Actas de las XXXVI Jornadas de Automática, 2 - 4 de septiembre de 2015. Bilbao
ISBN 978-84-15914-12-9 © 2015 Comité Español de Automática de la IFAC (CEA-IFAC)

Image compression with Region Of Interest for Underwater Robotic Archaeological Applications

Eduardo M. Rubino, Diego Centelles, Jorge Sales, José V. Martí, Raúl Marín, Pedro J. Sanz

Computer Science and Engineering Department
University of Jaume-I, 12071 Castellón de la Plana (Spain)

{rubino, al132071, salesj, vmarti, rmarin, sanzp}@uji.es

Abstract

The increasing demand for underwater robotic intervention systems around the world in several application domains requires more versatile and inexpensive systems. The use of supervised semi-autonomous robots can help at this task. To achieve this goal, a wireless communication system can provide freedom of movements to the robot and, at the same time, allows the operator to get camera feedback and supervise the intervention. This paper proposes the use of progressive image compression and the use of regions of interest (ROI) for underwater robotic applications, specially when there is limited bandwidth (i.e., wireless underwater RF channels) allowing for a much more agile data exchange between the vehicle and a human operator supervising the underwater intervention. The operator can dynamically decide the quality, frame-rate or resolution of the received images so that the available bandwidth is utilized to its fullest potential and with the required minimum latency.

Keywords: Underwater Robotics Intervention, Image Compression, ROI (Region of Interest), Autonomy Image Transmission, Wireless Communications.

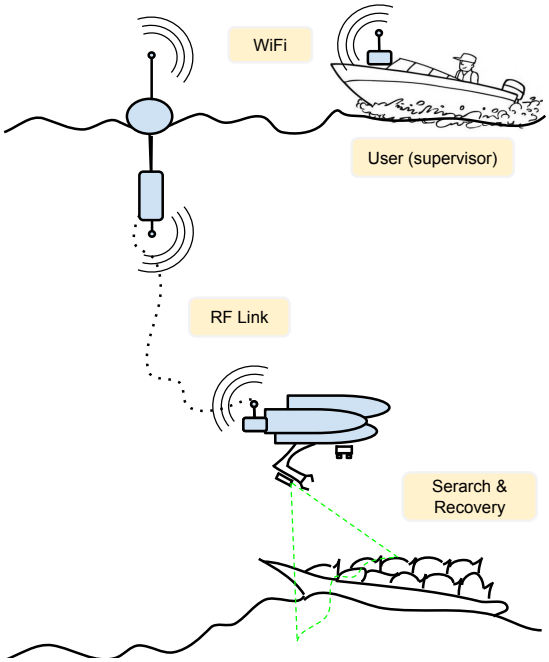

Figure 1: Search and Recovery in the context of archaeology. A wireless RF link provides feedback to the user that is supervising the intervention.

1 Introduction

In the context of the MERBOTS research project, a three-year coordinated project funded by the Spanish government for the period 2015-2017 under grant DPI2014-57746-C3 [1], one of the roles of our research group inside the MERMANIP subproject is to build a wireless communication system that can provide freedom of movements to the underwater robot and, at the same time, to allow the operator to get feedback and supervise the intervention (see Figure 1). The robotic system under development will assist the archaeologists in the detailed work of monitoring, characterization, study, reconstruction and preservation of archaeological sites, always in accordance with the continuous supervision of the human expert.

MERBOTS represents the natural continuation of several national and international research projects in the last years in the field of underwater robotic intervention (i.e. RAUVI [2], TRITON [3], TRIDENT [4]). One of the objectives of the MERMANIP subproject is to provide different communication technologies that can be used to allow the operation of a vehicle without any physical connection to the surface operators, which are supervising and controlling an intervention task. Our research team is in charge of the design of a wireless communication underwater system that is able to transmit telemetry data as well as compressed low-resolution images, allowing the implementation of cooperative intervention missions.

The underwater communications for the proposed system will be implemented over RF channels. Taking into account the low data transfer ratio that can be achieved using RF, advanced image compression techniques must be used. The sys-

ISBN 978-84-15914-12-9

Figure 2: Example of scene with a Region of Interest (ROI).

tem will include a specific communication protocol that will adjust in real time the compression ratio depending on the available bandwidth at each time, thus guaranteeing a minimum quality to allow proper monitoring of the intervention by the expert operator. It is also considered the objective of improving the compression system, allowing the quality adjustment of the scene by Regions of Interest (ROI), when the user requires higher image quality in a specific area (e.g around the manipulator arm). The Fig. 2 shows an example of scene with a ROI.

2 The intervention domain

Robotic applications and, particularly, Autonomous Underwater Vehicles for Intervention (I-AUV) use images from its built-in camera(s) (video) as one of its main sources of data, among others, in order to control its internal algorithms. In a supervised system, these images should reach the operator with the lowest latency and with the highest quality possible so that he can interact with the system and adjust the task execution in a supervised manner. As an example, this kind of control has been experimented in the FP7 TRIDENT project, to perform autonomous visually guided grasping in the sea [5].

Besides this, communications is a crucial subsystem in any robotic application, specially the ones that permit the user to interact remotely with the system. Because of that, image compression and transmission is necessary in order to send the required information with the lowest latency and without compromising the network and the whole system.

Although recent studies demonstrate that, using the most efficient modulation methods, it is possible to transmit video through an underwater channel using acoustic signals [6, 7] and *Blue Light* [8], both acoustic and optical signals are not capable to pass through solid objects that could be in the line of sight of the wireless transceivers. Moreover, the performance of these methods depends heavily on the characteristics of the underwater scenario and the type of the channel. On one hand, acoustic systems are greatly affected due to multi-path if the link is horizontal, and also by the acoustic noise originated by human activity or the noise of the sea waves, animals, etc. The acoustic noise constrains the range of typical frequencies used in acoustic systems between 8 and 155 KHz [9], which makes it very difficult to achieve high data rates. On the other hand, communication methods based on visible light only work fine on very clear waters and are greatly affected by scattering, suffer attenuation by absorption and usually need accurate alignment.

Nevertheless, RF based solutions are not as affected by the typical problems of the acoustic and optical methods, and are much cheaper. Moreover, RF signals can propagate easier from a medium to another, allowing the establishment of a communication link to an underwater transducer from the surface. The main problem of using RF is the high attenuation that it suffers when the waves go through the water. However, different studies [10, 11, 12] indicate that, with the necessary antennas, at lower frequencies, and using the best modulation methods, it is possible to set up a communication link up to several tens of meters through the water.

The application of the most advanced progressive image compression algorithms, as the ones presented in this document, would allow image transmission rates of several frames per second, at the typical latency of the radio-frequency communications. This proposal would overcome the communications challenge in most common underwater robotic interventions.

In the proposed system, a progressive image compression technique and the use of ROI are introduced. At the moment of writing, a compression image server is already implemented and the researchers are working to adapt it to the robot platform.

As can be seen in Fig. 1, the system will need to connect the surface with the intervention vehicle, allowing the access to the network through a wireless Wi-Fi channel at 5 GHz (<10Km), and connecting this device to a second one, which will establish a radio connection with the first one by using two channels, one for sending and one for receiving. Through these channels, the compressed images, the feedback from the sensors and the control commands will be sent to the vehicle.

ISBN 978-84-15914-12-9

3 Image Compression Techniques

The objective of image compression is to reduce its entropy in order to store or transmit it in a more efficient manner. We can also clearly distinguish between lossless and lossy compression. In lossless compression, the decompressed image will be exactly the same as the original image while, in lossy compression, the decompressed image will be an approximation of the original image.

Digital images usually have 3 color components, which means that what we perceive as one color image is (without loss of generality), in fact, composed of a luminance channel (black and white version of the color image) and 2 color difference channels (which can usually be subsampled without much loss).

Progressive image compression is such that it is trivial and very inexpensive in terms of processing power (there is no need to decompress and recompress the image) to supply an image which is either resolution or quality progressive, meaning that the image data can be truncated at any point and we would still get a lower resolution or quality version of the original image (in this sense, lossless streams can become lossy by simple truncation). In the case of color images, we could also prepare the image in such a way that a monochrome version of it could be obtained with the same progressive characteristics as before.

As opposed to video compression, image compression does not address the compression of the high temporal correlation between adjacent frames in a video sequence. Even though the compression is not as good as it is for video, this has the advantage of being able to rapidly adapt to changing conditions in the communications channel as well as increased flexibility in dynamically changing the frame rate and quality parameters, which are of great importance in low latency communications.

It is important to note that, even though the final objective of image compression is reduction in size, many other factors come into play when discussing actual algorithms. For example, if the reduction in size is such that the time taken for compression and transmission is smaller than transmitting the uncompressed image, we may say that the objective of compression is the reduction of the transmission time of the image. Other factors such as latency or delay play a very important role in the design of an image and/or video compression algorithm, as well as random access to past frames, among others.

For example, for the monitoring of security cameras in real time, it is not necessary to have neither high resolution nor high quality but, when searching for details on an image we need all the resolution, quality, and color we can get. Also, while browsing an image sequence database to find a frame that represents a certain moment when an event occurs, there is no need for high quality and resolution and, when this frame is found, it can be retrieved with full quality, resolution, and color, so that it can be analyzed in all its available detail. Such a scheme would allow for much faster searches, specially if these searches are done remotely, by minimizing the bandwidth and processing needs.

Therefore, compression in general and image compression in particular is a very application specific task, with many available trade-offs and many different algorithms that try to maximize (or minimize) some design criteria. Most image compression algorithms are lossy algorithms designed with the sole purpose of minimizing the resulting size of the image with minimal regard to its execution latency and, in most cases, the whole compressed data is necessary in order to be able to decompress it.

Fortunately, there is a class of image compression algorithms that possess many desirable properties simultaneously and which can also be implemented very efficiently, with complexity similar to the JPEG standard in uniprocessors [13], and with less latency in modern multi-core processors by taking advantage of algorithm parallelism. These algorithms possess the following properties:

- lossless/lossy
- quality scalability
- resolution scalability
- color channel scalability
- random access
- ROI (Region Of Interest)

among others (some algorithms may combine many of the above attributes simultaneously).

Scalable compression usually takes advantage of multiresolution signal decomposition, which is natural for dyadic wavelet decomposition but can also be used with DCT [14] or other block transforms by simply rearranging its coefficients. The coefficients are sent either in bitplane order across all frequencies (resolution) or frequency order across all bitplanes (quality), or a combination thereof, achieving the desired scalability. During compression, special markers can be inserted to

ISBN 978-84-15914-12-9

separate the color components and also to introduce futher blocking within the frequency bands to achieve the desired color scalability and random access, respectively. Also, bitplane shifting (scaling) can be used to introduce regions of interest, as long as their positions and shapes are transmitted as side information to the decoder.

A few of the known progressive algorithms are: EZW *Embedded Zerotree Wavelet* [15], SPIHT *Set Partitioning In Hierarchical Trees* [16], SPECK *Set Partitioning Embedded bloCK* [17], and EBCOT *Embedded Block Coding with Optimized Truncation* [18], which is the algorithm used for the JPEG2000 standard [13], among others. It should be noted that there are many extensions and variations for all these algorithms and most of them do not have an optimized implementation readily available. This means that an efficient implementation, preferably parallel, should be developed so that it can be deployed in a real-world scenario in low-power SBC (Single Board Computers) which are commonly used in underwater vehicles.

Embedded image compression is a very efficient way to cope with varying transmission bandwidth problems in hard real time systems, where it would be better to have a low quality version of the current image instead of a high quality version of an old image. The main idea behind using embedded image compression in the current scenario is to group the source of the data (image) with the transmission channel into one manageable whole, increasing the adaptability of the whole system by varying the amount of data transmitted when the channel capacity changes, i.e., increase the image quality when there is bandwidth available and decrease it when there is not, in order to meet a predefined maximum latency or bandwidth. Also, in order to cope with the need of very low latency, the current algorithm has been developed with parallelism in mind, being able to use many threads of execution in order to decrease the latency as much as possible.

There are many ways in which different compression algorithms can be evaluated and compared. For quantifying the error between images, two measures are commonly used. They are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE between an image y_k and its approximation $\hat{y_k}$ is given by:

$$\mathrm{MSE} = \frac{\sum_{k=0}^{N-1} \left(y_k - \hat{y_k}\right)^2}{N}$$

where N is the total number of pixels in each image. The PSNR between two (8 bpp) images, in decibels is given by

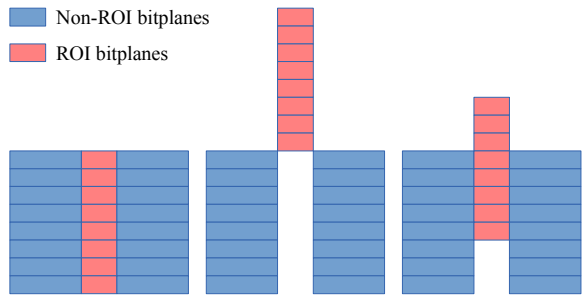

Figure 3: ROI coding methods: (a) Uniform compression, (b) bitplane shifts for the maxshift method, and (c) bitplane shifts for the scaling-based method.

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right)$$

and is used more often since it is a logarithmic measure and the human brain seem to respond logarithmically to changes in intensity. Increasing PSNR means increasing fidelity of compression and, as a rule of thumb, when the PSNR is greater than or equal to 40 dB, it is said that the two images are virtually indistinguishable by human observers.

4 ROI (Region of Interest)

In a low-bandwidth scenario or when the images are highly compressed, there may be circumstances where the image is still not good enough for an operator to distinguish the necessary details. In this case, the use of ROI is an elegant solution to the problem of being able to see the details in part of the image while still maintaining a very high compression level, at the expense of making the other areas in the image less detailed. ROI has been most extensively used in conjunction with medical imaging and are an integral part of the JPEG2000 standard. Most ROI techniques are usually used in conjunction with wavelet based image coding techniques [19].

For Region Of Interest (ROI) processing, there must be a way for the decoder to know which regions were encoded with higher priority than others. A common method known as MAXSHIFT [20] [21] is commonly used so that the bitplanes of the ROI region are encoded in its entirety before any bitplanes of the rest of the image (background) (see Fig.3). This has the advantage of almost no overhead (only the number of extra bitplanes are sent so the decoder knows that after

ISBN 978-84-15914-12-9

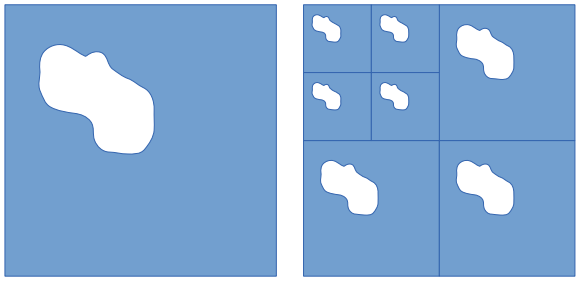

Figure 4: (a) ROI mask, and (b) Wavelet mask for 2 levels of DWT.

reaching this number of bitplanes it should unscale the received coefficients by the amount of bitplanes remaining) but has the disadvantage of having to send the whole ROI, with all its details, before receiving a single bit from the rest of the image.

A more useful method known as SCALING [20] [21] consists of simply shifting the ROI coefficients by a certain number of bits so that they fool the bit allocation algorithm into thinking that they are more important than they actually are and coding them before other coefficients that became smaller due to the scaling (see Fig. 3). In fact, this effectively blends the ROI coefficients with background coefficients which are also important (same order of magnitude) such that the results are seen with good quality in a lower quality background. The main disadvantage of this method is that a ROI map must be sent as extra information to the decoder (overhead), increasing the minimum amount of bits necessary to recover a suitable approximation to the original image.

This map information can consist of object coordinates (rectangles, ellipses, or arbitrary polygons) or, in case of an arbitrary region, a bitmap of the ROI. In this last case, in order to reduce the amount of overhead, this map could be the resulting map on the last decomposition subband (LL) thereby significantly decreasing the bitmap size but having the drawback of using a coarser scale, depending on the number of dyadic decompositions (for an n-level dyadic decomposition the grid would be $2^n X 2^n$ pixels). This bitmap usually consists of a small region and, therefore, is a good candidate for some form of run-length encoding (Fig. 4).

Other methods, which interleave the ROI coefficients with the background coefficients, in a predetermined and alternating order, have also been devised in order to minimize the transmission of overhead information but most of them require fundamental algorithmic changes so that both the encoder and the decoder scan the bitplanes in

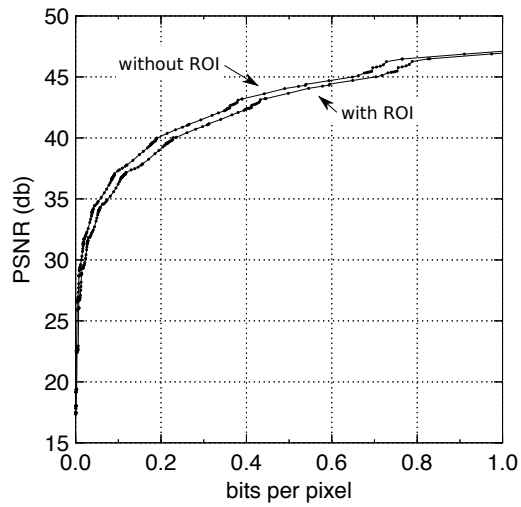

Figure 5: Effect of ROI in PSNR.

the same order and are more complex than the MAXSHIFT and SCALING method.

The examples used in this paper were prepared with the SCALING algorithm with arbitrary coarse regions as described above.

It should be observed that, in general, the use of ROI will impact negatively in the PSNR of the whole reconstructed image but will improve significantly the fidelity in the ROI itself. In our example, Fig. 5 shows the difference for coding the region around the text with 4 bits of shifting in comparison of the coding of the image without any ROI. The image used for this is the monochrome 1024x768 image used in Fig. 6 and the ROI is the rectangle of dimensions 64 x 192 with top left corner at $(x_0, y_0) = (417, 65)$ and bottom right corner at $(x_1, y_1) = (480, 256)$.

5 Implementation

A working implementation has been developed in the C programming language with special vector code for both Intel and ARM processors to speed up the inner loop of the wavelet transform routine.

For the 1024x768 pixel, 8 bpp graylevel image used (lower left image in Fig.6) we ran the compressor in both a Raspberry Pi 2 Model B (RPi2B) and a Beaglebone SBC (Single Board Computer). The RPi2B is based on a 900 MHz quad code ARM processor manufactured by Broadcom (BCM2836 SoC) while the Beaglebone is based on a 720 MHz single core ARM (AM335x 720MHz ARM Cortex-A8) manufactured by Texas Instruments.

The timings for each board for compressing the image on the lower left corner of Fig.6 with and without ROI are displayed on Tables 1 to 4. The

ISBN 978-84-15914-12-9

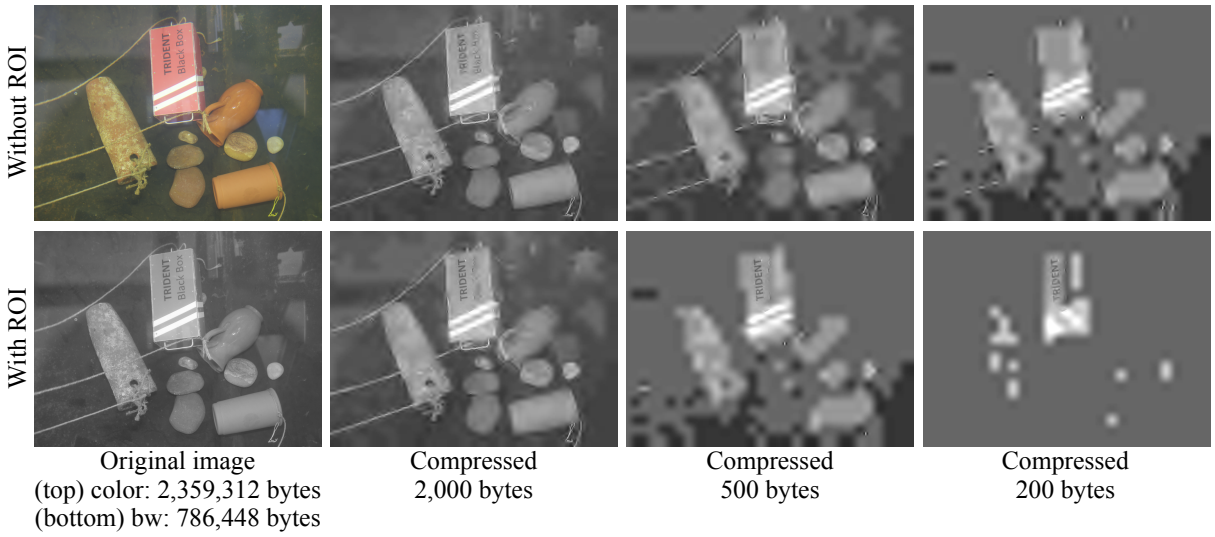

Figure 6: Comparison of compressed image without (top line) and with (bottom line) ROI around the text "TRIDENT" (SCALING with shift=4). The numbers are the actual size in bytes of the compressed image (1024 x 768 pixels)

algorithm has a parameter that specifies either the max size or a "quality" factor (which bears some inverse relation with the PSNR). The normal usage (if lossless compression is not required) is to use a "good" quality parameter for local storage and transmission of any prefix of this file for lower quality versions of the compressed image. The "quality" used for each line of the tables was 0, 4, 8, 12, 16, and 24. The wavelet used was the CDF(4,4) (Cohen-Daubechies-Feauveau) b-spline integer transform, also known as the 13/7-T transform, with 5 decomposition levels. The first line of each table (where the "quality" parameter is 0) is for the lossless case, where MSE = 0 (PSNR = ∞).

The column labeled "pre" is the time (in milliseconds) for the wavelet transformation and all other tasks needed to actually start running the compression algorithm (mean extraction, conversion to sign-magnitude, etc). This step is independent of the amount of bits output and is in fact a lower bound for images of this size (it is almost independent of the contents of the image itself and mostly dependent on the image dimensions alone).

The column labeled "code" is the time (in milliseconds) for the actual compression algorithm and is directly proportional to the amount of bits output. Therefore, the specification of a quality factor or a maximum size will have great impact on this part and in the total running time for the image compression.

The implementation for the inner loop of the wavelet transform was done using the NEON vector instructions for the ARM with the help of the GCC (GNU Compiler Collection) intrinsics. A plain C implementation takes about 50 ms for the RPi2B and 90 ms for the Beaglebone for the "pre" column.

Table 1: RPi2B Timings (without ROI)

PSNR (db)	Size (bytes)	Pre (ms)	Code (ms)	Total (ms)
∞	277346	25.0	110.5	135.5
44.04	48111	25.0	24.5	49.5
40.66	22998	25.0	12.3	37.3
39.48	17356	25.0	9.0	34.0
37.38	10055	25.0	5.5	30.5
36.66	8366	25.0	4.5	29.5

Table 2: Beaglebone Timings (without ROI)

PSNR (db)	Size (bytes)	Pre (ms)	Code (ms)	Total (ms)
∞	277346	58.0	172.0	230.0
44.04	48111	58.0	40.5	98.5
40.66	22998	58.0	20.8	78.8
39.48	17356	58.0	15.0	73.0
37.38	10055	58.0	9.4	67.4
36.66	8366	58.0	7.5	65.5

6 Conclusions

This paper proposes the use of progressive image compression and the use of regions of interest

Table 3: RPi2B Timings (with ROI)

PSNR (db)	Size (bytes)	Pre (ms)	Code (ms)	Total (ms)
∞	283292	26.0	111.5	137.5
44.07	53667	26.0	26.5	52.5
40.72	27648	26.0	14.0	40.0
39.56	21074	26.0	10.7	36.7
37.48	13633	26.0	7.2	33.2
36.76	10590	26.0	5.5	31.5

Table 4: Beaglebone Timings (with ROI)

PSNR (db)	Size (bytes)	Pre (ms)	Code (ms)	Total (ms)
∞	283292	59.5	174.0	233.5
44.07	53667	59.5	44.0	103.5
40.72	27648	59.5	24.0	83.5
39.56	21074	59.5	17.8	77.3
37.48	13633	59.5	12.2	71.7
36.76	10590	59.5	9.5	69.0

(ROI) for underwater robotic applications, specially when there is limited bandwidth (i.e., wireless underwater RF channels) allowing for a much more agile data exchange between the I-AUV and a human operator supervising the underwater intervention. The operator could dynamically decide the quality, frame-rate or resolution of the received images so that the available bandwidth is utilized to its fullest potential and with the required minimum latency.

The results show that one core of a RPi2 can compress relatively high-resolution images (1024x768) in very high quality. It can be seen from table 1 that it can process close to 30 frames per second with a PSNR around 40 db and there are still 3 more cores to be used by other processes.

The Beaglebone is a less powerful board for images of this dimension but is still able to compress around 12.5 frames per second at the same 40 db PSNR. In this case, either the quality, frame size, or frame rate could be tuned so that the compression would use a certain predefined portion of the CPU, allowing other tasks enough timeslices to be executed on the same board.

As long as the ROI is a simple region, there is not much difference in encoding times for using it, even though there is a small penalty to pay in compression efficiency for the whole image (around 2% in our example).

A specially designed progressive compression algorithm has been implemented so that it possesses both quality and resolution scalability, ROI, and is simple enough with the goal of being usable in limited resource computers. The results show that it is quite competitive with state-of-the-art compression algorithms like JPEG2000 while being much faster.

Currently, an underwater short-distance wireless RF communications device is being developed. The expected bandwidth is around 40 kbps and it should be enough to transmit good quality detailed images of the region of intervention between 2 and 4 frames per second.

Acknowledgements

This work was partly supported by Spanish Ministry under grant DPI2014-57746-C3 (MERBOTS Project) and by Universitat Jaume I grants PID2010-12 and E-2015-24.

References

[1] P. J. Sanz, A. Peñalver, J. Sales, J. J. Fernández, J. Pérez, D. Fornas, J. García, and R. Marin, "Multipurpose underwater manipulation for archaeological intervention," in *Sixth International Workshop on Marine Technology (MARTECH'15)*, Cartagena, Spain, Sep 2015.

[2] P. J. Sanz, M. Prats, P. Ridao, D. Ribas, G. Oliver, and A. Orti, "Recent progress in the RAUVI project. A reconfigurable autonomous underwater vehicle for intervention," in *52-th International Symphosium ELMAR-2010*, Zadar, Croatia, September 2010, pp. 471–474.

[3] P. J. Sanz, A. Peñalver, J. Sales, D. Fornas, J. J. Fernández, J. Perez, and J. A. Bernabé, "GRASPER: A multisensory based manipulation system for underwater operations," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Manchester, UK: IEEE, Oct 2013.

[4] P. J. Sanz, P. Ridao, G. Oliver, G. Casalino, Y. Petillot, C. Silvestre, C. Melchiorri, and A. Turetta, "TRIDENT: An european project targeted to increase the autonomy levels for underwater intervention missions," in *OCEANS'13 MTS/IEEE conference*, San Diego, CA, Sept 2013, pp. 1–10.

[5] M. Prats, A. P. del Pobil, and P. J. Sanz, *Robot physical interaction through the combination of vision, tactile and force feedback. Applications to assistive robotics*, ser. Springer Tracts in Advanced Robotics, Volume 84. Springer Publishing Company, Incorporated, 2013.

[6] C. Pelekanakis, M. Stojanovic, and L. Freitag, "High rate acoustic link for underwater video transmission," in *OCEANS 2003. Proceedings*, vol. 2, Sept 2003, pp. 1091–1097 Vol.2.

[7] J. Ribas, D. Sura, and M. Stojanovic, "Underwater wireless video transmission for supervisory control and inspection using acoustic OFDM," in *OCEANS 2010*, Sept 2010, pp. 1–9.

[8] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey, "An integrated, underwater optical /acoustic communications system," in *OCEANS 2010 IEEE - Sydney*, May 2010, pp. 1–6.

[9] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 84–89, January 2009.

[10] H. Zhang and F. Meng, "Exploiting the skin effect using radio frequency communication in underwater communication," in *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, Aug 2012, pp. 1150–1153.

[11] M. Siegel and R. W. P. King, "Electromagnetic propagation between antennas submerged in the ocean," *Antennas and Propagation, IEEE Transactions on*, vol. 21, no. 4, pp. 507–513, Jul 1973.

[12] A. Shaw, A. Al-Shamma'a, S. Wylie, and D. Toal, "Experimental investigations of electromagnetic wave propagation in seawater," in *Microwave Conference, 2006. 36th European*, Sept 2006, pp. 572–575.

[13] W. B. Pennebaker and J. L. Mitchell, "JPEG still image data compression standard," New York: Van Nostrand Reinhold, 1992, 1992.

[14] Z. Xiong, O. Guleryuz, and M. Orchard, "A DCT-based embedded image coder," *Signal Processing Letters, IEEE*, vol. 3, no. 11, pp. 289–290, Nov 1996.

[15] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3445–3462, Dec 1993.

[16] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, no. 3, pp. 243–250, Jun 1996.

[17] W. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 11, pp. 1219–1235, Nov 2004.

[18] D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi, "Embedded block coding in JPEG 2000," *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 49 – 72, 2002.

[19] V. J. Rehna and M. K. J. Kumar, "Wavelet based image coding schemes: A recent survey," *CoRR*, vol. abs/1209.2515, 2012. [Online]. Available: http://arxiv.org/abs/1209.2515

[20] M. Subedar, L. Karam, and G. Abousleman, "An embedded scaling-based arbitrary shape region-of-interest coding method for JPEG2000," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 3, May 2004, pp. iii–681–4.

[21] X. Delaunay, C. Thiebaut, M. Chabert, V. Charvillat, and G. Morin, "Progressive coding of satellite images with regions of interest," in *On-Board Payload Data Compression Workshop*, Toulouse, France, Oct. 2010.