https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-114

DRAM Bandwidth Optimal Perspective Transform Engine

Mihir Mody, Rajasekhar Allu*, Gang Hua+, Brijesh Jadav, Niraj Nandan*, Ankur, Mayank Mangla*
Embedded Processor Business, Texas Instruments
Bangalore- India, *Dallas & +Houston- USA

Abstract

Perspective transform (or Homography) is commonly used algorithms in ADAS and Automated Driving System. Perspective transform is used in multiple use-cases e.g. viewpoint change, fish-eye lens distortion correction, chromatic aberration correction, stereo image pair rectification, This algorithm needs high external DRAM memory bandwidth due to inherent scaling, resulting in non-aligned two dimensional memory burst accesses, resulting in large degradation in system performance and latencies. In this paper, we propose a novel perspective transform engine to reduce external memory DRAM bandwidth to alleviate this problem. The proposed solution consists of multiple regions slicing of input video frame with block size tuned for each region. The paper also gives an algorithm for finding optimal region boundaries with corresponding block size tuned for each region. The proposed solution enables average BW reduction of 67% compared to traditional implementation and achieves clock up-to 720 MHz with output pixel throughput of 1 cycle/pixel in 16nm FinFET process node.

I. Introduction

In this paper, we present a memory optimal perspective transform hardware engine to support typical imaging and vision processing problems that can be either a standalone engine or within the image signal processor (ISP) [1][2][3].

A. Perspective Transform (Homography)

When a camera is viewing a scene from two different positions or when multiple cameras are viewing the scene from different positions, a transformation between the two different "perspectives" is needed to align the images [4]. Under specific conditions, these perspectives are related by a "homography" – a transformation that captures the exact geometric relationship between the images. A typical application of homography is to align and stitch multiple frames of the same scene to compute a panoramic output image. The homography is also used in computing depth maps from a stereo image pair. By rectifying the two views, the search to compute disparity between the two views is simplified to a 1-D search problem. The homography transformation matrix from (hu, vu) to (hp, vp) is shown in Fig. 1 and Fig. 2 with transform parameters a to h and depth z. The first step is an affine transform and the second step is the perspective correction.

$$\begin{bmatrix} h_{aff} \\ v_{aff} \\ z \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \bullet \begin{bmatrix} h_u \\ v_u \\ 1 \end{bmatrix}$$

Fig. 1 Homography transformation – Affine transform

$$\begin{bmatrix} h_p \\ v_p \end{bmatrix} = \begin{bmatrix} h_{aff} \\ v_{aff} \end{bmatrix} \bullet \frac{1}{z}$$

Fig. 2 Homography transformation – Perspective transform

Fig. 3 Example of perspective correction (Left – Original, Right – Transformed)

Fig. 3 shows a practical application of such a transform to change the view of camera by applying the perspective transform. In the stereo rectification application, the matrix is pre-computed at the calibration step and remains fixed [5][6]

B. Perspective Transform Hardware Algorithm

The perspective transform is implemented as back mapping of two-dimensional (2D) output blocks in the frame as shown in Fig. 4 as part of image processing pipe [7][8][9]. Back mapping gives coordinates of the distorted image as a function of coordinates of the undistorted output image. Correction involves back-mapping each output pixel to a location in the source distorted image, and thus the corrected image is fully populated. As the distorted pixel locations mostly fall onto fractional coordinates, correction involves interpolation among the nearest available pixels.

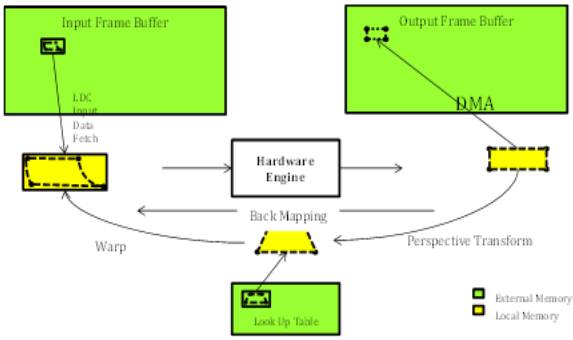


Fig. 4 Perspective transform Hardware using Back-mapping of 2D-Block

C. DRAM Memory Expansion in Input Frame Read

The perspective transform results in inefficient input memory read during back mapping operation. The graph in Fig. 5 shows the amount external memory (DRAM) BW needed for input and output frame. As seen in figure, on an average 3.2X input frame bandwidth is needed for back mapping operation instead of the ideal ratio of 1X, i.e same as output frame memory bandwidth.

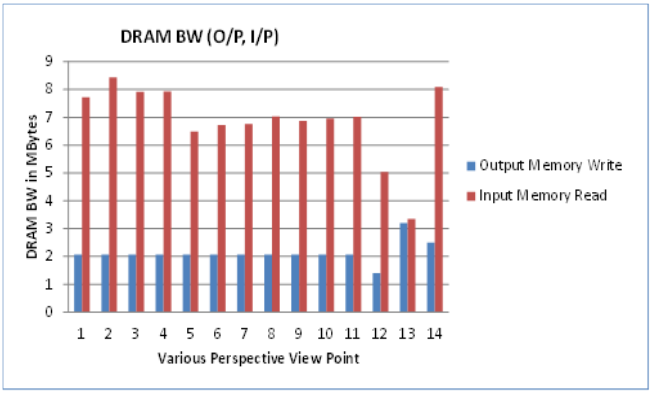


Fig. 5 DRAM BW for Input Read vs Output Write

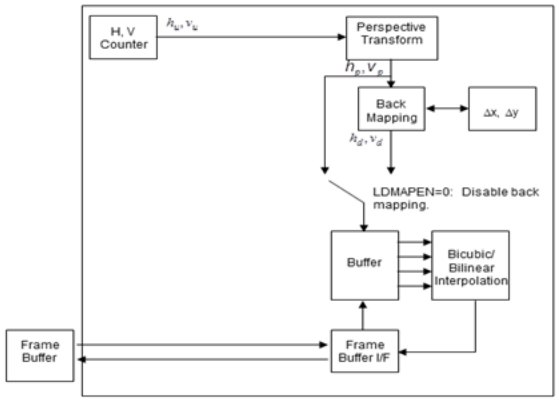


Fig. 6 Perspective Transform Engine

The typical cause of inefficiency is due to overlapping input blocks for different output blocks during 2D block-based operation and DRAM overheads in terms of un-aligned input block read and DRAM bank thrashing. The inefficiency is dependent on the dimension of 2D input block i.e. bigger block being more efficient.

The rest of the paper is organized as follows. Section II explains the proposed solution in terms of basic architecture, concept of spatially adaptive slicing of image frame into multi-region and block dimension selection. Section III explains the algorithm to find the boundaries of multi-region frame slicing as well as optimal 2D block dimensions. Section IV provides results, with section V concluding the paper.

II. Proposed Solution

Perspective Transform Engine computes 2D block of input image required to produce each 2D block in the output image. The input block is fetched and stored in internal memory for the corresponding output block to effectively handle the input image data fetch [10]. As shown in Fig. 6, the Perspective Transform Engine module consists of a Back Mapping block, a Δx/Δy offset table for back-mapping, a memory interface to access input and output frame buffers and an interpolation block. The frame buffer is external to this module and is usually in an off-chip DRAM. To apply the distortion and perspective correction efficiently in terms of time and memory bandwidth, the hardware processes the frame buffer using 2D blocks. The firmware configures appropriate parameters and then initiates the transform engine by writing to a control register. The hardware controls the sequencing through 2D blocks, DMA transfers, and computation to process an entire image autonomously. The proposed solution is flexible to supports multiple data formats e.g. YUV420 and YUV422. The proposed solution supports programmable output block sizes enabling a wide range of use cases with existing internal buffers. It also supports Region-Of-Interest processing optimizing performance and bandwidth.

A. Spatially Adaptive Slicing

Fig. 7 shows the mapping of input block and output block for a given perspective transform. As shown in the figure, input block dimensions vary across the image depending on the lens, viewing angle and location of the block in a given frame.

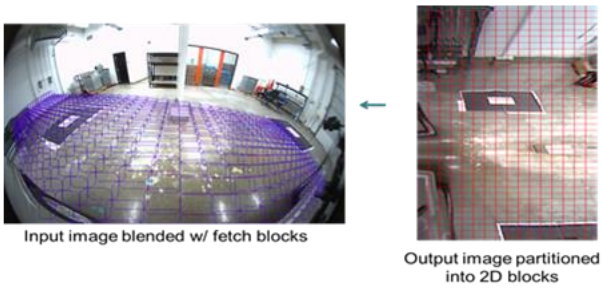


Fig. 7 Output to Input block image back mapping view

Fig. 8 captures variation in the magnification factor (input block size vs output block size) across the image. The majority of the block has almost no magnification, while a few blocks towards the center of the image have large magnification. This means block size dimensions have to be different at the border of image compared to center to allow hardware engine to work within a fixed size of internal memory.

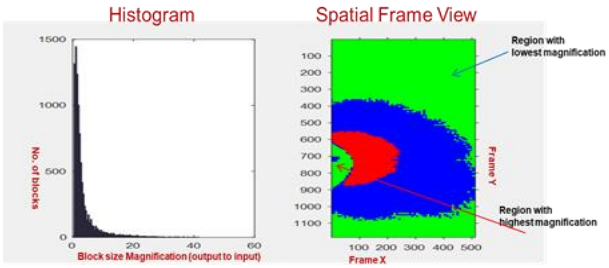


Fig. 8 Histogram and Spatial view of the magnification factor

To solve the above challenge, the paper proposes spatially adaptive output block size across the frame. In this case, the entire frame can be divided into up to 3x3 regions and each region can be programmed with independent block size based on the worst magnification factor, as shown in Fig. 9. In addition, the proposed solution allows selecting block size to be non-multiple of region size, allowing any arbitrary 2D block size for the worst magnification ratio.

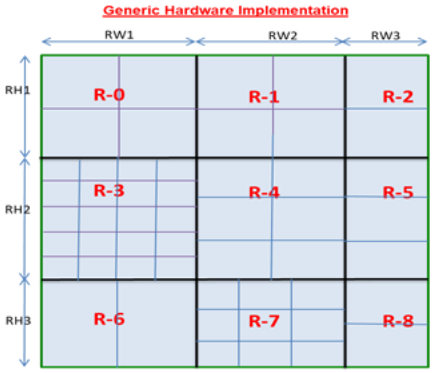


Fig. 9 Frame Division into multiple regions

III. Algorithm for Spatially Adaptive Slicing and Block Dimension Selection

The problem of optimally dividing a frame into 3x3 regions and calculating optimal block size for each region is an NP-Hard problem, due to its non-linear and non-deterministic nature. The paper proposes an innovative and unique algorithm to solve this problem using constraint minimization. The proposed solution consists of two step algorithm for calculating optimal parameters. In step 1, the algorithm finds optimal block size, while in step 2, the frame is divided into multiple regions along with finding optimal block size for that region.

A. Optimal Block Size Selection Algorithm

The algorithm uses a heuristic approach to calculate the optimal block size for the entire frame as shown in Fig. 10. It starts with the maximum possible block size and iterates over block size, reducing block size by a fixed amount. DRAM BW required is calculated on each iteration and block size with the lowest DRAM BW is selected as the optimal block size.

B. Spatially Adaptive Slicing Exhaustive Algorithm

The problem of optimally dividing the entire frame into regions can be considered as a constraint minimization problem. The idea is to minimize the value of function *f*, subject to certain constraints. The function *f* could be written as

$$f = \sum_{i=1}^{9} B_{opt}(obw_i, obh_i, \lceil \frac{X_i}{obw_i} \rceil, \lceil \frac{Y_i}{obh_i} \rceil),$$

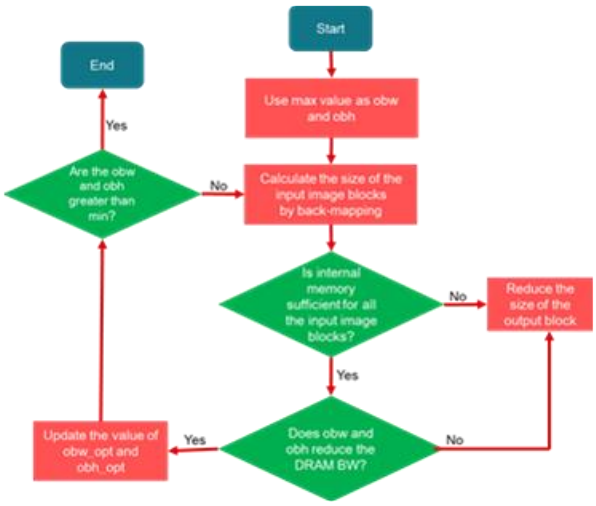


Fig. 10 Optimal Block Size calculation algorithm

In the equation,

obw_i, obh_i = Optimal Block Dimensions for Region-i

B_{opt} = Method of finding DRAM BW for Region-i

X_i, Y_i = Dimension of Region-i, where Xi € {W1, W2, W3} and Yi € {H1, H2, H3}

The constraint can be written as {W1 + W2 + W3 = frame width} and {H1 + H2 + H3 = frame height}.

The initial value of the optimal block size is calculated as shown in section A – Optimal Block Size Selection Algorithm. Since region size cannot be less than the optimal block size of the entire frame, it is used as initial region size (W1, H1). Keeping the value of W1 constant, the value of W2 is iterated over by a fixed jump value as shown in Fig. 11. On each iteration, optimal block size and optimal DRAM BW are calculated for this region division and this region configuration is assigned as f_{new}. When $f_{new} < f_{old}$, then f_{new} = optimal region configuration. The same process is also repeated on the vertical side.

The jump size is determined based on the following two factors,

1. W1 , W2 cannot be less than obw_opt
2. H1 , H2 cannot be less than obh_opt

The jump size is chosen to be twice the optimal block size so that the regions get enough space to try out for the maximum possible block sizes.

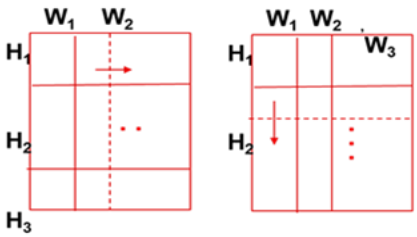


Fig. 11 Optimal Region Size calculation algorithm

IV. Results

Perspective Transform Engine described in this paper is available in the Texas Instruments (TI) J7ES System-on-Chip (SOC) processor family [11]. TABLE I. gives hardware engine design and throughput details.

TABLE I. PERFORMANCE DETAILS OF SOLUTION

Process	16 FinFet
Throughputs	1 cycle/output pixel
Frequency	720MHz

The proposed algorithm is validated using various frames, each with different FOV (field of view) and each requiring a different degree of perspective transformation and distortion correction. As shown in Fig 12, the result clearly shows that using optimal block size alone can reduce the DRAM BW by an average of 58%.

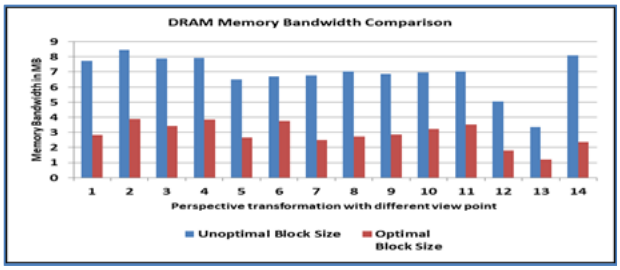


Fig. 12 DRAM BW improvement due to optimal block size

Additionally, the given frame can be divided into multi-region by spatial adaptive frame slicing into 3x3 regions. The DRAM BW can further be reduced by average of 21% due to multi-region slicing as per the proposed algorithm. When both optimal block size and multiple region methods are applied, there is average reduction of DRAM BW by 67% as shown in Fig 13.

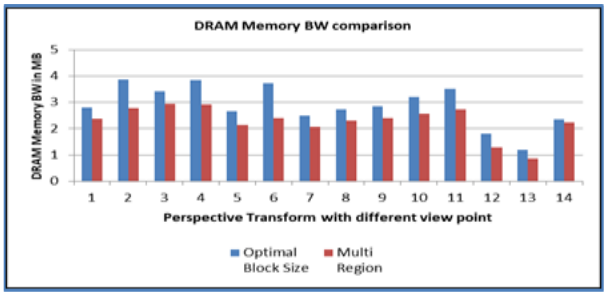


Fig. 13 DRAM BW improvement due to splicing into multiple regions

V. SUMMARY

The paper proposed a memory (DRAM) bandwidth (BW) optimal perspective transform (or homography) engine addressing multiple use-cases spanning across computer vision and image processing domain. This solution consists of multiple knobs, namely multiple region selection with block size tuned for each region and algorithm for optimal block dimension selection. These techniques resulted in saving up to 67% DRAM BW for typical use-case scenarios.

References

[1] M. Mody, H. Sanghvi, N. Nandan, , et. al., "High performance and flexible imaging Sub-system," Advances in Computing, Communications and Informatics (ICACCI), Sept. 2014

[2] M. Mody, N. Nandan, S, Dabral, et. al. "Image Signal Processing for Front Camera based Automated Driver Assistance System", Consumer Electronics (ICCE) Berlin, 2015, IEEE Conference, Sep. 2015.

[3] M. Mody, S, Dabral, M. Magla et. al. "High Quality Image Processing System for ADAS", Consumer Electronics (ICCE) Asia, IEEE Conference, Sep 2016

[4] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, 2004

[5] C. Vancea and S. Nedevschi, "LUT-based Image Rectification Module Implemented in FPGA," in IEEE International Conference on IntelligentComputer Communication and Processing, 2007

[6] A. Akin, I. Baz, L. Gaemperle, A. Schmid, and Y. Leblebici, "Compressed Look-Up-Table Based Real-Time Rectification Hardware," in IFIP/IEEE 21st Int. Conf

[7] M. Mody, G. Hua, S. Dabral, et. al., "High Quality Image Processing System for ADAS", IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2019

[8] M. Mody, K. Chitnis, A. Dubey, et. al., "A 216 GOPS Flexible WDR Image Processor for ADAS SoC", IEEE Coolchips, April 2017

[9] M. Mody, S, Dabral, M. Magla et. al. "High Quality Image Processing System for ADAS", Consumer Electronics (ICCE) Asia, IEEE Conference, Sep 2016

[10] M. Mody, S. Shivalingapa, A. Rajashekar, et. al., "Flexible and Efficient Perspective Transform Engine", ICCE Banglore 2017

[11] Next Generation JacintoTM 7 automotive processors, http://www.ti.com/processors/automotive-processors/featured-platform.html

Author Biography

Mihir Mody is Senior principal architect (DMTS) in embedded processor business, responsible for defining IP & SoC for Automotive and Industrial market in Texas Instrument (TI). He received his master's in electrical engineering from Indian Institute of Science (IISc) in 2000. He is working in TI from 2001 to architect an area efficient and low power solutions for image processing, computer vision, video coding, deep learning and control systems.

Rajasekhar Allu is Senior Designer Engineer with Texas Instruments with more than 18 years' experience in Digital design and Architecture. He has extensive experience in the field of Vision, Imaging and other Multi-Media domains.

Gang Hua received the B.S. and M.E. degrees in electrical engineering from Tsinghua University, Beijing, China in 1998 and 2000, respectively, and the M.S. and Ph.D. degrees from Rice University, Houston, TX, in 2005 and 2010, respectively. He joined Texas Instruments in Houston in 2008 and worked on video coding and imaging technologies. Now, he is an image architect with Texas Instruments.

Brijesh Jadav is Lead Software Engineer for Automotive ADAS business at Texas Instruments (TI) Incorporated. His domains of interest are imaging, video, vision system software and frameworks for embedded solutions. He received Bachelor of Engineering (BE) degree from the Sardar Patel University, 2001.

Ankur completed his B. Tech in Computer Engineering from Netaji Subhas Institute of Technology (2012). He has worked on wireless connectivity (Wi-Fi) and working in ADAS processors team with a focus on Vision HWA at Texas Instruments India..

Niraj Nandan has completed Bachelor of Technology in electrical engineering department from Indian Institute of Technology, kanpur, in May 2001. He is currently working at Texas Instruments, Dallas,USA. He worked on multiple domains spanning across analytics, vision, Imaging and video for past 18 years. He has contributed in design and architecture of multiple generations of Jacinto, OMAP and its IPs. Currently He is leading the development effort of Imaging, vision and multimedia solutions for Automotive and industrial Markets.

Mayank Mangla is an experienced imaging engineer, leading camera R&D for Texas Instruments Embedded Processors product line. He has been passionate about image science from days when digital cameras had just arrived on the horizon. Over the years he has been instrumental in 100s of millions of cameras reaching the consumers. His current research interests include application of imaging and computer vision in ADAS and Robotics.

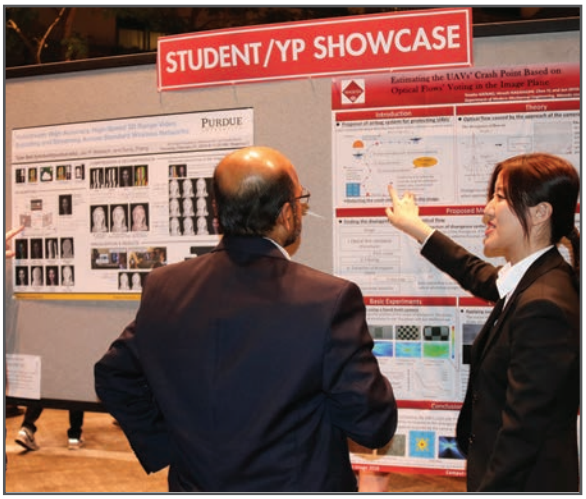
JOIN US AT THE NEXT EI!

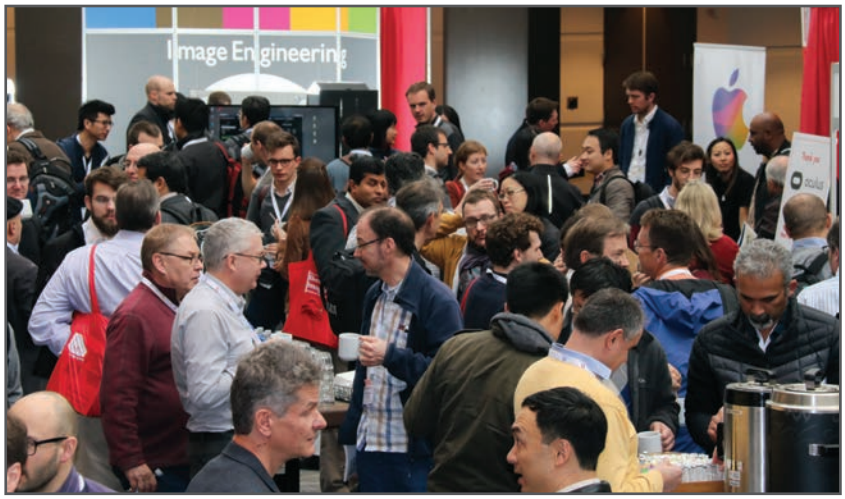
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org