# A ROBUST MESSAGE PASSING BASED STEREO MATCHING KERNEL VIA SYSTEM-LEVEL ERROR RESILIENCY

*Eric P. Kim*[⋆]*, Jungwook Choi*[⋆]*, Naresh R. Shanbhag*[⋆]*, and Rob A. Rutenbar*[†]

[⋆]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
[†]Department of Computer Science, University of Illinois at Urbana-Champaign

## ABSTRACT

In this paper, we present an error resilient Markov random field (MRF) message passing based stereo matching hardware (HW) architecture. Previously, algorithmic noise tolerance (ANT) has been applied at the arithmetic level of the reparameterize unit and showed greatly enhanced robustness of message passing inference based architectures. In this work, correction was targeted at the system level to reduce correction overhead while maintaining performance. An erroneous FPGA based accelerator was employed as our emulation platform. Through relaxed synthesis, we show that timing errors occur within the message passing unit, and are successfully compensated. Error correction has been implemented at several hierarchical levels, including end of iteration, and the final depth map output. Significant enhancement in robustness is achieved with minimal correction overhead. Compared to HW error compensation at the arithmetic level, system level error compensation reduces overhead by more than $50\,\%$, while maintaining stereo matching performance with only $3.8\,\%$ degradation.

***Index Terms***— error-resiliency, FPGA, timing errors, message passing, stereo matching

## 1. INTRODUCTION

Non-idealities due to statistical parameter variations including process, temperature and voltage variations will be unavoidable in the future [1]. Present day approaches try to avoid these errors via over-designing and are often wasteful and unaffordable in many power-limited applications. Alternatively, statistical techniques have been developed to improve the performance of signal processing systems [2, 3] and also have been applied to tolerate errors in hardware [4].

Recently, machine learning based inference is gaining importance as a key kernel in processing massive data in signal processing systems including computer vision and speech recognition. Such applications contain an inference kernel which is inherently resilient to small magnitude errors [5, 6]. By combining the statistical performance metric of machine learning applications and statistical nature of circuit non-idealities, statistical error compensation (SEC) techniques [7] can achieve significant enhancement in error resiliency [5, 6]. By trading off the increased robustness with energy, significant energy savings can be achieved as well. In [5], algorithmic noise tolerance (ANT) has been applied to a message passing based low density parity check (LDPC) decoder and shown to achieve $45.7\,\%$ energy savings while maintaining less than $4.7\,\mathrm{dB}$ degradation in bit error rate (BER) at a HW error rate (percentage of clock cycles in which an erroneous output exists) of 30%. In [6], an architecture implementing a sequential tree-reweighted (TRW-S) Markov random field (MRF) message passing based stereo matching [8] has shown to achieve similar results with energy savings of $41\,\%$ with a bad pixel ratio (BPR) degradation of less than $2.96\,\%$ at a HW error rate of $21.3\,\%$. However, in [5, 6], ANT was applied at the arithmetic level incurring large overhead.

In this paper, we apply SEC at the system level for the TRW-S stereo matching system to reduce compensation complexity while maintaining performance. The CPU+FPGA platform [8] is used to verify the results in a system-on-chip (SoC) setting. Real hardware errors were induced in the reparameterize unit (see Section 3) of the TRW-S architecture through relaxed synthesis. A study on the tradeoff between compensation overhead and error correction capabilities was performed by implementing several estimation approaches at different levels. At the system level, a CPU was used to aid in the estimation and correction of the final stereo matching output (i.e., depth map). This approach may seem similar to the existing ERSA [9] paradigm, but is significantly different because: a) an application specific accelerator is used and permitted to make hardware errors, instead of a general purpose relaxed redundancy core (RRC) used in [9], and b) in accelerators, errors are generated in the data path rather than the control path, and ANT provides extremely low overhead compensation with minimal performance degradation.

The remainder of the paper is organized as follows. Section 2 provides background information (ANT and TRW-S stereo matching). Implementation of TRW-S and HW error generation and compensation is discussed in Section 3. Section 4 shows the results while Section 5 concludes the paper.

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

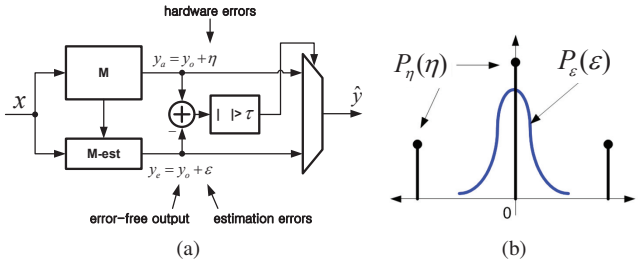


**Fig. 1**: Block diagram: (a) algorithmic noise tolerance, and (b) error distributions.

## 2. BACKGROUND

### 2.1. Algorithmic noise tolerance

Algorithmic noise tolerance (ANT) [10, 5] is one method of statistical error compensation that utilizes the statistics of errors to perform detection and estimation to compensate for errors. It also incorporates system level statistical metrics, such as signal-to-noise ratio (SNR) or bit error rate (BER). As shown in Fig. 1(a), ANT incorporates a *main* block and an *estimator*. The main block is permitted to make hardware/timing errors, but not the estimator. The estimator is a low-complexity block (typically $5\%$-to-$20\%$ of the main block complexity) generating a statistical estimate of the correct main block output, i.e.,

$$y_a = y_o + \eta \quad (1)$$
$$y_e = y_o + e \quad (2)$$

where $y_a$ is the actual main block output, $y_o$ is the error-free main block output, $\eta$ is the hardware error, $y_e$ is the estimator output, and $e$ is the estimation error. The final output of an ANT system $\hat{y}$ is obtained via the following decision rule:

$$\hat{y} = \begin{cases} y_a, & \text{if } |y_a - y_e| < \tau \\ y_e, & \text{otherwise} \end{cases} \quad (3)$$

where $\tau$ is an application-dependent parameter chosen to maximize the performance of ANT.

### 2.2. TRW-S message passing based stereo matching

Stereo matching infers depth information given stereo images. It can be restated as a maximum a posteriori (MAP) problem, where each pixel is given a label that corresponds to a discrete depth level and the goal is to find the most probable label assignments for all the pixels in the image. This MAP problem can be formulated in terms of cost functions defined on an undirected grid graph (i.e., a grid MRF) with nodes $(V)$ and edges $(E)$ as follows [11, 12]:

$$\min_{\boldsymbol{l}} \mathcal{E}(\boldsymbol{l}) = \min_{\boldsymbol{l}} \left\{ \sum_{s \in V} d_s(l_s) + \sum_{(s,t) \in E} V_{st}(l_s, l_t) \right\} \quad (4)$$

A unary cost function $d_s(l_s)$ represents the likelihood of a depth label $l_s$ being assigned to a node $s$, and a pairwise (smoothness) cost function $V_{st}(l_s, l_t)$ models prior preference of smooth label assignment among neighboring nodes. $\mathcal{E}(\boldsymbol{l})$ is referred to as *energy*, and thus, (4) finds the label assignments $\{l_s\}$ which minimize energy $\mathcal{E}(\boldsymbol{l})$; a lower energy corresponds to a higher quality label assignment.

In general, this energy minimization problem is intractable as the complexity of the problem grows exponentially with the size of the graph. However, message passing algorithms such as sequential tree-reweighted message passing (TRW-S) [13] have been shown to efficiently solve (4) by selecting the best label for each node based on local information from its neighbors (i.e. messages). Messages between nodes of an MRF are updated iteratively via two-step message passing (reparameterize and update message [13]) until they converge to a fixed value. In each message passing, previous messages from neighbors of a node are aggregated (reparameterize) to be used along with the smoothness cost for computation of new messages (update message). The final label is then assigned based on the converged messages. In practice, TRW-S has shown superior performance thanks to its favorable message convergence property [13].

## 3. SYSTEM ARCHITECTURE

A TRW-S based stereo matching system has been implemented in a hybrid CPU+FPGA platform to perform high quality stereo matching in video-rate speed [8]. Figure 2 shows the overall architecture of TRW-S stereo matching system. The platform (Convey HC-1 [14]) contains an Intel Xeon dual core processor and four Virtex 5 (V5LX330) Xilinx FPGAs, with a cache-coherent virtual memory system across both multicore and FPGA fabrics. Two-step (reparameterize, REPARAM, and update message, UPDMSG) message passing algorithm is accelerated in an FPGA, where its data path is fully pipelined and MRF data (unary/smoothness cost) is streamed via FIFOs to achieve high throughput; the message passing unit (MPU) starts and retires a complete pixel and all its message computations in each clock cycle. The CPU not only controls FPGA operations but also processes image input and stereo matching output (i.e. depth map). Through the utilization of both CPU and FPGAs, video-rate stereo matching is achieved [8].

In the past [6], to further enhance error resiliency, ANT was applied to the message computation unit of the TRW-S stereo matching system. Each basic computation is protected by reduced precision replica (RPR) based ANT to tolerate timing induced arithmetic errors occurring in the message passing hardware at low supply voltage ($V_{dd}$). However, this fine grain protection causes high overhead, reducing benefits of voltage over-scaling. In this work, we reduce the overhead of ANT at the system level by utilizing the computational power of the CPU.

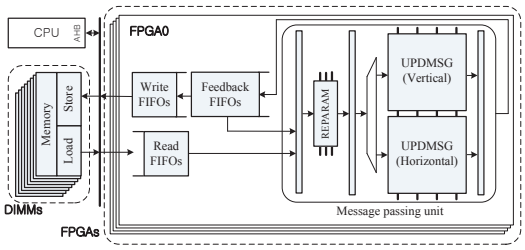


**Fig. 2**: Architecture of streaming TRW-S stereo matching CPU+FPGA system.

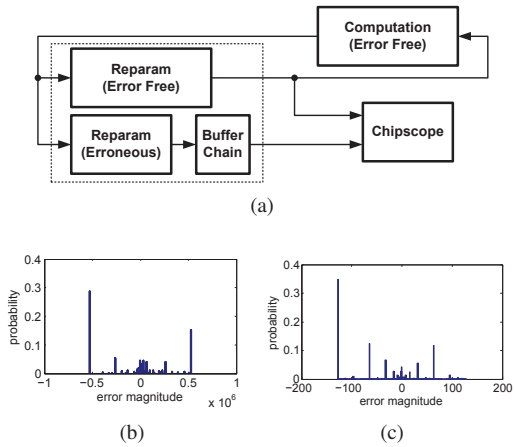


**Fig. 3**: Timing errors in FPGA: (a) block diagram for error verification and statistics collection, (b) measured error statistics (20-bit), and (c) simulated error statistics (8-bit) [6].

### 3.1. Error generation

To evaluate the results under real HW errors, the message passing unit is implemented as an FPGA accelerator exhibiting timing violations via relaxed synthesis. At a target frequency of 150 MHz, all the paths through the REPARAM block were set as false paths. Xilinx's ChipScope Pro was used to verify that the errors were generated within REPARAM, and compared against error free results to obtain the error statistics (Fig. 3(a)). Figure 3(b) shows similar error statistics to the ones obtained through HSPICE simulations (Fig. 3(c)) based on the modeling methodology in [6].

However, controlling the error rate $p_e$ in an FPGA system is known to be difficult due to various synthesis/mapping constraints and unpredictable routing within the FPGA [15]. The buffer chain in Fig. 3(a) is such an example. Contrary to our expectations, $p_e$ and the resulting matching performance did not have strong dependency with the length of the buffer chain. To enable flexible control of $p_e$ we extracted the error statistics of the FPGA (Fig. 3(b)) and used error injection in a simulator. The results of the simulator are comparable with the FPGA emulation results (Sec 4.1), justifying this approach.

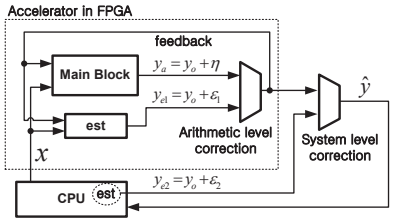


**Fig. 4**: Block diagram of system level error compensation.

### 3.2. Error compensation

Errors are compensated via ANT at two levels as illustrated in Fig. 4. At the system level, errors are compensated directly on the resulting depth map, while for the iteration level, errors are compensated once per iteration, in an online fashion (compensated results are directly used as updates for the adjacent nodes). A hybrid scheme that employs compensation at both levels can enable further reduction in complexity of the estimator. The erroneous TRW-S architecture (Sec. 2.2-2.3) is used as the main block (20-bit precision), while different estimators are used for each level of compensation.

1) System level compensation: the Intel Xeon processor available within the convey HC-1 platform is used to compute estimates of the depth map. We have employed a sum of absolute difference (SAD) based stereo matching algorithm [16] and a scaled version (scaled by a factor of $S$) of TRW-S based on hierarchical belief propagation (HBP) [17] as estimators. The SAD based stereo matching is one of the simplest matching algorithms available while the scaled version of TRW-S utilizes a crude graph for significant reduction in complexity. This compensation method can be easily extended to system-on-chips (SoC) as such systems make extensive use of HW accelerators combined with a general purpose processor.

2) Iteration level compensation: compensation occurs within the HW architecture. Errors are compensated once per iteration in an online fashion. For correction within the message passing loop, a reduced precision replica (RPR) of the TRW-S architecture is used. With sufficient precision of the estimator (at least 8-bit estimator with a 20-bit main block), significant error resiliency is obtained, but at the cost of high compensation complexity. Thus, in this work, we use a hardware estimator with lower precision (4-bits) for reduced overhead while maintaining error compensation capability through system level error compensation.

## 4. RESULTS

The Tsukuba image from the Middlebury testbench [18] is used as our test image. First, the correctness of our simulator is verified by comparing the results against FPGA emulation. Then, error statistics obtained from the FPGA emulation (Fig. 3(b)) are used to inject errors in the simulator to show the performance of system and iteration level error compensation.

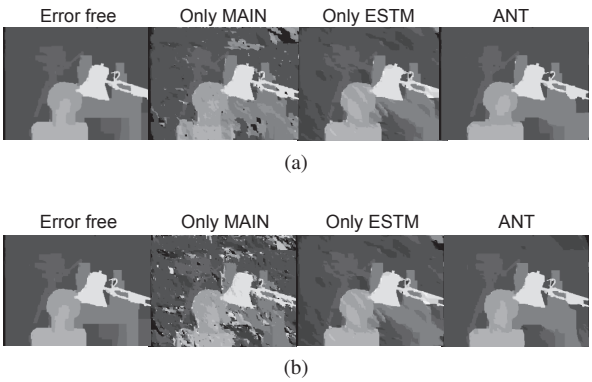


**Fig. 5**: Results for RPR-ANT applied at the arithmetic level: (a) FPGA emulation, and (b) error injection based simulation.

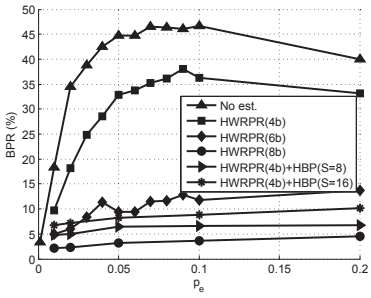


**Fig. 6**: Bad pixel ratio vs. $p_e$. With no error compensation, TRW-S cannot tolerate more than 1% errors.

### 4.1. Comparison of FPGA emulation and simulation

FPGA emulation results of ANT applied at the arithmetic level for Tsukuba is shown in Fig. 5. The resulting performance is very similar, and we conclude that the simulator faithfully represents the FPGA emulation results.

### 4.2. Simulation results

Simulation enables us to explore injection of the same error statistics, but at different error rate $p_e$. Figure 6 summarizes the error compensation performance of system level and iteration level compensation. BPR was used as our performance metric. A RPR based HW estimator (HWRPR) was used for iteration based compensation with a precision of 4-bit, 6-bit, and 8-bit, while SAD of window size 7 ($W = 7$), and for HBP, a scaled down graph of $S = 8$ and $S = 16$ was used for the estimator at the system level. As expected, HBP with low precision HWRPR shows comparable error resiliency to high precision (and large overhead) HWRPR.

SAD based correction at the system level with $p_e = 0.05$ is shown in Fig. 7(a). At this error rate, system level correction is insufficient for correct operation (main block BPR is 44.8%). Using HBP (Fig. 7(b)) alone, the final output is rather worse than the estimator. Further combining with

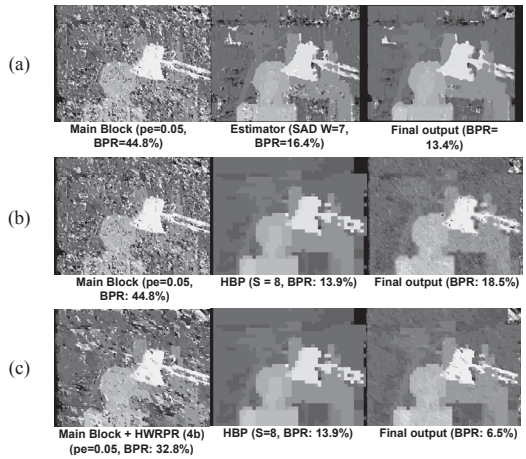


**Fig. 7**: Correction performance at: (a) system level with SAD, (b) system level with HBP ($S = 8$), and (c) hybrid with HBP ($S = 8$) and HWRPR (4-bit).

**Table 1**: Summary of LUT and register overhead of HWRPR.

| | MPU | | Main | | HWRPR | | Overhead | |
|---|---|---|---|---|---|---|---|---|
| | Reg | LUT | Reg | LUT | Reg | LUT | Reg | LUT |
| 4b | 27048 | 26956 | 2016 | 2992 | 1554 | 591 | 77% | 20% |
| 8b | 27240 | 27116 | 2016 | 2992 | 1746 | 751 | 87% | 43% |

HWRPR, BPR of $6.5\,\%$ is achievable with a 4-bit HWRPR estimator (Fig. 7(c)), which is a 3.8% degradation compared to the error free BPR (2.7%).

### 4.3. Correction Overhead

The overhead of HWRPR for iteration based compensation is derived through the FPGA MAP report of the synthesized blocks. A summary of the overhead compared to the message passing unit (MPU) is given in Table 1, for a 4-bit and 8-bit estimator. It can be seen that a 4-bit estimator has $13\,\%$ less overhead in registers and $50\,\%$ less overhead in LUTs compared to the 8-bit estimator. For system level compensation (HBP), the complexity depends on the number of nodes in the graph, which is $\frac{1}{64}$ for $S = 8$, and $\frac{1}{256}$ for $S = 16$. The execution time for the HBP estimator on a Intel Xeon @ 2.13GHz with ten iterations was approximately $250\,\mathrm{ms}$ and $60\,\mathrm{ms}$ for $S = 8$ and 16, respectively, while the main block took approximately $130\,\mathrm{ms}$ for forty iterations.

## 5. CONCLUSION

System level correction reduces correction complexity by a significant amount, but is inadequate. When combined with iteration level compensation, performance gains are significant with little increase in complexity.

## 6. REFERENCES


[1] "International Technology Roadmap for Semiconductors," Online: http://www.itrs.net.

[2] Y. Lee and Y. Altunbasak, "A collaborative multiple description transform coding and statistical error concealment method for error resilient video streaming over noisy channels," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, vol. 2, 2002, pp. II–2077.

[3] A. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, vol. 4, 2007, pp. IV–1229.

[4] J. Choi, B. Shim, A. Singer, and N. Cho, "Low-power filtering via minimum power soft error cancellation," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5084–5096, 2007.

[5] E. P. Kim and N. R. Shanbhag, "Energy-efficient LDPC decoders based on error-resiliency," in *IEEE Workshop on Signal Process. Syst. (SiPS)*, 2012, pp. 149–154.

[6] J. Choi, E. P. Kim, R. A. Rutenbar, and N. R. Shanbhag, "Error resilient MRF message passing architecture for stereo matching," in *IEEE Workshop on Signal Process. Syst. (SiPS)*, 2013.

[7] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *Proc. 47th Design Automation Conf. (DAC)*, 2010, pp. 859–864.

[8] J. Choi and R. A. Rutenbar, "Video-rate stereo matching using markov random field TRW-S inference on a hybrid CPU+FPGA computing platform," in *Proc. ACM/SIGDA Int. Symp. on Field programmable gate arrays (FPGA)*, 2013, pp. 63–72.

[9] L. Leem, H. Cho, J. Bau, Q. Jacobson, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications," in *Design, Automation, and Test in Europe Conference and Exhibition (DATE)*, 2010, pp. 1560–1565.

[10] N. R. Shanbhag, "Reliable and efficient system-on-a-chip design," *IEEE Computer*, vol. 37, no. 3, pp. 42–50, Mar. 2004.

[11] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, 2003.

[12] M. Tappen and W. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *IEEE Int. Conf. on Comp. Vision*, 2003, pp. 900–906.

[13] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.

[14] Convey Computer, "Convey Reference Manual," Online: http://www.conveycomputer.com, Sep. 2009.

[15] S. Morozov, A. Maiti, and P. Schaumont, "An analysis of delay based PUF implementations on FPGA," in *Reconfigurable Computing: Architectures, Tools and Applications*, ser. Lecture Notes in Computer Science, P. Sirisuk, F. Morgan, T. El-Ghazawi, and H. Amano, Eds. Springer Berlin Heidelberg, 2010, vol. 5992, pp. 382–387.

[16] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *IEEE Conf. on Comp. Vision and Pattern Recognition*, 2007, pp. 1–8.

[17] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41–54, 2006.

[18] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.