Pattern Recognition Letters 32 (2011) 2250–2253

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Real-time stereo matching using memory-efficient Belief Propagation for high-definition 3D telepresence systems

Jesús M. Pérez, Pablo Sánchez *

University of Cantabria, Av. Los Castros S/N, 39005 Santander, Spain

ARTICLE INFO

Article history:
Available online 30 June 2011

Keywords:
Stereo-vision
Belief Propagation
Real-time
FPGA

ABSTRACT

New generations of telecommunications systems will include high-definition 3D video that provides a telepresence feeling. These systems require high-quality depth maps to be generated in a very short time (very low latency, typically about 40 ms). Classical Belief Propagation algorithms (BP) generate high-quality depth maps but they require huge memory bandwidths that limit low-latency implementations of stereo-vision systems with high-definition images.

This paper proposes a real-time (latency inferior to 40 ms) high-definition (1280×720) stereo matching algorithm using Belief Propagation with good immersive feeling (80 disparity levels). There are two main contributions. The first is an improved BP algorithm with pixel classification that outperforms classical BP while reducing the number of memory accesses. The second is an adaptive message compression technique with a low performance penalty that greatly reduces the memory traffic. The combination of these techniques outperforms classical BP by about 6.0% while reducing the memory traffic by more than 90%.

1. Introduction

3D telepresence systems have implementation requirements that are hard to comply with. For example, the (Vision, 2009) prototype requires high-definition (HD, at least 1280×720 pixels at 30 fps), low latency (depth estimation in less than 40 ms) and good immersive feeling (high-quality depth estimation and more than 80 disparity levels). Additionally, it has to be implemented in a high-performance HW platform with a memory bandwidth of about 19 GB/s. As far as we know, there are no previous works satisfying all these requirements.

Stereo matching using Belief Propagation (BP) is one of the most efficient solutions for obtaining high-quality depth maps. This iterative algorithm passes messages, which model the cost of assigning disparities, among pixels in order to find a global solution with minimum assignation cost. Most of the work using BP is based on Felzenszwalb and Huttenlocher (2006), which is referred to as classical BP in this paper. However, the execution time of this algorithm in a PC cannot satisfy real-time (RT) requirements with HD images. Even ASIC/FPGA-based implementations, such as (Liang et al., 2009), cannot satisfy all (Vision, 2009) requirements. Some BP algorithms have been implemented in GPUs (Grauer-Gray and Kambhamettu, 2009) although they have limited performance mainly due to main-memory bandwidth limitations. Nowadays, memory bandwidth has become a more performance-limiting factor than the number of algorithm operations. To overcome this (Tseng et al., 2007) split the image into several unconnected regions. However, for RT applications, the size of these regions is normally very small and this greatly reduces performance. Moreover, some proposals have concentrated on reducing or compressing the number of messages in BP (Huq et al., 2008, Yu et al., 2007). However, they require the decompression of the messages and are not able to meet HD and RT constraints.

One way to improve BP is to include occlusion, potential error and texture-less region handling. A simple method of detecting occlusion is the cross-checking technique (Egnal and Wildes, 2002). Other occlusion-handling approaches generate better results (Sun et al., 2005) but they double the computational complexity. Some other techniques have improved depth estimation in texture-less areas (Yang et al., 2008), but only with low-resolution images and 48 disparity levels. Other approaches attempt to reduce potential errors (Gong and Yang, 2005), but they work with medium-resolution images.

Another way to improve BP is to chain several executions of the algorithm. Some proposals (Yang et al., 2009) use several BP modules and show good performance but high execution times.

This work presents a BP architecture that complies with actual telepresence system requirements (Vision, 2009). There are two main contributions. Firstly, it splits the algorithm into two BPs that work serially while reducing the number of memory accesses. Between the two blocks, the pixels are classified into four categories (occlusion, potential-error, texture-less and reliable pixel) and

* Corresponding author.
E-mail addresses: chuchi@teisa.unican.es (J.M. Pérez), sanchez@teisa.unican.es (P. Sánchez).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.06.016

their disparity assignation costs are redefined. Secondly, it defines an adaptive message compression technique to reduce memory traffic with little performance penalty, mainly because there is no need to uncompress.

The remainder of this paper is organized as follows. In Section 2, we discuss the double BP with occlusion, error and texture-less handling methods, as well as the compression technique used to fulfill the memory access requirements. Finally, we present the experimental results and conclusions in Sections 3 and 4.

2. Proposed real-time high-definition belief propagation

In Vision (2009), classical BP with a linear truncated model for messages was evaluated with an image library. This experimental study concluded that telepresence system requirements can be fulfilled with a minimum of 7 iterations and 80 disparity levels. Additionally, if variables are quantified using 16 bits the impact on results will be acceptable. With these heuristic parameters the classical BP technique satisfies most of the quality constraints of a telepresence system. However, it cannot satisfy the RT and memory bandwidth restrictions.

One of the most restrictive parameters is the number of external memory accesses. The actual high-performance platform, which is used as the hardware reference model in this work (Synplicity, 2009), could support up to 6 DDR2-400 memories with 64-bit words. The maximum number of memory accesses that can be performed in this platform is about 384 million, while the algorithm requires 2881 million. Thus, the system is not implementable in RT in an actual high-performance platform and it would require a reduction in the number of accesses by almost 90%.

2.1. Main algorithm

The proposed algorithm chains two BP executions (see Yang et al., 2009) with an important improvement: between executions, the pixels are classified by the OE module into reliable, occlusion, error and texture-less pixels. The OE module will be presented on Section 2.2. Moreover, in order to achieve the real time requirements, a new compression algorithm (explained in Section 2.3) is applied to the messages of the two BP algorithms. Hereinafter, this algorithm will be denoted as real-time high-definition belief propagation (RT-HD BP). It performs the following steps:

1. Read left and right images and **compute data-cost**
2. Iterative **BP** (BP1) **over all the pixels**
3. **Output**: for each pixel, send to the output (see Section 2.2):
   (a) **Minimum disparity label of the left-image** depth map.
   (b) **Third minimum disparity label of the left-image** depth map.
   (c) **Minimum disparity label of the right-image** depth map.
4. **Classify pixels** into reliable, occlusion, error and texture-less (OE Module)
5. Calculate **new data-cost** based on previous classification (OE Module)
6. Iterative **BP** (BP2) only **over non-reliable pixels**
7. **Output**: for each pixel, send to the output:
- **Minimum disparity label of the left** depth map (final result).

The aim of BP1 is to provide the OE module with enough information to classify the image pixels. This classification can be obtained with a relatively low number of iterations, as convergence pixels and occluded ones are rapidly located. After the pixel classification has been obtained, BP2 generates the final depth map with a reduced number of iterations, due to the extra information provided by BP1. Moreover, it also saves memory traffic, performing message calculation only on non-reliable pixels (about 20% of the pixels). It might seem that the complexity and memory bandwidth requirements of the proposed technique could double classical BP (there are 2 BP blocks, steps 2 and 6). However, the BP1 and BP2 blocks can be implemented in the same hardware module (they have exactly the same architecture) and the total number of memory accesses is reduced with respect to classical BP because RT-HD BD needs fewer iterations to converge. In classical BP, the number of iterations is constant, but in RT-HD BP it changes depending on the level and it is reduced in the last and most computationally expensive steps. This reduction is a consequence of two advantages of the proposal. First of all, BP1 makes use of an empirical observation: most of the pixels that converge to correct values will normally do so in a low number of iterations. Thus, the number of iterations of the BP1 block can be very small. Secondly, after the pixel classification, the pixel data cost depends of the pixel type, improving BP2 convergence. Additionally, BP2 only calculates messages on non-reliable pixels, reducing the number of iterations. Both contributions reduce the number of iterations and memory accesses.

2.2. Pixel classification techniques

When a pixel has converged in the BP algorithm, the sum of the incoming belief messages (SoIM function) tends to have a linear "V" shape (Fig. 1(a)). This shape is centered on the label index (disparity value). It has been empirically observed that the pixels that converge will normally present a SoIM function with a well-defined "V" shape during the first iterations of the last levels (0,1) in the BP1 module. The rest of the pixels usually present a non-"V" shape or a SoIM function with several local minima (Fig. 1(b)). Based on this observation, we use a simple technique to identify the pixels that probably converge: if the SoIM function has a "V" shape, the first (1M in Fig. 1), second (2M) and third (3M) minimum disparity values will normally be consecutive values. However, if the shape is different, this does not usually occur (Fig. 1(b)). This simple observation normally produces good results with a very low computational effort. The pixel whose SoIM function has a "V" shape will be classified as a reliable pixel and the rest are classified as potential error pixels.

The OE module generates a two-bit per pixel map that classifies the pixels into four categories: reliable, potential error, occluded and texture-less pixels:

(1) Occluded: generates the occlusion map using a cross-checking technique based on Egnal and Wildes (2002). Since the left and right images are viewing roughly the same scene, the horizontal disparity images derived from matching right-to-left and left-to-right should be negatives of each other. The hypothesis is that the points where the two images are not negatives of each other are occluded.
(2) Texture-less: we observe differences between the first ten minimum values on the fly. When the medium difference is below an experimental constant it is a texture-less pixel.
(3) Reliable pixels: SoIM function with a "V" shape.
(4) Potential error: Pixel which is not (1) or (2) and whose SoIM function does not have a "V" shape.

2.3. Message compression

BP2 limits the message calculation to non-reliable pixels, as reliable pixels have already converged. Performing message passing on non-reliable pixels reduces the memory traffic by about