Maximum-Likelihood Stereo Correspondence using Field Programmable Gate Arrays

Siraj Sabihuddin & W. James MacLean

Department of Electrical and Computer Engineering, University of Toronto,
Toronto, Ontario, Canada,
{siraj|maclean}@eecg.utoronto.ca

Abstract. Estimation of depth within an imaged scene can be formulated as a stereo correspondence problem. Typical software approaches tend to be too slow for real time performance on high frame rate ($\geq$ 30fps) stereo acquisition systems. Hardware implementations of these same algorithms allow for parallelization, providing a marked improvement in performance. This paper will explore one such hardware implementation of a maximum-likelihood stereo correspondence algorithm on a Field Programmable Gate Array (FPGA). The proposed "FastTrack" hardware implementation is a first stage prototype that demonstrates comparable results to equivalent software implementations. Future optimizations will have the added advantage of high-speed (up to 200fps) and motion-compensated stereo depth estimation.

1 Introduction

Stereo vision makes use of two images from different view points to construct a three dimensional representation of the world. This is a particularly valuable representation as it allows depth estimation of scene points via stereo correspondence. The problem of stereo correspondence has been extensively researched by the computer vision community.

Attempts at achieving accurate correspondences can be roughly categorized into two main classes: sparse stereo and dense stereo correspondence [7]. Representations of these correspondences generally take the form of disparity maps, each map storing separating distances between matching pixels of the stereo images.

Software implementations running on general purpose personal computers (PCs) work fine when computing stereo depth estimates for low frame rate stereo image acquisition systems. For faster systems ($\geq$ 30fps) these same software implementations may be trans-

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

lated to customized hardware. Platforms based on Field Programmable Gate Arrays (FPGAs) are popular choices for such hardware. Several such FPGA-based solutions to the stereo correspondence problem have been explored by the vision community in the past, [2, 3, 5, 6] are some examples.

This paper will explore another hardware-based approach that makes use of a maximum-likelihood stereo depth estimation algorithm to compute dense disparity maps on FPGAs for high frame rate stereo correspondence. The proposed implementation, referred to as "FastTrack", is a first stage prototype that produces comparable results to existing software implementations. Future optimizations of this prototype are aimed at achieving high frame rate (up to 200fps) motion-compensated stereo estimation.

2 Maximum-Likelihood Stereo Vision

Dense stereo correspondence may be performed using a maximum-likelihood formulation. One such formulation has been presented by Cox [1], who argued that a Dynamic Programming based Maximum-Likelihood (DPML) approach applied to pixel intensities can produce good stereo depth estimates.

Given two images, left and right, from a binocular stereo camera, we note that most pixels in the left camera image can be matched to a corresponding pixel in the right. Under ordinary conditions, such correspondence requires a search over an entire image for each pixel disparity estimate. Rectification reduces the search space to a region along an epipolar line [4].

The left-right pixel pair that maximize the likelihood metric represent a stereo correspondence that may be used to compute the disparity. However, a correspondence may not be possible for all pixels due to occlusions. Appropriately, a fixed occlusion penalty is often used as a threshold for a positive correspondence.

Maximizing the likelihood is equivalent to minimizing a cost function. The cost represents the degree of difference between two pixels from the stereo image pair and is defined by Equation (1), with its associated occlusion penalty, Equation (2).

$$NOC(I_l(x), I_r(x+d), \sigma^2) = \frac{(I_l(x) - I_r(x+d))^2 \sigma^2}{4} \qquad (1)$$

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

$$OC(P_d, \sigma^2, \phi) = \log \frac{P_d \phi}{(1 - P_d)\sqrt{\frac{2\pi}{\sigma^2}}} \quad (2)$$

$I_l(x)$ and $I_r(x + d)$ represent pixel intensities at positions x and $x + d$ along the scanlines of the left and right images respectively. The disparity, $d = 0 \ldots D - 1$, indicates the offset corresponding to the search region (D) in the right image. σ^2 represents the variance associated with camera sensor noise. P_d is the probability of each camera imaging a point in the scene, while ϕ is the associated field of view. Generally the values P_d, ϕ, and σ^2 may be fixed since they model the physical properties of the imaging system and scene.

The use of the comparison metric does not guarantee the best choice for matching pairs. In order to produce more accurate correspondences and to constrain the search space further, two assumptions are commonly made about pixels within a stereo image pair:

1. *The Uniqueness Assumption:* Which states that each pixel in the left image can be matched to no more than one pixel in the right.
2. *The Monotonic Ordering Assumption:* Which states that every pixel in the left image should match a pixel in the right in a monotonic ordering.

2.1 The Algorithm

The two assumptions, along with the cost equations mentioned earlier, lead to a dynamic programming solution to the stereo correspondence problem. Listings 1 and 2 describe the algorithm associated with this DPML solution. The variable NOC is computed for each pixel in a scanline (along an epipolar line) using Eq. (1). The cost matrix, C, stores the cost comparisons between pixels in the left and right scanlines over a search region D. The indices, cl and cr, store the current pixel positions for the left and right scanlines respectively. The match matrix, M, stores indicators for occlusions or non-occlusions. Both M and C are NxN matrices, where N is the number of pixels in a single image scanline of the stereo pair. Once computed, an optimal path is determined through the match matrix such that correspondence and occlusion results adhere to uniqueness and ordering assumptions.

3 Hardware Design

Software implementations of stereo correspondence algorithms generally operate at acceptable speeds for low frame rate stereo image

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

```
% Initialize match and cost matrices
for c=1:N
    M(c,1:c)=1; M(c,(c+1):N)=2;
    C(c:N,c)=(((c-1):(N-1))*OC);
    C(c,c:N)=(((c-1):(N-1))*OC);

% For each pixel in a row compute NOC, OC
for cl=2:N
    for cr = cl:(cl-D)
        min1 = C(cr-1, cl-1) + NOC;
        min2 = C(cr-1, cl) + OC;
        min3 = C(cr, cl-1) + OC;
        C(cr,cl) = min(min1, min2, min3) = cmin;

        if (min1==cmin)      M(cr,cl)=0; % No occlusion
        elseif (min2==cmin) M(cr,cl)=1; % Right occlusion
        elseif (min3==cmin) M(cr,cl)=2; % Left occlusion
```

Listing 1: The Cox DPML algorithm, first phase, in Matlab-like notation. Cost and match computations for each pixel location in a scanline, over a disparity range, D.

acquisition. At high resolutions and frame rates ($\geq$30 fps), the general purpose hardware and associated software overhead make it difficult, if not impossible, for these implementations to achieve real time performance. FPGAs provide a robust platform for development of hardware implementations of these algorithms for high frame rate camera systems. This section will discuss the design of FastTrack, an FPGA-based prototype aimed at eventually achieving high frame rate stereo correspondence using the Cox DPML algorithm.

3.1 Algorithm Modifications

Typical FPGA development systems have resource limitations not encountered on general purpose PCs. These limitations necessitate the adaptation of stereo algorithms to minimize logic and memory usage. The DPML algorithm presented in Listings 1 and 2 can be modified to reduce this resource usage as shown in Listing 3, the FastTrack DPML algorithm.

We note that we need only represent two rows of the cost matrix at a time, since once the corresponding match matrix elements have been, the cost values are no longer needed. Cost matrix memory utilization is thus decreased from N^2 elements to $2N$ elements in this algorithm. The reduction is significant considering that cost results may take as much as 16 bits per memory element. Since the

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

```
p = N; q = N;
while (p≠1 && q≠1)
    if M(p,q)==0
        DISP(q)=abs(p−q);
        OCC(q)=0;
        p−−; q−−;
    elseif M(p,q)==1
        OCC(q)=1; p−−;
    elseif M(p,q)==2
        OCC(q)=1; q−−;
```

Listing 2: The Cox DPML algorithm, second phase. Computation of optimal disparity values for pixels, in a scanline, relative to the left image of the stereo pair. Disparity/occlusion computations makes use of the cost/match computations in Listing 1.

N^2 element match matrix only stores three distinct values (2 bits per element), its memory usage remains reasonable. Intermediate computations can be approximated by integer operations to reduce logic and memory usage further. Both the cost and match matrix sizes can be decreased even more by taking into account a maximum disparity range, $d_{max} = max(D)$.

3.2 Architectural Overview

The FastTrack DPML algorithm (Listings 2 and 3) may be translated to a high level hardware architecture (see Figure 1). Optimizations to this architecture are discussed in the next section.

Stereo image data is stored into off-chip memory by the camera system, and subsequently retrieved by the FastTrack hardware and stored in the Image Scanline Memory (ISM). The data is processed and results communicated to an external module via a Data Transfer Unit (DTU). Iterators in the ISM run through and compare left and right scanline pixels by computing associated costs through the Compute NOC (CNOC) module. The iterator addresses are resolved (via the Resolve Address & Data (RAD) unit) into indexing addresses for two dual port RAMs. These RAMs form a partial Cost Matrix Memory (CMM) of size: 16 bits by $2N$ and store the intermediate minimum cost computations for the current ISM pixel addresses. The minimum cost value with and without occlusion penalty, is associated with an index value between 0 and 2 and stored in a 2 bit by N^2 Match Matrix Memory (MMM). Once match values have been computed for all N^2 elements of the MMM, a second set of iterators

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

```
% Initialize match and cost matrices
M(1:N,1:N) = 1;
mem(1,1)=0; mem(2,1)=OC; mem(1,2)=OC;
first_cr_flg=false;

% For each pixel in a row compute NOC, OC
for cl = 2:N
    for cr = cl:(cl-D)
        current_mem = mod(cl-1,2)+1;
        if (current_mem == 1) previous_mem = 2;
        else previous_mem = 1;

        if (first_cr_flg==true)
            mem(current_mem,cr-1)=(cl-1)*OC;
            first_cr_flg=false;

        C_cr_1_cl_1 = mem(previous_mem,cr-1);
        C_cr_1_cl = mem(current_mem,cr-1);
        C_cr_cl_1 = mem(previous_mem,cr);
        min1 = C_cr_1_cl_1 + NOC;
        min2 = C_cr_1_cl + OC;
        min3 = C_cr_cl_1 + OC;
        C_cr_cl = min([min1, min2, min3]);
        mem(current_mem,cr)=C_cr_cl;
        if (cr==cl)
            mem(current_mem,i+1)=cr*OC;
            first_cr_flg=true;

        if (min1==C_cr_cl) M(cr,cl)=0;
        elseif (min2==C_cr_cl) M(cr,cl)=1;
        elseif (min3==C_cr_cl) M(cr,cl)=2;
```

Listing 3: The FastTrack DPML algorithm used for hardware implementation. Optimal disparities are computed as shown in Listing 2.

in the Compute Disparity (CD) module run through an optimum matching pixel pathway. At each iteration, the optimum pathway is used to compute the disparity and occlusion values of the particular pixel in the scanline. Data processing occurs sequentially, searching exhaustively over all pixels in the image scanline. This sequential processing is controlled by a centralized state machine.

3.3 Architectural Improvements and Timing Analysis

The standard FastTrack architecture (Figure 1), alone, cannot perform high frame rate stereo correspondence. However, certain opti-

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

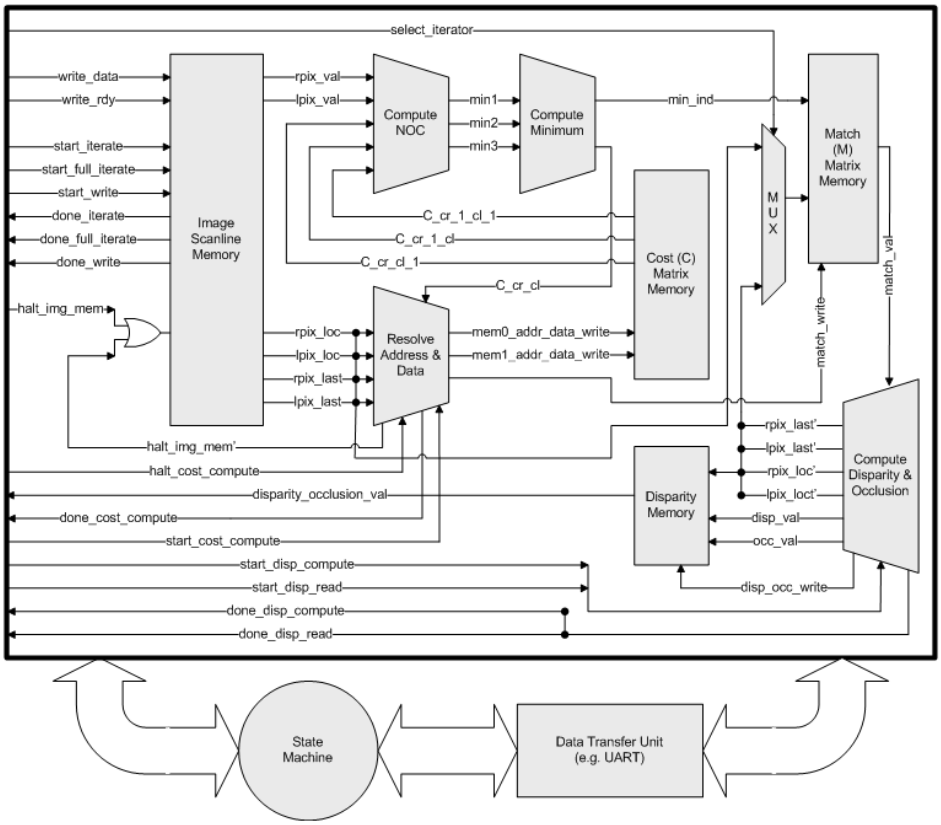


Fig. 1. High level architecture for the DPML-based FastTrack stereo correspondence hardware.

mizations can improve timing characteristics enough to achieve high frame rates (i.e. $\geq$ 30fps). These optimizations are listed below:

1. *Pipelining:* Increases clock frequency (f_{clk}) by reducing combinational path delays. These delays are particularly long through the CNOC, CM, RAD path of the design. These delays can be reduced from 21ns to 10ns.
2. *Parallelization:* Decreases processing by computing disparities for multiple scanlines simultaneously. The use of two copies ($P = 2$) of the FastTrack architecture in Figure 1, for example, cuts the processing time of a stereo pair in half. Better still, pixel rates can be achieved if all pixel-wise comparisons are done in parallel as scanline pixels arrive.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

3. *Reduced Pixel by Pixel Sequential Processing:* Increases processing by limiting correspondence search to a maximum disparity, d_{max}. Fewer memory transfers and cost computations are required. Additional improvements can be made by limiting the number of clock cycles (rw_{clk}) required, to read previous and write current cost computations.

Timing characteristics can be analyzed to determine frame rates for stereo correspondence results. Equations 3 and 4 provide a generalized model for computing these rates, with no optimization and with optimization respectively. The value M represents the number of scanlines, while the value N represents the number of pixels per scanline. d_{max}, f_{clk} and rw_{clk} refer to the maximum disparity, clock frequency, and RAM read/write clock cycles (per cost computation). The variable P indicates the level of parallelization.

$$R_{noop}(M, N, f_{clk}) = \frac{f_{clk}}{M(4N + N^2)} \tag{3}$$

$$R_{op}(M, N, P, f_{clk}, d_{max}, rw_{clk}) = \frac{P f_{clk}}{M(5N + rw_{clk} d_{max} N)} \tag{4}$$

Algorithmic and architectural changes, that incorporate motion based constraints for depth estimation, can provide additional improvements in timing and accuracy.

4 Results

FastTrack stereo correspondence frame rates are compared to those of the original Cox algorithm [1] and MATLAB software equivalents. Frame rates for the hardware optimizations, discussed earlier, are computed using equation (4) with parameters: $P = 2$, $rw_{clk} = 1$, $d_{max} = 16$, $f_{clk} = 100Mhz$. The results can be found in Table 1.

Figure 2 and Table 2 provide a comparison of the hardware and software stereo correspondence results on the standardized Tsukuba and Venus image sequences. Comparisons were made to SSD, correlation, and DPML software implementations. These results clearly demonstrate that FastTrack hardware provides superior stereo estimation to SSD and correlation. More exhaustive results comparing stereo algorithms have been compiled by Scharstein *et al.*in [7].

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Algorithm	f_{clk}	**Img. Resolution**	**Frame Rate (R)**
FastTrack (un-optimized)	47 Mhz	384 x 288	1.10fps
FastTrack (un-optimized)	47 Mhz	512 x 512	0.35fps
FastTrack (optimized)	100 Mhz	384 x 288	86.12fps
FastTrack (optimized)	100 Mhz	512 x 512	36.33fps
DPML MATLAB software	3.70 GHz	384 x 288	$7.69\text{x}10^{-3}$fps
DPML Cox software	35 Mhz	512 x 512	0.03fps
DPML Cox software	3.70 Ghz	512 x 512	3.15fps

Table 1. A comparison of frame rates for hardware/software implementations of the DPML algorithm

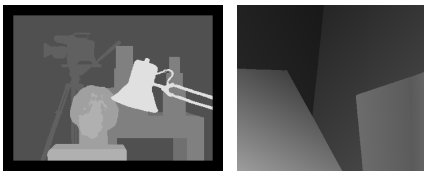
Algorithm	**RMS Error**		**% Bad Pixel Match**	
	Tsukuba	Venus	Tsukuba	Venus
FastTrack	0.9431	2.5345	5.03%	13.17%
DPML Cox *et al.*	0.9899	2.6662	5.33%	13.58%
Correlation	1.8015	3.7353	10.28%	20.71%
SSD	3.6852	6.4253	29.10%	48.50%

Table 2. A comparison of stereo correspondance results for common hardware stereo algorithms. Comparison is to ground truth disparities using Root Mean Squared (RMS) error and % pixels incorrectly matched.

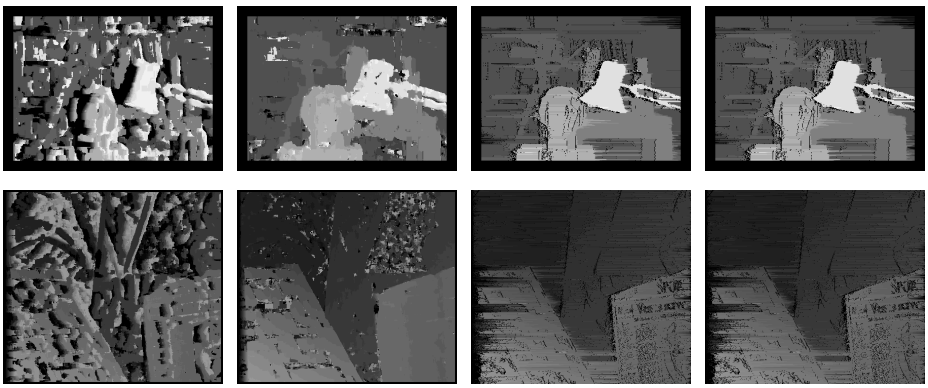
5 Conclusion

The FastTrack hardware implementation of the Cox *et al.*[1] dynamic programming stereo correspondence algorithm (DPML) is a first stage prototype that can achieve results comparable to other algorithms. Future improvements aim to yield higher frame rate depth estimation by optimizing timing characteristics. The single most important lesson learned is that left-right pixel comparisons must happen in parallel as pixel arrive to achieve the best processing rates. Combined with motion based constraints, these optimizations hold the promise of accurate and high speed (up to 200fps) stereo depth estimation.

The authors would like to acknowledge financial support from Ontario Centres of Excellence, and equipment provided by CMC Microsystems.



(a) Ground Truth



(b) SSD (c) Correlation (d) DPML (e) Fasttrack

Fig. 2. Stereo correspondence results for the Tsukuba and Venus image sequences. (a) Ground Truth, Left:Tsukuba, Right:Venus, (b) Sum of Squared Differences (SSD), 9x9 window, (c) Correlation, 9x9 window, (d) Dynamic Programming-Maximum Likelihood (DPML), (e) FastTrack DPML

References

1. Ingemar J. Cox, Sunita L. Hingorani, Satish B. Rao, and Bruce M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.
2. Ahmad Darabiha, Jonathan Rose, and W. James MacLean. Video-rate stereo depth measurement on programmable hardware. *CVPR*, 2003.
3. M. Hariyama, Y. Kobayashi, H. Sasaki, and M. Kameyama. FPGA implementation of a stereo matching processor based on window-parallel-and-pixel-parallel architecture. 2005.
4. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
5. R. P. Jacobi, R. B. Cardoso, and G. A. Borges. Voc: a reconfigurable matrix for stereo vision processing. USA, 2006. Dept. of Comput. Sci., Brasilia Univ., Brazil.
6. D. K. Masrani and W. J. MacLean. A real-time large disparity range stereo-system using FPGAs. Dept. of Electr. Comput. Eng., Toronto Univ., Ont., Canada, 2006.
7. Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8