9.03	1
FBP(a)	1.53	0.69	6.43	0.71
FBP(b)	0.74	0.33	3.03	0.33

When given correct parameters, this algorithm can overcome regional bias and generate the optimum solution for the entire image. Devillers[17] and Valgaerts[18] did research work on this respect.

There are two versions of improved algorithms, FBP(a) and FBP(b). The difference between the two is that FBP(a) only goes up to 5 levels in the data pyramid but FBP(b) to 10 levels. In addition, FBP(a) performs 5 iterations per level while FBP(b) does only 2 iterations per level. In other words, FBP(b) is made to run faster than (a) while giving degraded final result. The original algorithm by Felzenszwalb and Huttenlocher[2] uses 5 level pyramid and performs 5 cycles of iteration much like FBP(a). The differences are FBP(a) skips the discontinuity function transform and does calculations only with integers.

One additional improvement was made on (b) by getting rid of the mathematical division in the message normalization step and simply replaced the average value with the minimum value. The speeding-up gained by FBP(a) and FBP(b) algorithms are about 30% and 60%, respectively. The FBP(a) gives the result almost similar in quality to that of the paper by Felzenszwalb and Huttenlocher [2] but with significant speed improvement. This is important in many real-time applications. Recently, Dickscheid et al. [19] reported their application of this approach on local features.

B. Future Research

The improved algorithms suggested by this research need to be tested with a variety of sample images to validate the improvements in terms of better restoration quality and faster speed.

Another area of improvement is to add more parameters such as lighting and occlusion information to fine tune the algorithm. Additional parameters can increase the accuracy, but they will slow down the algorithm. A generic algorithm can be applied to find the best set of parameters that can work in the Belief Propagation (BP) frame work, especially improving the data cost function and the discontinuity cost function.

REFERENCES

[1] M. Tappen and W. Freeman. “Comparison of Graph cuts with Belief propagation for Stereo, using identical MRF parameters” IEEE Intl. Conference on Computer Vision (ICCV), Nice, France, October, 2003.

[2] P. Felzenszwalb and D. Huttenlocher, “Efficient Belief Propagation for Early Vision,” Proceedings of IEEE CVPR, 1, 261-268, 2006.

[3] J. Yedidia, W. Freeman and Y.Weiss, “Understanding Belief propagation and its Generalizations,” International Joint Conference on Artificial Intelligence (IJCAI 2001), Distinguished Papers Track, 2001.

[4] N. Scherer-Negenborn, R. Schaefer, “Model Fitting with Sufficient Random Sample Coverage. I JCV (5) 120-128, 2012.

[5] R. Gonzalez, R. Woods and S. Eddins, “Digital Image Processing Using MATLAB”, Pearson Education, New Jersey, 2004

[6] D. Forsyth and J. Ponce, “Computer Vision A Modern Approach, Pearson Education,” New Jersey, 2003.

[7] Y. Weiss and W. Freeman, “On the Optimality of Solutions of the Max-product Belief propagation algorithm in Arbitrary graphs,” IEEE Transactions on Information Theory, 47(2),723–735, 2001

[8] J. Yedidia, W. Freeman and Y. Weiss, “Generalized Belief Propagation,”Advances in Neural Information Processing Systems, 13, 689-695, 2001.

[9] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(6), 721–741, 1984.

[10] Y. Boykov, O. Veksler and R. Zabih, “Fast Approximate Energy Minimization via Graph cuts,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11),1222–1239, 2001.

[11] J. Sun, N. Zheng and H. Shum, “Stereo Matching using Belief propagation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(7),787–800, 2003.

[12] O. Veksler, “Efficient Graph-Based Energy Minimization Methods in Computer Vision” in PhD thesis, Cornell University, 1998.

[13] T. Kanade and M. Okutomi. “A stereo matching algorithm with an adaptive window: Theory and experiment,” IEEE Trans. PAMI, 16(9):920–932, 1994.

[14] L. Zitnick and T. Kanade, “A cooperative algorithm for stereo matching and occlusion detection,” IEEE Trans. PAMI, 22(7),675–684, 2001.

[15] D. Scharstein and R. Szeliski, “A taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” International Journal of Computer Vision, 47, 1, 7-42, 2002

[16] Y. Weiss and W. Freeman, “On the Optimality of Solutions of the Max-product Belief propagation algorithm in Arbitrary graphs,” IEEE Transactions on Information Theory, 47(2),723–735, 2001.

[17] O. Devillers and M. Golin, “Incremental Algorithms for finding the Convex hulls of circles and the Lower envelopes of Parabolas,” Inform. Process. Lett., 56(3),157–164, 1995.

[18] L. Valgaerts, A. Bruhn, M. Mainberger, M. and J. Weickert. “Dense versus Sparse Approaches for Estimating the Fundamental Matrix”, International Journal of Computer Vision, 96(2), 212-234, 2012.

[19] T. Dickscheid, F. Schindler and W. Gorstner, “Coding Images with Local Features,” International Journal of Computer Vision 94(2) 154-174, 2011.