An FPGA implemented bridge over water recognition for an image evaluation on-board of satellites

Sebastian Beulig[a], Maria v. Schönermark[a] and Felix Huber[a]

[a]German Aerospace Center (DLR), German Space Operations Center (GSOC), Münchner Str. 20, 82234 Wessling, Germany

ABSTRACT

This paper presents an advanced method for the automatic recognition of bridges over water in high resolution satellite image data, intended for an application on-board of satellites. The algorithm is implemented in reconfigurable hardware, a so-called Field Programmable Gate Array (FPGA). Within a few seconds a thematic map is derived from the original satellite image. The map contains information about the water areas, islands and bridge deck areas in the captured scene. No a-priory knowledge is needed. Due to the autonomous image processing and the low power consumption of the FPGA, this implementation seems suitable for an application on-board of satellites. Especially in case of a natural disaster it could provide quick information about accessible transportation routes. The algorithm as well as experimental results on panchromatic and near-infrared satellite imagery are presented in this article. The obtained results are promising.

Keywords: FPGA, change detection, bridge recognition, on-board image processing, satellite autonomy

1. INTRODUCTION

The image acquisition chain of an Earth Observation (EO) satellite follows even nowadays a rather strict concept. The acquisition date is scheduled on ground and uplinked to the satellite when it comes into view of one of its ground stations. The satellite executes the tasks according to the time-line. Later, the captured images are handled after the 'store-and-forward' scheme,[1] i.e. the image data is stored in the on-board mass memory and downlinked to the ground station afterwards. These steps are executed without regard to the actual image information content. Not until the imagery is analysed on ground, its true value can be identified.

A smarter and more time efficient approach would be to evaluate the captured satellite image data directly on-board. Yet, a brief examination could reveal crucial information. Corrupted or unusable data could be erased to free valuable memory space.[2] In case of spotted changes, the observation time-line could be rearranged to acquire additional shots of the scene.[3] These ancillary information would support the work of crisis information centres and raise the benefits of a satellite mission.

In case of a flooding event, not only the affected area but also the accessible transportation routes are of interest. Therefore, a retrieval of still intact bridges over water is reasonable. But because computational resources on-board a satellite are limited, a processing unit that deals with large image data volumes though having a small power consumption is needed. So-called Field Programmable Gate Arrays (FPGAs) fulfil these characteristics.[4] Besides, they are successfully operated on-board of several satellite missions, e.g. the FedSat[5] or the X-Sat.[6] For this reason, the proposed approach is the implementation of a bridge-over-water-recognition directly into hardware. Due to the fact that the complexity of an algorithm is limited by the size of the chip, a new algorithm was particularly designed to fit into a small FPGA.

Within a few seconds a thematic map is generated from the original satellite image. This map contains useful information about the captured scene. Furthermore, it can be compared with map data of previous points in time to discover changes. If larger differences are recognised, a trigger signal could be sent to the on-board planning unit. With regard to the satellite's status, e.g. the available power and memory space, a second data take could be initiated immediately. The working flow of the method as well as experimental results will be presented in this article.

Send correspondence to Sebastian Beulig
E-mail: sebastian.beulig@dlr.de, Telephone: +49 (0) 8153 28 3509

The structure is as follows: In section 2 the steps of the algorithm are described in detail. The results of the experiments are presented in section 3. In section 4 there is a discussion about the obtained output. Finally, the article is summarized in section 5.

2. THE ALGORITHM

2.1 Principle working flow

The overall goal of the proposed method is to derive a thematic map from a grey value satellite image. In the end, the map contains useful information about water areas, islands and bridges over water in the captured scene. This map can be further used to discover changes.

At first a water mask is generated from the original satellite image. One method to derive a mask from panchromatic or near-infrared imagery is presented in.[7] In general, the applied sensor data as well as the method to derive such a water mask are not really of relevance, because all further processing steps only rely on a binary mask. In principle, the proposed method to recognise bridges over water consists of three steps.

1. Detection of all islands
2. Recognition of potential bridge deck areas
3. Reduction of false positives

In the following sections these steps will be explained in detail.

2.2 Island detection

In the beginning, the term *island* shall be clarified. It means: A non-water area completely surrounded by water. The nature of an island, i.e. whether it is a ship or a rock or something else, is of no importance. All islands have to be particularly assigned, to prevent distractions in the later bridge recognition process. Nevertheless, in case of a flooding event, it is an additional useful information to know where small islands are, since they could be rooftops, where people are waiting for rescue.

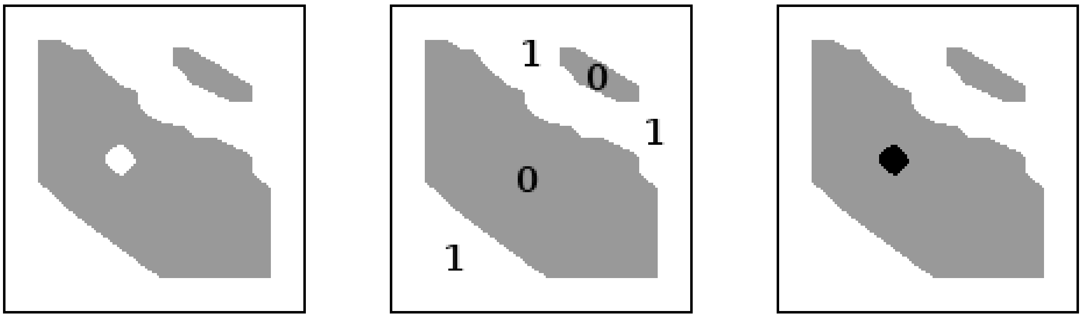
From another point of view, these islands represent holes in the water areas and there are approaches in the literature to fill them, e.g. by *region growing*.[8] But in contrast to the standard procedure, to erase these holes, here they are labelled separately. The working flow is depicted in figure 1(a)-(c). Figure 1(a) shows a small cut-out from a water mask. The water is coloured grey and there is a small (white) island in the centre of the image. Now, a binary segmentation is performed and the result is depicted in figure 1(b). It represents a second, separate memory space, that has the same dimension as the original image/water mask. All white pixels in the water mask, that have direct contact to the white rim of the image, are labelled with the number 1 and remain white. And all grey pixels (water) as well as all white pixels that have no direct contact to the white image rim are labelled with the number 0 and are coloured grey. Please note that in this figure, the colouring is only for visual purposes to outline the different segments.

In the end, both figures are compared. All white pixels in the water mask, that are labelled with the number 0 in the separate memory space, are designated as island pixels. The resulting map is presented in figure 1(c). The necessary information, i.e. the island, is coloured black. With that method, all islands of any size can be marked in a water mask. This first thematic map is then the input for the next processing step.

2.3 Potential bridge deck areas

In the following step all (potential) bridge deck areas shall be recognised. The result looks similar to an approach presented in,[9] but the procedure differs. Basically it consists of two morphological operations, erosion and dilation. In figure 2(a) a cut-out from another water mask is depicted. The water is coloured grey and the white passage between the two water segments represents a bridge. Additionally, there is a big concave in the water segment on the right hand side. It is a ship that was connected to the non-water area due to an erosion step in the water mask generation.

The present approach starts with an extension of the water segments (dilation) by 16 pixels. This value is fix and depends on the spatial resolution of the data. To the best of our knowledge, all bridges over water on earth

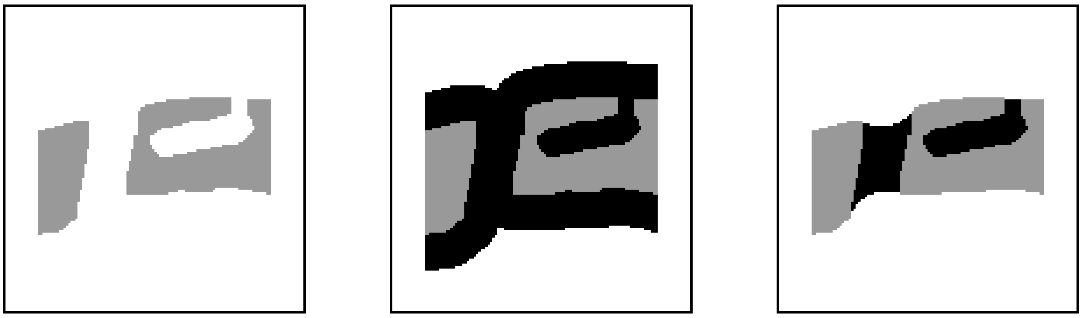


(a) (b) (c)

Figure 1. Main steps of the island detection algorithm. (a) The water mask. (b) Visualisation of the binary segmentation. (c) Resulting thematic map, the island is black.

have a width no greater than $100m$. With the given spatial resolution of $5m$ this represents 20 pixels. And because a bridge over water is adjacent to two water segments, each segment has to be enlarged by 10 pixels to close the gap. However, a more robust output can be obtained by taking 16 pixels. With this step, a new water area is generated, that is coloured black in figure 2(b). As it can be seen from the image, both water segments are melted together and the concave is filled.

Then, beginning from the new water rim, this extended water area is reduced (erosion) by 16 pixels. The result is displayed in figure 2(c). Due to a distance of more than 16 pixels from the new water rim, some pixels on the bridge and in the concave remain black. So, the whole bridge deck area is labelled, but also the concave, that is obviously not a bridge. To distinguish between a real bridge and a false positive an additional processing step will be performed. That is explained in the next section.



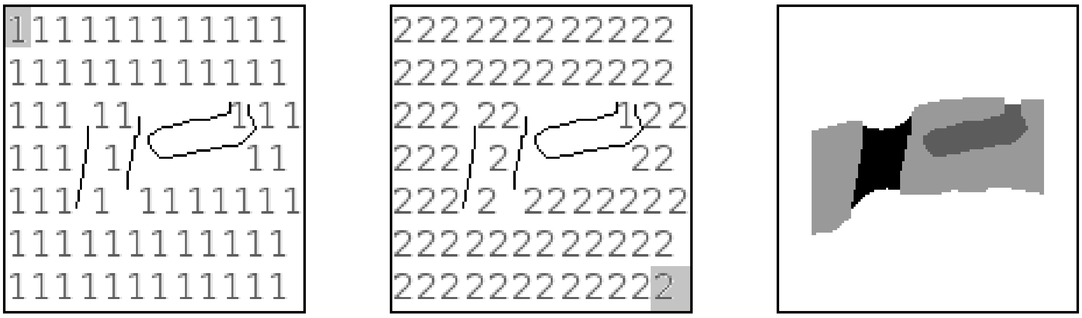
(a) (b) (c)

Figure 2. Main steps of the bridge recognition. (a) Water mask. (b) Water mask with extended water areas in black. (c) Resulting thematic map with extracted potential bridge deck areas, labelled in black.

2.4 False positive reduction

The approach of a brief image evaluation on-board of satellites allows a certain inaccuracy. But, due to the fact that the previous processing step is performed blindly on the whole image, a lot of unwanted areas are declared as bridges, too. So, the present step significantly reduces the number of false positives and is recommended although it increases the processing time.

Figure 3(a) again represents a separate memory space that has the dimension of the original image/water mask. Only the water pixels from figure 2(c) that have a direct contact to a potential bridge pixel are taken over in this image/memory space. They are visualised as thin black lines. Now in figure 3(a), the first pixel in the upper left corner is set to 1. Then, line by line, all white pixels will be labelled with 1, if they have a white pixel, already labelled with 1, in their adjacent neighbourhood. The black lines act as barriers. Thereby, some small areas inside and around the concave remain unlabelled.



(a) (b) (c)

Figure 3. Main steps of the false positive reduction. (a) The forward step. A visualisation of the separate memory space filled with numbers of 1, starting from the upper left pixel. (b) The backward step. The same area filled with numbers of 2, starting from the lower right pixel. (c) The resulting thematic map. The approved bridge is black and the false positive area is labelled dark grey.

When the whole image is processed, the first pixel in the lower right corner is set to 2 (see figure 3(b)). Again, all pixels, that are labelled with 1 and that have a pixel labelled with 2 in their adjacent neighbourhood, are also labelled with 2. Now, because the actual pixel can only 'look' backwards (and downwards), and the black lines are 'non-transparent', also pixels inside the concave that have been labelled with 1 in the forward step remain untouched in this step. In the end, this specially labelled image is compared with the result of the bridge recognition step. All potential bridge pixels that are labelled with 2 are designated as real bridge pixels. Whereas all potential bridge pixels, that have no label or labelled with 1 are marked as false positives. The result is shown in figure 3(c). The real bridge pixels are coloured black and the false positive pixels are in dark grey. How well this method works on real satellite image data is presented in the next section.

3. EXPERIMENTS

3.1 Satellite image data

In this study two kinds of satellite image data were analysed: Pan-chromatic imagery taken by the *High Resolution Camera* (*HRC*) on-board the Proba-1 satellite and near-infrared imagery acquired by the *RapidEye* (*RE*) satellites. The *HRC* and the *RE* level-3A data have a spatial resolution of $5m$. In order to ensure a large variety of scenes, different locations from all over the world have been selected. A dataset of 20 *HRC* and 20 *RE* images was arranged.

To demonstrate the result of the proposed algorithm, an example image was selected from that dataset and is presented in figure 4. It is a cut-out of a *RE* scene near the city of Düsseldorf, Germany. The image was captured on May 30 2011 and the approx. center coordinates are 51°11'10"N 6°46'30"E. In the center of the image the river Rhine is visible. A motorway is crossing and several ships cruise the river. Please note that the original image was enhanced here in brightness and contrast for visualisation purposes. The water mask obtained from that image[7] is displayed in figure 5. This mask is the basis for the onward processing step.

3.2 Island detection results

The outcome of the island detection is displayed in figure 6. The water is coloured grey and the islands are in black. Both islands are marked, regardless of their size. Overall, every island in the evaluated water masks was found by the proposed method.

3.3 Bridge recognition results

The final outcome, i.e. the thematic map, for the example image is depicted in figure 7. The water bodies are again in grey, the islands are now in light grey, the false positive areas are in dark grey and the real bridges are in black. For the actual image the proposed method works quite well. Nevertheless, still some falsely labelled bridges occurred in the test set, due to an inaccurate water mask. Besides, there appeared some difficulties with shorter bridges. In general, the assessment of 40 satellite images revealed a recognition rate of around 85%.

Figure 4. An enhanced NIR image of an area near the city of Düsseldorf. The image was acquired by RapidEye. ©DLR/RapidEye

Figure 5. The water mask derived from the image in figure 4. Water areas are in grey.

3.4 Hardware configuration

All algorithms are running on a Xilinx Spartan 3E-1200 FPGA. This chip is embedded on the Nexys-2 development board. The clock frequency is limited to $12.5MHz$ due to RAM access delay times. But because of this low clock rate, the power consumption is less than $1W$. The algorithms are written in Handel-C, a C-like hardware description language. The evaluated satellite images have a size of 900x900 pixels. To avoid boundary problems, a white rim frames each image, so that the actual size is 1024x1024 pixels. On average, an image is

Figure 6. The result of the island detection. Islands are in black.

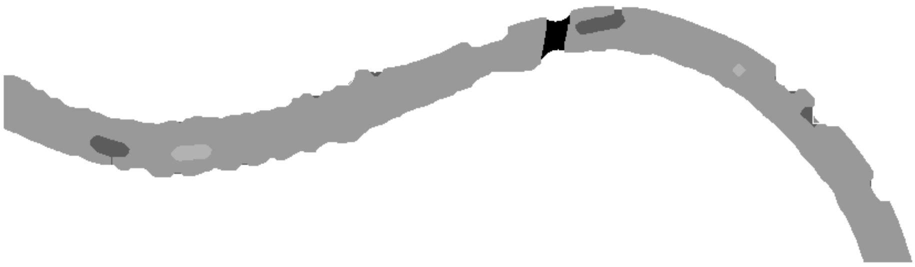


Figure 7. The resulting thematic map. The approved bridge is in black. False positives are in dark grey. Islands are in light grey.

processed in around 18 seconds. It takes around 4 seconds for the island detection and around 14 seconds for the whole bridge recognition, including the false positive reduction. These times vary significantly with the number and size of the water segments.
For an application on-board of a satellite, the final implementation would be on a space-proofed FPGA, e.g. from the Virtex chip family, with a higher clock frequency and faster RAM access times and thereby these processing times would decrease.

4. DISCUSSION

The proposed method derives a thematic map from a binary water mask. This mask represents the basis for the actual algorithm, but is not under consideration here (for discussion see[7]). Nevertheless, it is often a root for false detections.
The so-called *island detection* performs well. All islands in the water masks are labelled by this method. But be aware of non-water areas that are near to the water's edge in a satellite image. If they are in conjunction with the main land on the water mask, then they are not recognised as islands.
A new approach was presented to recognise the bridge deck areas. The procedure is rather straightforward and it yields robust results. It only failed in cases, where the water mask was too inaccurate. But in addition to real bridges also a lot of unwanted areas were marked.
To exclude these falsely labelled areas, an additional processing step was invented, too. This step greatly reduces the number of false positives. Additionally, it can handle the case when a ship is crossing under a bridge, to distinguish automatically between the ship and the bridge. But it also comes at a cost. If a bridge has a very strong curvature, it will not be recognised as a bridge any more. Small bridges pose a limitation, too. But this can eventually be overcome with the usage of satellite image data with a higher spatial resolution. Overall, the recognition rate of around 85% is promising and is in accordance with results of 89.61%[10] and 80.95%[11] in the literature.

5. SUMMARY

In this paper an advanced method for the recognition of bridges over water in high-resolution satellite imagery was introduced. It is dedicated to an application on-board of satellites to raise the efficiency of the image acquisition process. With regard to power and resource constrains on-board of a spacecraft, the algorithm has been implemented in a small FPGA. The processing of a satellite image is accomplished autonomously and within a few seconds. The final thematic map provides useful information about the captured image. With labelling the whole bridge deck area, a detailed analysis of the bridge's structure may be addressed in future research activities. Altogether, the proposed hardware implementation seems a reasonable tool for a brief image evaluation on-board. Thus, it could strengthen the pursued autonomy of satellites.

ACKNOWLEDGMENTS

The authors would like to thank the European Space Agency (ESA) for providing satellite image data from Proba-1/HRC. Further gratitudes go to the DLR for providing satellite image data from the RapidEye Science Archive.

REFERENCES

[1] Sophiayati Yuhaniz, S. and Vladimirova, T., "An onboard automatic change detection system for disaster monitoring," *International Journal of Remote Sensing* **30**(23), 6121–6139 (2009).

[2] Dawood, A., Williams, J., and Visser, S., "On-board satellite image compression using reconfigurable fpgas," in [*Field-Programmable Technology, 2002. (FPT). Proceedings. 2002 IEEE International Conference on*], 306 – 310 (2002).

[3] Chien, S., Sherwood, R., Tran, D., Castano, R., Cichy, B., Davies, A., Rabideau, G., Tang, N., Burl, M., Mandl, D., et al., "Autonomous science on the eo-1 mission," (2003).

[4] Hauck, S. and DeHon, A., [*Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*], Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007).

[5] Dawood, A., Visser, S., and Williams, J., "Reconfigurable fpgas for real time image processing in space," in [*Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*], **2**, 845–848, IEEE (2002).

[6] Bretschneider, T., Ramesh, B., Gupta, V., and McLoughlin, I., "Low-cost space-borne processing on a reconfigurable parallel architecture," in [*Proceedings of the International Conference on Engineering of Reconfigurable Systems and Algorithms*], 93–99, Citeseer (2004).

[7] Beulig, S., von Schönermark, M., and Huber, F., "A fpga-based automatic bridge over water recognition in high-resolution satellite images," in [*SPIE Remote Sensing*], 853713–853713, International Society for Optics and Photonics (2012).

[8] Pedrino, E. C., Saito, J. H., Senger, H., Roda, V. O., and Marinho, M., "An fpga-based region-growing architecture for binary images," *17th International Conference on Systems, Signals and Image Processing* (2010).

[9] Gu, D., Zhu, C., Shen, H., Hu, J., and Chang, H., "Automatic bridge extraction for optical images," in [*Image and Graphics (ICIG), 2011 Sixth International Conference on*], 446–451, IEEE (2011).

[10] Chen, A., "Novel approach for recognizing bridges over water in large remote sensing images," in [*Computational Intelligence and Design, 2008. ISCID'08. International Symposium on*], **2**, 47–51, IEEE (2008).

[11] Zhang, J. and Sun, G., "Recognition of bridge over water in remote sensing image using discrete hopfield neural network," in [*Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*], 439 –442 (2011).