

## Design and hardware implementation of a stereo-matching system based on dynamic programming

J.A. Kalomiros<sup>a,\*</sup>, J. Lygouras<sup>b</sup>

<sup>a</sup> Department of Informatics and Communications, School of Technological Applications, Technological Educational Institute of Serres, Terma Magnisias, 62124 Serres, Greece

<sup>b</sup> Section of Electronics and Information Systems Technology, Department of Electrical Eng. & Computer Eng., School of Engineering, Democritus University of Thrace, Xanthi, Greece

### ARTICLE INFO

#### Article history:

Available online 5 May 2011

#### Keywords:

Hardware design  
Real-time systems  
Stereo vision  
Dynamic programming

### ABSTRACT

A new real-time stereo system is presented based on a hardware implementation of an efficient Dynamic Programming algorithm. A simple state-machine calculates the cost-matrix along the diagonal of the 2-D disparity space for each epipolar pair of image scan-lines. Minimum transition costs are stored in embedded RAM and are used to backtrack disparities at clock rate. All calculations are within a pre-determined slice of the cost plane, representing the useful disparity range. The system is designed as a VHDL library component and is implemented as a SoC in a medium-capacity Field Programmable Gate Array chip. It can process stereo-pairs in full VGA resolution at a rate of 25 Mpixels/s and produces 8-bit dense disparity maps within a range of disparities up to 65 pixels. The design is evaluated comparing to ground truth and in terms of resource usage. It is also compared to a software implementation of the Dynamic Programming algorithm and to other FPGA-based stereo systems.

© 2011 Elsevier B.V. All rights reserved.

### 1. Introduction

Stereo vision estimates the depth of scene points from two or more images obtained from distinct viewpoints. In Fig. 1 the case of non-merged stereo geometry with two parallel cameras is shown. A location in space projects to two points, one in the left and one in the right stereo image, both on the same scan-line. The displacement of the projection in one image with respect to the other is the *disparity* [1].

Extracting the depth of a scene point using stereo pairs is based on finding correspondences, i.e. finding for each point in one image the matching point in the other image. Mapping all pixels in one image (the reference image) to disparity values, results in a set of values termed a *dense disparity map*. In order to visualize a dense disparity map one may attribute a gray level to each disparity value. In the resulting gray-scale image, lighter tones represent objects in the foreground whereas darker spots lie in the background.

Let us consider a point  $P_L$  in the left image with horizontal and vertical coordinates on the image plane  $u_L, v_L$ . It can be translated into a point  $P(X, Y, Z)$  in the frame of reference of the stereo head, using similar triangles, as shown in Fig. 1. This process is called triangulation:

$$X = \frac{fb}{(u_L - u_R)} \quad (1.1)$$

$$Y = \frac{1}{f}X(u_L - u_0) - \frac{b}{2} \quad (1.2)$$

$$Z = \frac{1}{f}X(v_L - v_0) \quad (1.3)$$

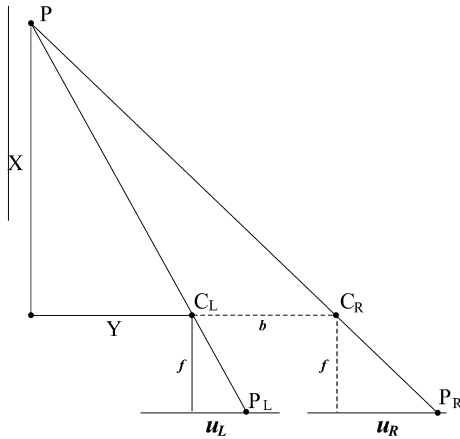
where  $f$  is the camera focal length measured in pixels,  $b$  is the “baseline”, i.e. the distance between left and right camera focal axes and  $(u_0, v_0)$  is the center of the image measured in pixels.

Stereo vision is an active area in machine vision as it has applications in robotics, navigation systems, object recognition, virtual reality etc. [2]. Real-time dense-depth extraction is a requirement for many robotic applications and is often based on special-purpose hardware, such as Digital Signal Processors (DSPs) and Application Specific Integrated Circuits (ASICs). In recent years, some applications are designed making use of Field Programmable Gate Array (FPGA) chips, which are reprogrammable, less expensive for prototyping than ASICs and have a relatively short design-cycle. Most implementations exploit the intrinsic parallelism of algorithms based on local correlations between image areas.

Local methods aggregate cost over a window defined around the pixel of interest and find correspondences by minimizing the cost function. A prime example is the well known Sum of Squared Differences (SSD) algorithm and its variance Sum of Absolute Differences (SAD) which find correspondences between local windows by minimizing the sum of squared or absolute differences along epipolar lines [1]. SAD uses a particularly simple and hardware-friendly metric of similarity, namely the sum of absolute intensity differences:

\* Corresponding author. Tel.: +30 2321022854; fax: +30 2321049128.

E-mail address: [ikalom@teiser.gr](mailto:ikalom@teiser.gr) (J.A. Kalomiros).



**Fig. 1.** View of two identical parallel cameras with focal length  $f$  at a distance  $b$  to each other. Similar triangles are used for depth extraction.

$$\sum_{u,v} |(I_1(u+x, v+y) - I_2(u+x+d, v+y))|. \quad (1.4)$$

$I_1$  and  $I_2$  refer to intensities in the left and right image,  $(x, y)$  is the center of a window in the first image,  $(x + d, y)$  is a point on the corresponding scan-line in the other image displaced by  $d$  with respect to its conjugate pair and  $u, v$  are indices inside the window. The point that minimizes the above measure is selected as the best match. While this method requires laborious search along epipolar lines in serial software implementations, it can be parallelized easily in hardware, allocating parallel comparisons between local windows to a number of processing elements.

SAD, normalized cross-correlation, phase correlation and other similar area-matching metrics have been used in hardware implementations of real-time stereo systems [1]. Although local methods can be efficient, they are sensitive to noise and to local ambiguities, like occlusion regions or regions with uniform texture. As a consequence, block-based techniques can result in many wrong matches or in reduced detail in object shape.

In contrast to local methods, global methods exploit non-local constraints that provide additional support for regions difficult to match locally, like occluded or uniform regions [3,4]. Such methods minimize wrong matches by providing an overall best solution for the correspondence problem along each epipolar line or even for the whole image. These methods result in dense disparity maps of good quality but are computationally very expensive. This is why global algorithms are not often used in real-time implementations, with Ref. [5] being among rare examples. Recently, there has been interest for hardware architectures for high speed implementation of global methods like belief propagation [6].

The most commonly used global algorithm for stereo matching is based on dynamic programming (DP) [7,8]. Dynamic programming stereo matching is a two-pass recurrent technique consisting of the cost-matrix building phase and the disparities backtracking phase. Because of the recursive nature of the computations, the technique lacks inherent parallelism, which makes it difficult to map into hardware. For this reason hardware implementations of stereo systems based on dynamic programming are very uncommon. Ref. [9] presents a hardware architecture for general real-time dynamic programming applications, while in a previous publication [10] we laid-down some design principles for a stereo system based on dynamic programming.

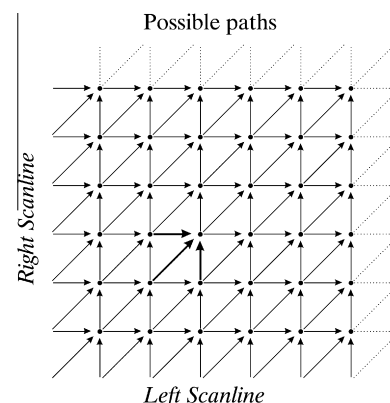
In this paper, a new stereo system is presented, which implements in hardware a variation of the DP stereo algorithm. We lay down the necessary key ideas and develop a prototype based on a Field Programmable Gate Array. The system processes stereo-

pairs in full VGA resolution, at a rate of 25 Mpixels per second and produces 8-bit dense disparity maps within a range of disparities up to 65 pixels. We show that the quality of the disparity maps is improved compared to block-matching hardware implementations of comparable size. The system is designed as a fully parametrizable VHDL library component that can be used in a stand-alone vision system or in a System-on-a-Programmable-Chip (SOPC) with an embedded processor, like Nios II.

The rest of the paper is organized as follows. In Section 2, a description of a standard dynamic programming stereo algorithm is given. In Section 3 a hardware-friendly algorithm is proposed and the main parts of the system are presented, namely the cost-computing and tag-storing stage and the backtracking stage. In Section 4 the system is evaluated in terms of resource usage in a variety of implementations. Also an evaluation of the resulting disparity maps is given, in terms of ground truth and by comparison to software implementations of stereo dynamic programming. In Section 5 our system is shown to compare favorably with other FPGA implementations of stereo-systems. The paper is concluded in Section 6.

## 2. Dynamic programming for stereo and hardware considerations

Dynamic programming is a general mathematical method that can reduce the complexity of optimization problems, by decomposing them into simpler and smaller problems [11]. The main idea behind a dynamic programming algorithm for stereo matching is to build a Disparity Space Image (DSI). A disparity space is a data space where a cost function is computed for all possible values of pixel displacement in a stereo-pair. In the present implementation, the DSI is computed for each scan-line as a two-dimensional matrix of the values of a global cost function. In this representation the  $x$  and  $y$  axes of the DSI are defined as the pixel-indices of the left and right scan-line [8]. The cost function is computed in stages, as the sum of the costs of partial paths obtained recursively. Cost-matrix computation proceeds from the lower left corner to the upper right corner of the DSI, as a result of the ordering constraint [12]. This procedure is shown in Fig. 2, where each point of the 2-D cost function is determined by taking into account the previously calculated cost values of the three neighboring points to the left. Cost computation for each path takes into account the local matching cost plus the occlusion cost for unmatched pixels. The matching cost used in the present implementation is the weighted squared intensity difference between any two pixels. The above procedure attributes a cost value to all disparities. After the cost-matrix computation, global optimization is efficiently performed along each



**Fig. 2.** Recursive calculations of the cost function. Each point in the cost plane is derived from the three previous values, as shown by the arrows.

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات