# Survey on Image Rectification on FPGA

R. FELLAG†, Z. IRKI †, and A. KHOUAS‡
†Ecole Militaire Polytechnique, Borj El Bahri, Alger, ‡Université de Boumerdes
fellag.ratiba@yahoo.com , zohir_irki@yahoo.fr, akhouas@hotmail.com

***Abstract* — Stereo vision is a technique for building a 3D description of a scene observed from two or more view points. It is most often used for many applications in robotics including 3D object recognition, localization as well as 3D navigation of mobile robots. Applications that are based on stereo vision exploit the correspondence between pixels between images at hand. However, this correspondence problem is one of the major issues in stereo vision, as it requires the search for matches in a 2D area but this problem can be reduced to 1D line search if images were rectified.**

**Stereo image rectification is the process of transforming images to obtain image planes to be parallel, epipolar lines to be collinear and parallel to one of the image axes (usually the x-axis).**

**Several systems have been successfully implemented to perform real time image rectification on FPGA. Most of these implementations use Look Up Table (LUT) approach on a calibrated stereo rig.**

***Keywords* —Calibration, FPGA, image rectification, LUT, rectification algorithmes.**

## I. Introduction

HUMAN beings have two eyes; most animals also have two eyes. The study of the human vision system gives us an idea of how images of the scene are captured by the eye and processed by the brain. This knowledge will allow us to use machines to implement similar functionalities. Furthermore, the knowledge of how the human vision system uses the scene information to estimate depth will give us an idea of how to implement this type of functionality with a machine. In computer vision systems, we must rely on cameras to capture images of the scene at hand and thus we must understand the image formation process.

Stereo vision is a technique for building a 3D description of a scene observed from two or more view points. It is most often used for many applications in robotics including 3D object recognition and localization as well as 3D navigation of mobile robots [1]. However, one of the major issues in the application of stereo vision techniques is the correspondence problem which can be defined as locating a pair of image pixels from two different images (left and right images) where these two pixels are projections of the same scene element. Normally this is a 2D area search problem, but it can be reduced to a 1D line search problem if the images were captured using a canonical stereo system. Such a system requires both cameras to be perfectly aligned, image planes to be parallel, epipolar lines to be collinear and parallel to one of the image axes (usually the x-axis).

In practice it is almost impossible to obtain such a canonical configuration of the stereo rig, but it is possible to apply a rectification process over the images which will make them appear as if they were captured using a canonical system.

The purpose of rectification is to transform the general stereo geometry to that of the standard stereo setup. In this simple geometry the image planes are parallel and coordinate systems aligned. Then the epipoles are at infinity and the epipolar lines are parallel and along the x-direction (scan lines). After performing the image rectification process, the problem of stereo matching reduces to a one-dimensional search along the rows of the images. The depth of a 3D point becomes inversely proportional to the disparity, i.e. the horizontal distance between corresponding image points.

The main objective of this survey is the study of the different rectification algorithms and the state-of-the-art implementations on FPGA of real time image rectification systems. Such systems acquire stereo images, perform stereo rectification and generate a pair of stereo rectified images, which will be used later for different applications.

The remainder of this paper is organized as follows. In section two, we give the theoretical background on image rectification and epipolar geometry. Section three describes the rectification algorithms we found in the literature. Section four presents the state-of-the-art real time image rectification systems on FPGA. Section five draws some conclusions. The paper terminates with a reference section.

## II. Rectification and Epipolar Geometry

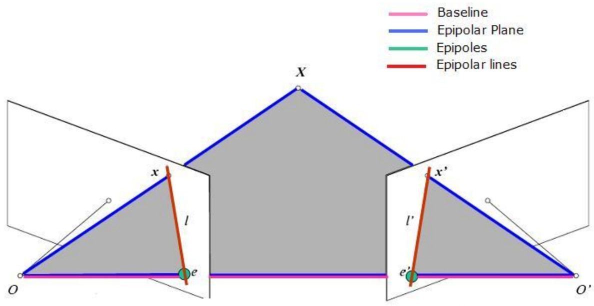


Fig. 1. Epipolar Geometry

Epipolar geometry is the geometry of stereo vision. When two cameras view a 3D scene from two distinct positions, there are a number of geometric relations between 3D points and their projections onto the 2D image that lead to constraints between image points. These relations are derived based on the assumptions that the cameras can be approximated by pinhole camera model [3].

From Fig. 1, we can give the basic definitions of the terms used in epipolar geometry:

- *Baseline:* is the line connecting the two camera's centers.
- *Epipolar plane*: is the plane containing the baseline.
- *Epipoles*: are intersections of the baseline with the image planes, each epipole is the projection of a camera's center in the other image plane.
- *Epipolar lines:* are intersections of the epipolar plane with the image planes (always comes in corresponding pairs).

Using a stereo rig composed by two pinhole cameras. A 3D point W is projected onto both image planes to two points which constitute a conjugate pair. Given any point in the left image its corresponding point in the right image is constrained to lie on the epipolar line. This is called the epipolar constraint. All the epipolar lines in one image plane pass through a common point the epipole. Using these constraints, any pair of images can be transformed so that epipolar lines are parallel and horizontal in each image. This procedure is called rectification (see Fig. 2).

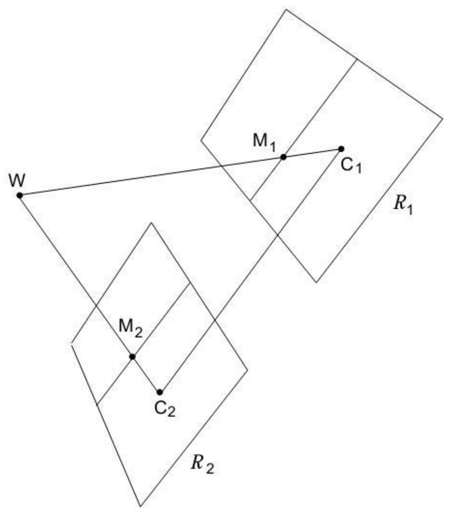


Fig. 2. Stereo rectified pair of images [2]

## III. RECTIFICATION ALGORITHMS

Many rectification algorithms may be found in the literature. They can be classified into three different approaches. Some of them use the fundamental matrix [4]-[8] in order to extract the epipolar line geometric information. Other methods are based on calculating the desired canonical system's Perspective Projection Matrices (PPMs) [2]-[9] which are used to project 3D points onto the rectified image planes. In the former case no calibration process is required while in the latter case, the rectification matrices are obtained using both new PPMs and original PPMs and this approach requires the intrinsic and extrinsic parameters of the stereo vision system, so a preliminary calibration process should be performed first. Other approaches [1] use image rectification on trinocular stereo vision systems, taking advantage of the new constraints imposed by having an extra baseline.

In this survey, we concentrated our study on rectification algorithms that are used for binocular stereo vision systems. Therefore, we can further classify the image rectification algorithms into two major types: those which are based on a calibrated stereo rig and those which are based on an uncalibrated stereo rig. We will describe the main rectification algorithms from both types in what follows.

### *A. Rectification algorithms: Uncalibrated case*

Among the rectification algorithms we found in the literature that are build upon an uncalibrated stereo rig, we can find Hartley's algorithm [5] which is a fundamental algorithm about image rectification. In this algorithm, at least eight match points between right and left images have to be found in order to construct the fundamental matrix F. Then a projective matrix (homography) is estimated through the constraint that the vertical difference between the corresponding points should be minimized and the epipole should be at infinity. Even though this method does not need to calibrate the cameras, but it is complex as several corresponding points are required to be previously known and many variables are needed to be resolved.

Loop and Zhang [6] build their work upon Hartley's algorithm but they suggested decomposing each homography into projective and affine components. They then found the projective component that minimizes a defined projective distortion criterion. They further decomposed the affine component of each homography into a pair of simpler transforms, one designed to satisfy the constraints for rectification, the other is used to further reduce the distortion which is introduced by the projective component. Therefore, the homography calculated is divided into three different transforms to attempt to make the projective component "as affine as possible".

Isgrò and Trucco [4] proposed a method in which the transformation matrix could be gained directly from the corresponding points without computing the fundamental matrix, but the corresponding points should be previously known while computing the rectification transformation matrix , using the same distortion criterion as in [5].

The principal steps for image rectification on an uncalibrated stereo rig are summarized below:

- Identify a set of correspondences (at least eight matches)
- Estimate the fundamental matrix F, except using [4].
- Estimate two homographies H, H' and minimize some selection criteria to improve the estimation.
- Apply those homographies to images.

### *B. Rectification algorithms: Calibrated case*

In the case of a calibrated stereo rig, we find mainly works by Ayache and Lustman [9] who combined the projective matrix taken from camera's calibration result with the condition that the corresponding epipolar lines should be in the same vertical coordinates after rectification to derive the required transformation matrix. But this method made the computations large as it requires the computation of epipolar lines. In derivation Fusiello, *et al* [2] improved the above algorithm to be more simple and compact for a calibrated, unconstrained stereo rig. The algorithm takes the two perspective projection matrices of the original cameras, and computes a pair of rectifying projection matrices. Details about the derivation of the rectification perspective matrices are given in the compact algorithm provided in [2].

The main steps for image rectification on a calibrated stereo rig using [2] are summarized below:

- Calibrate the stereo rig to get the intrinsic and extrinsic parameters using Matlab Camera calibration toolbox or OpenCv computer vision library.
- Calculate the Perspective Projective Matrices (PPMs) of the original system.
- Define two new PPMs which are obtained by rotating the old ones around their optical centers until focal planes become coplanar.
- Calculate the transformation matrices
- Apply those transformations to images.

### *C. Comparison between rectification algorithms*

In this section, we will present a comparison between the two types of rectification algorithms that we found in the literature.

Fusiello, *et al.*[2], [10] have worked on both cases calibrated and uncalibrated stereo rigs and they claim that the practice has shown that the rectification produced by the uncalibrated methods is not always satisfactory, if compared to results obtained by the calibrated case [10].

Furthermore, even though rectification algorithms are inclined to the stereo rectification based on uncalibrated cameras as compared with the methods based on calibrated cameras, they could save the process of resolving camera's intrinsic and extrinsic parameters, but most of which generally required a set of previous known corresponding points. Moreover, in the applications of stereo vision, especially in 3D reconstruction with the disparities, the intrinsic and extrinsic parameters of cameras generally should be obtained in advance [11]. Camera calibration process can be performed using Matlab camera calibration toolbox [12] or OpenCV library which is no more than the C implementation of this toolbox. Another advantage of using the calibrated case is the possibility to perform distortion minimization by including distortion models into the calibration process. Both radial and tangential distortions may be considered during the calibration process. Hence, image rectification not only enables the correspondence matching to be fast and accurate but also increases the visual quality of the images as 3D contents.

## IV. IMPLEMENTATIONS OF IMAGE RECTIFICATION SYSTEMS ON FPGA

Several works have been successfully implemented to perform real time image rectification on FPGA. There exist two approaches for implementing an image rectification system on a hardware architecture, one in which the calculation of the image coordinates is done in real time and for each pixel. The other one calculates the images coordinates for each pixel offline and store them back in a lookup table which will be loaded only once. The former approach requires less memory but time consuming whereas the latter requires more memory but it is suitable for real time applications. The look up table approach is only valid for the calibrated case.

Most of the implementations found in the literature use the look up table approach to perform real time stereo image rectification [13-18]. Among which we find, in [13] Park *et al.* have proposed a real time image rectification technique based on a compressed look up table. They used differential encoding to compress the look up table thus reducing its size. They implemented the rectification module on a Xilinx FPGA Virtex5 and they obtained a compression ratio of 73% to fulfill real-time requirement (i.e. 40 fps at 74.25 MHz). The Look up tables can be constructed by using software as a preprocessing since they represent the results of computing camera calibration with intrinsic and extrinsic parameters and those parameters are unchanged.

In reference [14] Maldeniya *et al.* implemented a stereo camera pair which produces a rectified real time image output with a resolution of 320x240 at a frame rate of 15fps and delivers them via a 100-Ethernet interface. They used a Spartan 3E FPGA for real-time processing within which they implemented an image rectification algorithm. They used the compact algorithm proposed by A. Fusiello *et al.* [2] to derive the calibration matrix and to obtain the new perspective projective matrices that are used later in the canonical image rectification module. A look up table is used to store the coordinates of the rectified image planes. To store the look up table on-board PSDRAM was used. Bilinear transformation was used to find pixel's intensity. Results of the implementation of the rectification module have shown that the system was capable of rectifying two videos in real-time minimizing the vertical disparity between them. Canonical rectification reduces the computation complexity and saves resources in the FPGA.

The work presented in [15] by Vancea and Nedevschi presents a real-time hardware architecture able to perform simultaneously image rectification and distortion removal. The entire process is based on look up tables relating pixels from rectified image and original image. The look up tables are filled using results from the computation of mathematical expressions which includes the camera model (pinhole), radial and tangential distortions. The size of images accepted by this hardware solution can vary by simply changing some configuration parameters of the VHDL description. The strategy is to parse the original image and to use the LUT to find the corresponding

TABLE 1: COMPARISON BETWEEN DIFFERENT IMPLEMENTATIONS OF RECTIFICATION SYSTEMS ON FPGA

| *Year* | *Authors* | *FPGA used* | *Image Resolution* | *Frame rate* | *Approach* |
|---|---|---|---|---|---|
| **2011** | D. H. Park, *et al.* | Xilinx Virtex5 | 1280x720 | 40 fps | Compressed LUT |
| **2010** | B. Maldeniya, *et al.* | Xilinx Spartan3E | 320x240 | 15 fps | LUT |
| **2010** | J. G. P. Rodrigues and J. C. Ferreira | Xilinx Spartan3XC3S1500 | 640x480 | 25 fps | Softcore processor (Microblaze) |
| **2009** | K. Jawed, *et al.* | Altera StratixIII | 1024x768 | 30 fps | LUT |
| **2008** | E. Staudinger, *et al.* | Altera Stratix EP1S60 | 1024x1024 | 35 fps | softcore processor (Nios II/f) |
| **2007** | C. Vancea and S. Nedevschi | Xilinx VirtexE600 | 640x512 | 85 fps | LUT |

position of each pixel in the rectified image.

In reference [16] Jawed *et al.* constructed a system that produces `3D movies' with 1Mpixel frames at 30 fps. It consists of three main modules: (a) rectification, (b) stereo correspondence and (c) scene interpretation. The rectification module removes distortion and misalignment of images to simplify stereo matching. An offline calibration process was accomplished using functions from the OpenCV cvCalibFilter class to determine distortion parameters and system extrinsic parameters. These displacements are stored in lookup tables which become part of the of the rectification module.

Another way to implement a stereo rectification system without the use of look up tables is presented in [17] where Rodrigues and Ferreira presented a different approach using a softcore processor , the MicroBlaze, to perform slow but accurate calculations and a fast dedicated hardware support to achieve real time image rectification. They used a Hartley like rectification algorithm in which they adopted a simplified method to solve the correspondence problem through the use of some small weighting functions. The transformation matrices required to rectify images are then calculated for each camera. For each coordinate of the final rectified images, the coordinates of the pixels from the unrectified images must be used to calculate (by interpolation) the new pixel value. The system was implemented on a Spartan-3 FPGA and was able to perform rectification on images from two cameras with a resolution of 640x480 and a frame rate of 25 fps.

In reference [18] Staudinger, *et al.* present the implementation of a Lens Distortion and Rectication Unit (LDRU) which is well suited for embedded real-time systems. It has been realized on an Altera STRATIX EP1S60 FPGA resulting in a performance of 35 frames per second for a 1024x1024 pixels input image. A generic IP-Core with a simple input/output image interface, a parameter cache interface for the combined undistortion and rectification coefficients and minimized FPGA utilization is designed and tested on an existing platform. A softcore processor Nios II/f is used to initialize the system and cameras, to ensure proper error handling and to generate the offset parameters and to stores them in an external DDR-RAM memory bank.

A comparison between state-of-the-art systems that implement real time rectification of stereo images on FPGA is given in Table. 1. As we can see, most of the implementations use the look up table (LUT) approach. A calibration process is performed before the construction of the LUTs. It can be done using Matlab Camera Calibration Toolbox, OpenCV library or by the evaluation of mathematical expressions using any other mathematical tool.

Advantages of using LUT approach to implement stereo image rectification systems are manifold; using LUTs we can efficiently achieve real time performance on a hardware architecture (FPGA), also by combining image rectification with distortion removal (radial and tangential distortions) in a single step, we increase the visual quality of the images. However, this approach requires a lot of memory space as the LUT's size is the same as the size of the image. Techniques to reduce the LUT's size have been developed such as the use of differential encoding as in [13] where the compression ratio was up to 73%. Adopting the other approach which is based on the use of a softcore processor to compute the transformation matrices that will be applied to rectify the pair of image can also provide good results. The softcore processor can provide slow but accurate calculations but a dedicated fast hardware must be provided in order to achieve real time performance. Contributions to improve the performance of the real time stereo image rectification systems on FPGA may be directed toward developing techniques to reduce the look up table size which will overcome the memory space requirements as we increase the image's resolution. Merging the two approaches look up table with the use of a softcore processor may also give better results.

## V. Conclusion

In this paper, we have done a survey about image rectification. We concentrated our study on the binocular stereo vision systems. We presented the theoretical background about stereo image rectification, epipolar geometry and rectification algorithms. We classified the rectification algorithms into two types: based on calibrated stereo rig and based on an uncalibrated stereo rig. After a comparison between the two types, we found from literature that some authors claim that results from the calibrated case are better than those of uncalibrated case. State-of-the-art real time image rectification systems on FPGA have been presented in this paper. Most of the existing systems use the look up table (LUT) approach to perform real time image rectification on FPGA. In the latter case, a calibration process in required to get the intrinsic and extrinsic camera parameters.

## References


[1] N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, pp. 73-85, 1991.

[2] A. Fusiello*, et al.*, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications,* vol. 12, pp. 16-22, 2000.

[3] *Epipolar Geometry*. Available: http://en.wikipedia.org/wiki/Epipolar_geometry

[4] F. Isgro and E. Trucco, "Projective rectification without epipolar geometry," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.* p. 99 Vol. 1.

[5] R. I. Hartley, "Theory and Practice of Projective Rectification," *International Journal of Computer Vision,* vol. 35, pp. 115-127, 1999.

[6] C. Loop and Z. Zhengyou, "Computing rectifying homographies for stereo vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.* p. 131 Vol. 1.

[7] D. V. Papadimitriou and T. J. Dennis, "Epipolar line estimation and rectification for stereo image pairs," *Image Processing, IEEE Transactions on,* vol. 5, pp. 672-676, 1996.

[8] M. Pollefeys*, et al.*, "A simple and efficient rectification method for general motion," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.* pp. 496-501 vol.1.

[9] N. Ayache and C. Hansen, "Rectification of images for binocular and trinocular stereovision," in *9th International Conference on Pattern Recognition* , 1988, pp. 11-16 vol.1.

[10] A. Fusiello and L. Irsara, "Quasi-Euclidean uncalibrated epipolar rectification," in *19th International Conference on Pattern Recognition. ICPR 2008.* pp. 1-4.

[11] H. Su and B. He, "Stereo rectification of calibrated image pairs based on geometric transformation," *International Journal of Modern Education and Computer Science (IJMECS),* vol. 3, pp. 17-24, 2011.

[12] J.-Y. Bouguet. *Matlab Camera Calibration Toolbox*. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[13] D. H. Park*, et al.*, "Real time rectification using differentially encoded lookup table," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, 2011.

[14] B. Maldeniya*, et al.*, "Computationally efficient implementation of video rectification in an FPGA for stereo vision applications," in *5th International Conference on Information and Automation for Sustainability (ICIAFs), 2010*, pp. 219-224.

[15] C. Vancea and S. Nedevschi, "LUT-based Image Rectification Module Implemented in FPGA," in *IEEE International Conference on Intelligent Computer Communication and Processing, 2007*, pp. 147-154.

[16] K. Jawed*, et al.*, "Real Time Rectification for Stereo Correspondence," in *International Conference on Computational Science and Engineering, 2009. CSE '09*, pp. 277-284.

[17] J. G. P. Rodrigues and J. C. Ferreira, "FPGA-based rectification of stereo images," in *Conference on Design and Architectures for Signal and Image Processing (DASIP),* 2010, pp. 199-206.

[18] E. Staudinger*, et al.*, "Fpga-based rectification and lens undistortion for a real-time embedded stereo vision sensor," *Proc. FH Science Day* , pp. 18--25, 2008.