Memory efficient belief propagation for high-definition real-time stereo matching systems

J. Pérez, P. Sánchez, M. Martínez

J. Pérez, P. Sánchez, M. Martínez, "Memory efficient belief propagation for high-definition real-time stereo matching systems," Proc. SPIE 7526, Three-Dimensional Image Processing (3DIP) and Applications, 75260N (4 February 2010); doi: 10.1117/12.838352

SPIE.

Event: IS&T/SPIE Electronic Imaging, 2010, San Jose, California, United States

Memory-efficient Belief Propagation for High-Definition Real-Time Stereo Matching systems

J. Pérez[a], P. Sánchez[a], M. Martínez[b]
[a]University of Cantabria, Av/Los Castros S/N, Santander, Cantabria, Spain 39005;
[b]DS2, Charles Robert Darwin 2 Parc Tecnológic, Paterna, Valencia, Spain 46980;

ABSTRACT

Tele-presence systems will enable participants to feel like they are physically together. In order to improve this feeling, these systems are starting to include depth estimation capabilities. A typical requirement for these systems includes high definition, good quality results and low latency.
Benchmarks demonstrate that stereo-matching algorithms using Belief Propagation (BP) produce the best results.
The execution time of the BP algorithm in a CPU cannot satisfy real-time requirements with high-definition images. GPU-based implementations of BP algorithms are only able to work in real-time with small-medium size images because the traffic with memory limits their applicability.
The inherent parallelism of the BP algorithm makes FPGA-based solutions a good choice. However, even though the memory traffic of a commercial FPGA-based ASIC-prototyping board is high, it is still not enough to comply with real-time, high definition and good immersive feeling requirements.
The work presented estimates depth maps in less than 40 milliseconds for high-definition images at 30fps with 80 disparity levels. The proposed double BP topology and the new data-cost estimation improve the overall classical BP performance while they reduce the memory traffic by about 21%. Moreover, the adaptive message compression method and message distribution in memory reduce the number of memory accesses by more than 70% with an almost negligible loss of performance. The total memory traffic reduction is about 90%, demonstrating sufficient quality to be classified within the first 40 positions in the Middlebury ranking.

Keywords: Stereo-vision, Belief propagation, High-Definition, Real-Time, FPGA

1. INTRODUCTION

In current telecommunication systems, the participants normally do not have the feeling of being physically together in one place. In order to improve the immersive face-to-face experience, tele-presence systems are starting to include 3D video and depth estimation capabilities. There are two main 3D systems used in tele-presence: the first of them is aimed at obtaining the Visual Hull of an object or scene, while the second one is attempts to obtain the depth of the different objects inside the scene. This work is focus on the second kind of systems.
A typical requirement for these depth estimation systems [1] includes high definition (at least 1280x720 pixels), good immersive feeling (more than 80 disparity levels) and low latency (depth estimation in less than 40 milliseconds).
We can classify depth estimation stereo matching techniques into two main categories: global and local, the former obtaining the best results. There are several global algorithms, but stereo matching using Belief Propagation (BP) is one of the most effective depth estimation techniques, covering the first positions in the Middlebury rankings. Most of the work using BP is based on the global approach presented in [3], because it converges faster and reduces memory requirements. However, the execution time of this algorithm in a CPU cannot satisfy real-time (RT) requirements with high-definition (HD) images. Other works [4] are focused on local or semi-global methods. They reduce the execution time, but they normally lose performance. There are some BP algorithms that have been implemented in GPUs although they have limited performance, working with low-resolution images and a small number of disparity levels [5][6]. Finally, several FPGA-based implementations of BP algorithms have been proposed. In [7], an approach that works with low-resolution images and 16 depth levels is proposed. In [2], a RT architecture is presented. However, they work with only 16 disparity levels and a phase-based depth estimation algorithm, which performs worse than BP-based algorithms.

This work has been partially supported by the CDTI under project CENIT-VISION 2007-1007 and the CICYT under TEC2008-04107.

Three-Dimensional Image Processing (3DIP) and Applications, edited by Atilla M. Baskurt,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7526, 75260N · © 2010 SPIE-IS&T
CCC code: 0277-786X/10/$18 · doi: 10.1117/12.838352

A recent publication on FPGA [20] is also focused on implementing BP-based stereo matching on RT. However, our proposal outperforms [20] in three key aspects: first, it performs 1843.2 million Disparity estimations per second (obtained as "width*height*fps*Disparity_labels"), which is three time faster than [20]. Secondly, its results are similar to those of the BP-M algorithm, which shows poorer results than our proposal. Finally, our proposal can be implemented in a FPGA, while the one in [20] is an ASIC (very expensive and ad-hoc solution).

With recent hardware advances, memory bandwidth has become a more performance-limiting factor than the total number of algorithm operations. To confront this problem, the image is split into several unconnected regions in [8]. The main drawback for RT applications is that the size of these regions is normally very small and this greatly reduces performance.

Here, we perform an analysis of the different methods presented in the literature that can be used to reduce the memory traffic. For the first time, they are tested and compared in real-time and high-definition environments. As we demonstrate that they are not able to reach the specifications, we present a novel stereo matching algorithm based on BP. It includes occlusion, potential error and texture-less region handling. Several techniques have been used in stereo matching for occlusion handling [9]. A simple method of detecting occlusion is the cross-checking technique [10]. Other occlusion-handling approaches generate better results [11] but they double the computational complexity. Some other techniques have improved depth estimation in texture-less areas [12]. However, they work with low-resolution images, 48 disparity levels and they do not satisfy RT requirements. Other approaches try to reduce potential errors [13], but they work with medium-resolution images, 14.7fps and 40 disparity levels.

In this paper, we propose a global approach based on a double serial BP. A recently presented work [14] also uses a two-step depth estimation algorithm, although with a local approach. Moreover, it does not comply with RT and HD requirements. Some proposals [15] use several BP modules and show better performance than ours. However, the time they take to obtain a small image disparity map is 250 times the time we use to obtain a HD disparity map. On the other hand, some proposals have concentrated on reducing the number of messages in the BP [16][17] or on compressing the messages to reduce memory [18]. However, they are not able to meet HD and RT constraints or to obtain good results.

We analyze and compare some of the main techniques, which have been presented up to now, aimed at reducing memory traffic in order to demonstrate they are incapable of reaching real-time and high definition requirements.

The system described here presents a BP architecture that complies with actual tele-presence system requirements [1]. The proposal includes two main contributions:

1. It splits the algorithm into two BPs that work serially. Between the two blocks, a new data-cost is calculated based on a pixel classification. This classification identifies occlusions, potential-error, texture-less and reliable pixels. This contribution improves the single BP results while reducing the number of memory accesses for HD and RT systems (250 times faster than [15]).
2. It defines an adaptive message compression technique to reduce memory traffic with little performance penalty. It provides better balance between performance, simplicity and implementation than [16][17][18]. Moreover, [18] shows some limitations: the message compression used in [18] is not linear, which means it has to uncompress, operate and compress again. In contrast, our proposal operates with compressed messages. Moreover, as was pointed out in [20], in [18] they assume data to be stored with floating point precision, but if the data precision is 8-bit, only 30-50% compression rate can be achieved. Our proposal achieves more than 70%.

The remainder of this paper is organized as follows. In Section 2, we comment on the requirements of the tele-presence system. In Section 3, we apply several existing methods to our system, in order to test whether they could fulfill the requirements. In Section 4, we discuss the double BP with occlusion, error and texture-less handling methods, as well as the compression technique used to meet the memory access requirements. Finally, we present the experimental results and conclusions in Sections 5and 6.

2. SYSTEM REQUIREMENTS

The tele-presence system, which is developed in [1] must satisfy the following constraints:

1. Real-time system with low latency: the depth-estimation processing time is limited to 40 milliseconds. This requirement is essential to provide presence feeling. It allows 25 frames per second video.
2. High resolution: the image size is 1280x720 pixels, although VGA format is supported, combined with higher frame rates.
3. Immersion feeling: in order to obtain a life-like 3D model, at least 80 disparity levels seem to be needed. Additionally, a high-quality depth-estimation algorithm (i.e. Belief Propagation) is necessary. This translates into a high number of different depth levels, enough to make the user feel the 3D experience in a satisfactory way.

4. Memory bandwidth of the hardware platform: an actual high-performance platform (for example, a commercial FPGA-based ASIC-prototyping board) has a limited maximum external-memory bandwidth (about 153 Gb/sec in the case of the paper reference platform [19]). This is the greatest limitation in fulfilling the previously commented restrictions as we will see later.

As far as the authors know, there are no previous works that can satisfy all these requirements.

In order to use reference [3]'s algorithm in a real system, several parameters have to be defined:

a. In this work we have assumed that the minimum number of iterations and levels needed to cover the Section requirements is 7. We reached this conclusion through subjective tests developed in our tele-presence room. In the VISION project [1], some psychologists are part of the team and their main task is to evaluate the impact of the 3D quality for the tele-presence user. They observed significant differences going from six levels/iterations to seven, but negligible impact going from seven to eight.

b. The algorithm variables are quantified using 16 bits and the number of disparity levels is set to 80. The first value was enough to obtain results with less than 1% of error in comparison with original C code. Moreover, it is inferior to 18 bits which allows every multiplication be carried out in a single FPGA DSP48. Finally, as we are using 64-bit width DDR2 external memory, quantifying with 16 bits enables the storage of 4 entire data within a single DDR2 word. The number of disparity levels was again subjectively evaluated by the psychologist team.

c. The linear truncated model [3] was chosen for the messages as it presents a good balance between edge information and noise information.

With these parameters the BP-based technique presented in [3] satisfies the quality constraint (point 3), despite edge error, occlusions and texture-less region flaws. However, it cannot satisfy the RT and memory bandwidth restrictions (points 1 and 4). This algorithm will be referred to as classical BP.

The parameter that limits the processing time is the number of external memory accesses. The actual high-performance platform, which is used as the hardware reference model in this work [19], could support up to 6 DDR2-400 memories with 64 bits per memory data bus. The maximum number of memory accesses that a depth estimation algorithm can perform in this platform is about 384 million. Two parameters have been taken into account to obtain this limit: the algorithm variables are quantified with 16 bits and the estimation time is less than 40 milliseconds. However, if the classical BP algorithm is analyzed with a, b and c parameters, the total number of required memory accesses will be 2881 million. Thus, the system is far from being implementable in an actual high-performance platform and it would require a reduction in the number of accesses of almost 90%.

In Section 3 we adapt several proposals to our requirements. We analyze whether they are good enough to fulfill constraints 1-4.

3. ALTERNATIVES TO REDUCE MEMORY TRAFFIC

We have just seen that a high reduction in the number of memory accesses is required. In this Section, we adapt and apply some of the recently presented alternatives to our high-definition and real-time system to see whether they are able to reach the 90% reduction requirement in memory traffic with enough quality or not.

We classify the methods to reduce the number of memory accesses in two categories:

1. On the one hand, methods to reduce the number of messages.
2. On the other hand, methods to reduce the number of disparity labels to be stored for each message (or at least some of them).

We define the first kind of method as inter-pixel and the second as intra-pixel. Most of the works published aim their efforts at the first kind of method. We will show that the performance when using the proposed intra-message algorithm outperforms the inter-pixel methods.

3.1 Inter-pixel methods

3.1.1 Adaptive Belief Propagation (ABP)

The main idea of this method is to detect the messages that have converged. These messages will not be updated again. To detect these messages, for each message we compare the minimum disparity label (MDL) in a given iteration with the MDL obtained in the previous iteration where the message was updated. Due to the use of the bipartite grid technique, each pixel is computed in either odd or even iterations but never in consecutive iterations.
This process starts after a certain level. The reason is that in early levels messages are not reliable. Therefore, a coincidence in the MDL does not mean a converged message. Empirically, we choose to apply the method to the last 3 levels. Given that we are using the multi-grid technique, the levels which cause a greater number of memory accesses are the final ones.
The pseudo-code for adaptive belief propagation is:

1. Compute levels 6 to 3 normally. As defined in [3].
2. For levels 2 to 0, for each iteration 'i', pixel (x,y) and message m(x,y):
   2.1. If m(x,y) has been marked as converged:
      2.1.1. Do nothing
   2.2. Else:
      2.2.1. Compare m(x,y) minimum disparity label (mdli) with the one obtained in iteration 'i-2' (mdli-2).
         2.2.1.1. If mdli ==mdli-2: we mark m(x,y) as converged
         2.2.1.2. Else: do nothing

When a message has converged we save two accesses (read and write) to memory (step 2.1). When it has not converged there are no memory savings (step 2.2).
Using this algorithm, the percentage of memory accesses saved depends on the convergence of the disparity map.
It is also possible to establish a certain convergence range. A message can be marked as converged when its MDL differs by a certain quantity in two consecutive iterations. Increasing the range saves more memory accesses, but introduces more potential errors.

3.1.2 Quad-tree preprocessing (QTP)

This method is based on the fact that an image can be faithfully represented by a subset of its pixels: the non-uniform samples. We apply the matlab quad-tree algorithm to obtain a sparse image of the left image that will be used as a mask for the BP algorithm. The sparse image will have non-null values in non-uniform pixels [17] and null values in the rest of the image. When running the BP algorithm, we perform the message passing algorithm only in non-null pixels. The pseudo-code for belief propagation with quad-tree preprocessing is:

1. Apply quad-tree algorithm to left image.
   1.1. Store a mask value:
      1.1.1. A '1' when a pixel is non-uniform.
      1.1.2. A '0' for the rest.
2. Perform BP message passing algorithm only over pixels marked with '1'.
3. Perform bilinear interpolation over sparse disparity map to obtain final dense disparity map.

The percentage of non-uniform pixels depends on the image itself and the threshold applied.
Using this algorithm, the percentage of memory accesses saved depends on the number of non-uniform pixels. A lower threshold implies more non-uniform pixels, better quality and less memory savings. On the contrary, a higher threshold implies bigger quad-elements, greater memory access savings but worse performance.

3.1.3 Foreground segmentation pre-processing (FGSP)

It uses a pre-processing module described in [1] to calculate silhouettes and evaluate BP over these silhouettes. With this method, zones outside silhouettes have no disparity information. Therefore, this method is not suitable for stereo vision applications that require full image depth mapping. Moreover, the preprocessing increases the overall latency.
We apply a foreground segmentation algorithm to the left image, obtaining the silhouettes. Then, we store a mask indicating whether a pixel is part of the foreground or the background in FPGA block RAM memory. Finally, the BP algorithm is applied over the foreground pixels. The foreground extraction can be developed in 10 milliseconds, 30 remaining for the BP algorithm. This increases the number of memory accesses to be avoided in the BP algorithm from 90% to 93.34%.
The pseudo-code for belief propagation with quad-tree preprocessing is:

0. Pre-process room disparities (depth map).
1. Apply foreground-segmentation to left image and store mask value:
   1.1. Store a '1' when the pixel is inside a Silhouette (Foreground).
   1.2. Store a '0' when the pixel is outside (Background).
2. Perform BP over foreground pixels.
3. Obtain final depth map by appending foreground to background disparities.

Using this algorithm, the percentage of memory accesses saved depends on the size of the silhouettes. Experimentally, it has been calculated to be around 80% for the tele-presence application when two people are in the room filling the field of view, which is more than 13% below the required 93%. The more people (i.e. silhouettes) are in the room, the less memory reduction this algorithm achieves. Therefore, this method could only be applied in combination with some other one to achieve the memory traffic reduction required for real-time and high-definition.

3.2 Intra-pixel methods

3.2.1 Message compression (MC)

We use the truncated model [3] for the BP algorithm. Using this model, reliable messages tend to have a 'V' shape around the minimum disparity label with truncated values on both sides of it.
Bearing in mind that structure, we propose a technique with some similarities to the Envelope Point Transform proposed in [18], but simpler and more precise when parameters are correctly chosen. Instead of storing only the envelope points [18], the proposed technique stores all the points inside a region around the minimum disparity label. It has two main advantages with respect to [18]. Firstly, we do not need to uncompress the message prior to operating with it. Secondly, the compression rate drastically decreases when using EPG with limited precision. In contrast, we achieve more than 70% even with fixed point variables. The number of stored points is a function of several parameters (adaptive approach): iteration, level and pixel type. This can reduce the compression factor, but increase the performance and reduce the quality penalty. We store 3 parameters and a group of labels instead of the 80 disparity labels, as can be seen in Figure 1.

1. The **offset** (OF): first disparity label of the selected region.
2. **Number of disparity labels** (NV) of the selected region.
3. **Information values**: we store all the values of the selected region (from disparity label OF to OF+NV).
4. **Truncating value** (TV): value that is assigned to all the disparity labels that are not included in the selected region.

It is important to notice that this kind of compression is lossless when the model is a perfectly truncated model.
The percentage of memory accesses saved depends on the compression. A greater compression causes more errors, but provides bigger savings in memory accesses.

It has been empirically demonstrated that this method reaches compressions up to 80% without significant losses in performance. On the other hand, when the number of points (NV) is further reduced to reach 90%, the performance is quickly degraded. Therefore, as we will see in Section 4, an extra effort is required to reach the 90% of memory traffic savings needed.

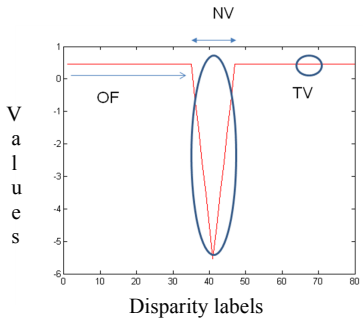

Figure 1. Parameters for message compression

3.3 Performance comparison for Section 3 proposals

We have applied all the proposed methods to a real tele-presence system. However, since the Middlebury test has become a reference in stereo vision articles, we present the results for a Middlebury test: the teddy bear. When we consider toys in the teddy bear test to be foreground (i.e people in the room), the results obtained are quite similar to those presented in real tele-presence systems.

We have seen in Section 2 that a reduction of around 90% in memory accesses is required to fulfill system RT constraints. However, due to the different kind of memory access reduction techniques used in each method, it is difficult to establish a fixed value. Therefore, we have chosen the parameters for the different methods to obtain a reduction between 70 and 80% in the total number of accesses to memory.

In Figures 2a-e we show the disparity maps of the algorithm without modifications and the ones obtained by using the aforementioned methods. The results are quantified in Table 1, in terms of percentage of errors compared to Classical BP and percentage of memory accesses saved. An extra row is also included with noteworthy characteristics, when present.

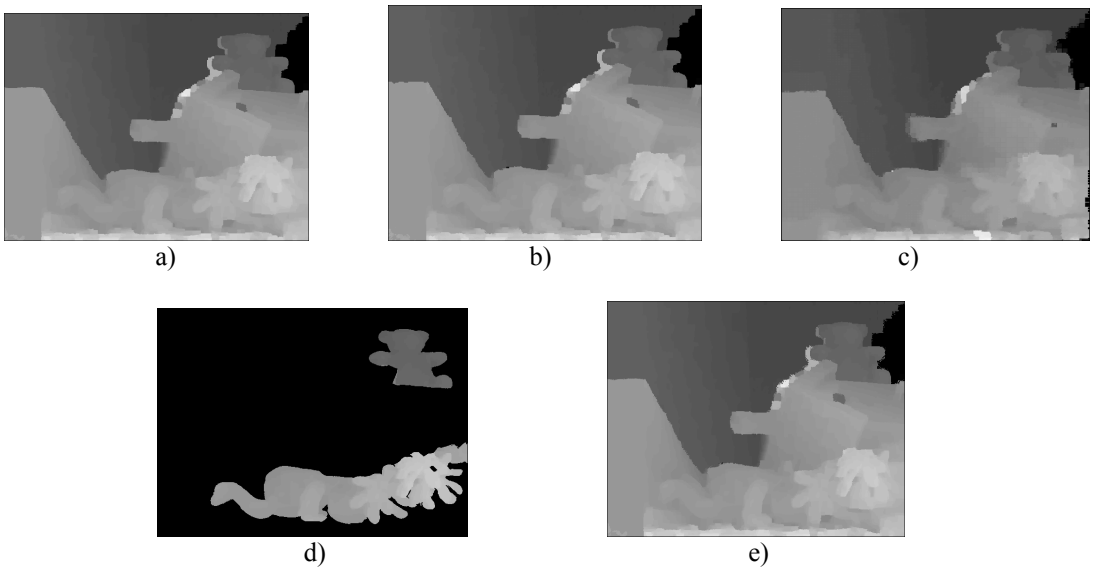

Figure 2. a Disparity maps: a)Classical BP. b)ABP. c)QTP. d)FGSP. e)MC.

As seen in Table 1, the method that has a better tradeoff between error and MS is FGSP. However, due to the 10 millisecond preprocessing, the MS required is more than 93%. Moreover, as the MS is limited by Silhouette size, the method cannot be applied.

The rest of the methods based on Inter-Pixel are not able to reach 80% of MS without important performance degradation, especially those based on Quad-tree preprocessing.

Table 1. Percentage of errors and memory accesses avoided for the system described

Method	Inter-Pixel			Intra-Pixel	Classical
	ABP	QTP	FGSP	MC	BP
%Error compared with Classical BP**	+1.7	+3.4	-4 *	<1	0
%Memory accesses	30	29	19	18	100
Characteristics	--	Fast degradation	%memory saving limited by Silhouette size. Disparity map only on Silhouette.	Flexible. Small error with big memory savings	--

*The errors have been computed only over the Silhouette, causing a reduction in the number of errors because most of them are on the borders.

**The percentage of errors is measured comparing with the original BP algorithm result. Obviously, when the original BP has errors (for example at occlusions), this measure is not reliable. However, this measure is representative enough to evaluate the methods, as they do not improve performance in such error regions.

Therefore, the best results are obtained using the proposed method based on Message Compression, with an almost negligible error increase for a MS of 82%.
Anyway, none of the methods presented up to now can reach 90% of memory traffic reduction without important performance loss. We propose in Section 4 a new method based on two serially developed BP modules which is able to reach the 90% of memory traffic reduction, even improving the final performance.

4. PROPOSED SYSTEM ARCHITECTURE

We have seen that classical approaches are not sufficient to reduce the memory traffic enough to reach real-time high-definition characteristics. Even with the proposed message compression system, the reduction in memory traffic without important performance degradation cannot go further than 80%. Moreover, the flaws in the original algorithm, i.e. occlusions, texture-less regions and edge errors, still remain in the Section 3 methods (and are even increased).

In this Section we propose a novel architecture aimed at reaching the 90% memory traffic reduction while reducing the errors mentioned.

In order to handle occlusions, potential errors and texture-less regions that degrade the performance of the classical approach, the proposal is to split the BP algorithm into two separate BP blocks. Between them, a new module (Occlusion, Error and texture-less handling module, OE) classifies the pixels into four categories. Additionally, this module will recalculate the values of the cost function taking into account the pixel category. 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:
   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. A very important advantage of the proposed technique is that this classification can be obtained with a relatively low number of iterations. After the pixel classification has been obtained, the second BP (BP2) generates the final depth map with a reduced number of iterations. Moreover, it also saves memory traffic, performing message passing 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 the 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, as they have exactly the same architecture. Moreover, the total number of memory accesses is reduced with respect to classical BP. In table I, the number of iterations for each level in RT-HD and classical BP are presented. In classical BP, the number of iterations is constant, but in RT-HD BP it changes with the level. Table I shows the total number of BP1 and BP2 iterations per level.

Even though the number of iterations is higher in the first levels (6 to 3), the algorithm reduces the iterations in the last level and this minimizes the total number of memory accesses: the classical BP algorithm needs 9.33x accesses while the RT-HD BD needs only 7.47x accesses (19.89% less memory traffic). The parameter 'x' is a function of the image size and disparity levels. It can be expressed as:

$$x = \text{Read\_Comp\_Data} + \text{Read\_Msg} + \text{Write\_Msg} =$$
$$\left(\frac{\text{Width} * \text{Height} * \text{Disp}}{4}\right) + \left(\frac{\frac{\text{Width} * \text{Height}}{2} * \text{Disp\_levels} * 4}{4}\right) + \left(\frac{\frac{\text{Width} * \text{Height}}{2} * \text{Disp\_levels} * 4}{4}\right)$$

Table 2. Relation between memory accesses and levels.

Level		0	1	2	3	4	5	6
Classical BP iterations		7	7	7	7	7	7	7
RT-HD BP iterations	BP1	2	3	3	7	7	7	7
	BP2	3	4	4	7	7	7	7
Memory accesses per iteration		x	x/4	x/16	x/64	x/264	x/1024	x/4096

The reduction in the number of iterations in the most computationally expensive step 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 it in a low number of iterations. Thus, the number of iterations of the BP1 block can be very little. Secondly, after the pixel classification, the pixel data cost depends of the pixel type and this improves BP2 convergence. Additionally, BP2 only performs message passing on non-reliable pixels, reducing the number of iterations. Both contributions reduce the number of iterations and memory accesses but their computational impact is very limited.

Occlusion, edge error and texture-less area handling

In the RT-HD BP algorithm, the pixels are classified into 4 categories in the OE module: occluded, potential error, texture-less and reliable pixels.

The OE module generates the occlusion map using a cross-checking technique based on [10]. The module also detects low-textured areas by observing differences between the first ten minimum values on the fly. When the medium difference is below an experimental constant, the pixel is classified as texture-less.

In BP, the disparity value for a given pixel is the label index that minimizes the sum of incoming messages and data-cost. 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 (Figure 3.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, while the rest of the pixels normally present a non-"V" shape or a SoIM function with several local minima (Figure 3.b).

Based on this observation, the proposed algorithm includes a simple technique to identify the pixels that probably converge. It is based on the comparison between the disparity label of the first and the third minimum. If the SoIM function has a "V" shape, the first (1M in Figure 3), second (2M) and third (3M) minimum disparity values will normally be consecutive values. However, if the shape is different, the third value will not normally be a consecutive value (Figure 3.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. As occluded and texture-less pixels have previously been identified, the pixels that are classified as reliable have a high probability of having converged to the correct value.

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

New data cost

This module uses the information provided by OE to calculate new data costs as:

1. Reliable pixels: data cost defined as 0 for their minimum disparity label and a pre-defined penalty, equal to the maximum truncated value, for the other labels.
2. Texture-less pixels: The data cost is 0 for all the disparity labels (unknown). This helps texture-less pixels to obtain correct disparity values.
3. Error pixels: they keep their data cost.
4. Occluded pixels: take the value of the first non-occluded pixel on their left.

BP2 limits the message passing to non-reliable pixels, reducing the memory traffic. The total memory reduction is about 21%. This reduction is still far from the required 90%. To reach this limit, the message compression technique defined in 3.1.4. is applied.

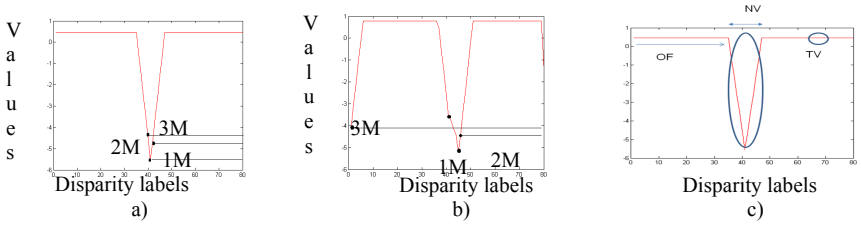

Figure 3. a)Reliable pixel b) Possible error pixel c) Parameters for message compression.

Adaptive message compression

We have seen that the memory traffic reduction obtained with the two-BP architecture is far from the 90% required. In order to reach this 90% we apply the message compression described in 3.2.1. This adaptive technique is applied only to the BP2 block reducing the memory traffic by about 70%, which is below the limit over which we can have important performance loss (see 3.2.1.). This combined with the 20 % saved with the double BP topology lead us to a memory traffic saving of a 90%, enough to fulfill the real-time high-definition requirements.

The pixels that converge will normally present a shape that is easily and efficiently compressed with the proposed techniques. This property, combined with the pixel classification that the OE generates, guarantees a good compression factor, allowing the fulfillment of the system requirements.

5. RESULTS

In order to validate the proposed algorithms, several video sequences [21] have been evaluated with the classical and the RT-HD BP methods. In Figure 4, we show the disparity maps that are obtained with classical BP (a), the proposed RT-HD BP without compression (b) and the RT-HD BP with enough compression to comply with the RT restrictions in Section 2 (c). Some occlusions have been mitigated, some errors corrected and some texture-less zones have been filled

in (b, c).

a) b) c)

Figure 4. Disparity maps: a)Classical BP b) and c)RT-HD BP without and with compression.

The RT-HD BP shows an improvement of more than 6% when compared to classical BP. At the same time, it satisfies section 2's RT and HD requirements. The memory reduction of the RT-HD BP with compression is about 90%. To finish this results section, we provide Middlebury test results for our proposal in Figure 5 and Table II, comparing the proposal with RT publications ranking Middlebury tests. For convenience, we maintain the names used in the Middlebury test in Table II. As can be derived from Table II, our proposal is the only ranking in the test that is able to achieve true RT for HD images. The only one whose latency is close to RT (RealtimeBP) works with small images. Moreover, among all of the proposals focus on real time, there is only one whose position in the ranking is significantly better than our proposal (PlaneFitBP) but working at 1fps, which is not RT at all.

AdaptPolygon [43]	34.4	2.12	42	2.69	37	8.27	39	0.39	32	0.64	28	2.99	23	7.00	40	10.0	32	13.1	37	3.42	33	10.0	35	9.06	35	5.81
RealTimeGPU [14]	35.2	1.34	33	3.27	41	7.17	35	1.02	41	1.90	42	12.4	47	3.90	15	8.65	26	10.4	17	4.37	42	10.8	39	12.3	44	6.46
YOUR METHOD	35.2	0.91	10	2.39	33	4.63	9	0.66	36	1.37	35	8.71	43	6.45	36	10.9	40	15.9	44	6.14	48	13.6	45	12.3	43	7.00
TensorVoting [9]	35.3	1.20	25	2.18	31	5.85	23	0.68	38	1.18	34	6.69	38	7.21	44	14.4	46	17.5	49	3.12	27	9.78	32	9.20	36	6.58
GenModel [20]	37.9	2.35	44	4.50	46	12.2	47	1.11	45	2.20	46	10.4	45	3.88	14	11.0	41	11.9	31	3.07	26	12.8	44	8.10	26	6.96
ReliabilityDP [13]	39.8	1.21	28	3.18	40	6.49	30	1.05	42	2.03	44	8.27	42	5.17	29	10.7	38	11.8	29	9.05	55	16.0	51	14.3	49	7.44
BP+MLH [40]	40.0	1.62	36	3.65	43	8.41	40	0.66	37	1.87	40	9.02	44	6.95	39	15.5	48	15.8	43	3.39	32	13.7	46	8.52	32	7.43

Figure 5. Middlebury ranking for our proposal.

Table 3. Proposals focused on real-time, in Middlebury ranking, comparison.

Parameter	Latency(msec)	Image resolution	Disp. Lev.	Rank.
Proposed RT-HD BP	40	1280x720	80	35.2
RealTime GPU	183	640x480	48	35.2
RTCensus	--	--	--	45.6
Realtime BP	62.5*	320x240	16	30.5
FastAggreg	600	450x675	60	32.7
PlaneFit BP	1000*	512x384	48	12.8

*Latency has been extrapolated from the fps data (18fps≈62.5millisecond and 1fps≈1000 milliseconds)

6. CONCLUSIONS

In this work we have presented a Real-Time High-Definition depth estimation algorithm based on Belief Propagation. We have tested currently-existing methods to reduce memory traffic, adapting them to our requirements of real-time and high-definition. We have demonstrated that they are insufficient to fulfill the requirements without important performance loss. We have also proposed a message compression method which performs better than current methods but is still insufficient.

Finally, we have proposed a new method that estimates depth maps in less than 40 milliseconds for HD images (1280x720 pixels at 30fps) with 80 disparity values. The work exploits the proposed double BP topology and it handles occlusions, potential errors and texture-less regions to improve the overall performance by more than 6% (compared with classical BP) while it reduces the memory traffic by about 21%. Moreover, the adaptive message compression method allows the system to satisfy Section-2's real-time and low execution latency requirements, reducing the number

of memory accesses by more than 70% with an almost negligible loss of performance (less than 0.5%). The total memory traffic reduction is about 90% with a 6.0% performance improvement (compared with classical BP).

REFERENCES

1. Vision project http://vision.tid.es . 2009
2. J. Diaz, E. Ros, R. Carrillo, A. Prieto, "Real-Time System for High-Image Resolution Disparity Estimation," in IEEE TIP07
3. P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," in IEEE CVPR04
4. H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," in IEEE TPAMI08.
5. Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, "Real-time Global Stereo Matching Using Hierarchical Belief Propagation," in BMCV06.
6. L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," in IEEE 3DPVT06.
7. S. Park and H. Jeong, "A fast and parallel belief computation structure for stereo matching," in IASTED IMSA07.
8. Y. Tseng, N. Chang and T. Chang, "Low Memory Cost Block-Based Belief Propagation for Stereo Correspondence," in IEEE ICME07.
9. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in IEEE ICCV01.
10. G. Egnal and R. Wildes, "Detecting binocular half occlusions: empirical comparisons of five approaches," in IEEE PAMI02.
11. J. Sun, Y. Li, S. Kang, and H. Shum, "Symmetric stereo matching for occlusion handling," in IEEE CVPR05.
12. Q. Yang, C. Engels and A. Akbarzadeh , "Near Real-time Stereo for Weakly-Textured Scenes,", in BMCV08.
13. M. Gong and R. Yang, "Image-gradient-guided real-time stereo on graphics hardware," in IEEE 3DIM05.
14. Z. Yilei, G. Minglun, Y. Yee-Hong, "Local stereo matching with 3D adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy," in IEEE ICPR08.
15. Q. Yang, L. Wang, R. Yang, H. Stewenius and D. Nister, "Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling," in IEEE TPAML09.
16. S. Huq, A. Koschan, B. Abidi, and M. Abidi, "Efficient BP Stereo with Automatic Parameter Estimation," in IEEE ICIP08.
17. M. Sarkis, Michel Diepold, Klaus, "Sparse stereo matching using belief propagation," in IEEE ICIP08.
18. T. Yu R. Lin Super, B. Bei Tang, "Efficient Message Representations for Belief Propagation," in IEEE ICCV07.
19. www.synplicty.com/products/haps/haps-52.html . 2009.
20. C. Liang, C. Cheng, Y. Lai, L. Chen, H. Chen, "Hardware-Efficient Belief Propagation," in IEEE CVPR09.
21. Feldmann, M. Mueller, F. Zilly, R. Tanger, K. Mueller, A. Smolic, P. Kauff, T. Wiegand "HHI Test Material for 3D Video", MPEG 2008/M15413, Archans, France, April 2008.