# Automatic bridge detection in high-resolution satellite images

Roger Trias-Sanz, Nicolas Loménie

# Automatic Bridge Detection in High-Resolution Satellite Images

Roger Trias-Sanz* and Nicolas Loménie

Université de Paris 5, laboratoire SIP-CRIP5.
45, rue des Saints-Pères; 75006 Paris; France
Roger.Trias.Sanz@gmx.net

**Abstract.** A set of methodologies and techniques for automatic detection of bridges in pan-chromatic, high-resolution satellite images is presented. These methods rely on (a) radiometric features and neural networks to classify each pixel into several terrain types, and (b) fixed rules to find bridges in this classification. They can be easily extended to other kinds of geographical objects, and integrated with existing techniques using geometric features. The proposed method has been tested in a number of experiments.

## 1 Introduction

Automatically detecting geographical objects such as bridges, roundabouts or road crossings on high-resolution satellite images is useful for keeping up to date geographical databases, automatically locating the imaging satellite at low cost and easily and quickly assessing the extent of damages in case of natural disasters such as flooding or earthquakes. In addition, it may help in content-based indexing of such satellite images.

A very limited number of articles exist in this particular direction. However, a lot of work has been done on sub-problems, such as terrain classification, that could be part of a geographical object detection system.

A system capable of detecting objects —such as chairs, cars, tables— which are large with respect to the image they appear in is described in [12]. It uses multiple cooperating, negotiating agents. No learning mechanism is used.

In [8], a system capable of extracting objects and regions such as roads, lakes and fields from aerial images is presented. It uses a few agents or specialists which are trained using a corpus-based learning mechanism.

Neural networks are used in [4] to classify pixels in LANDSAT images. [5] uses spatial regularities to do an unsupervised terrain classification. This kind of systems tend to give visually imperfect results: [2] proposes a rule-based system to improve the results of these classifications, but uses data which is not available to our system, such as terrain elevation.

* This author is also with Institut Géographique National; 2-4 av. Pasteur; 94165 Saint-Mandé cedex; France.

A previous approach [6] was developed at the SIP-CRIP5 center using only geometric models of bridges and roundabouts. It turned out not to be totally satisfying.

In contrast to [6], we wanted to incorporate radiometry —and in particular texture— into the detection process, and to use learning methods so that the system would be more adaptable.

## 2 System overview

We present a set of techniques and methodologies to automatically detect bridges on small high-resolution pan-chromatic satellite images. These are real images, provided by the French space agency (CNES), which feature bridges in different positions, orientations and sizes, and of different kinds (road over water, road over road, walkway over road, rail over water, ...). Modeling the "bridge" concept turns out to be a very difficult task.

Bridges appear together with other complex objects in these images, such as roundabouts, buildings, and road crossings. In addition, we not only want to decide the presence or absence of a bridge in an image, but also to determine its position, size and orientation.

These detection techniques rely on radiometric features and neural networks to classify each pixel into several terrain types, and fixed rules to find bridges in this classification.

We produced a hybrid system: a bottom-up part uses (mainly) texture analysis, neural networks and a voting mechanism to classify each image pixel into terrain classes: water, road, green, ... A top-down part uses fixed rules to detect bridges given that classification. To detect the bridges in an image (see figure 1):

1. a certain number of textural and geometric parameters are calculated for each pixel in the image (Sect. 3);
2. for each pixel, the set of parameters corresponding to it are fed into a neural network. This network tries to determine the kind of terrain the pixel belongs to (Sect. 4). The response is noisy and imperfect; a post-processing phase tries to improve it (Sect. 4.3);
3. a set of subroutines look for regions of a certain type and dimensions, according to some manually-produced detection rules (Sect. 5).

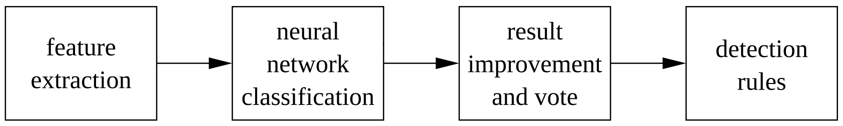

**Fig. 1.** processing steps

This article is structured as follows: first Sect. 3 describes the extraction of radiometric and geometric features , terrain type analysis is presented on Sect. 4, and detection rules on Sect. 5. Evaluation of the system's performances can be found on Sect. 6. In Sect. 7 we give some directions for future research. Some conclusions close the main part of this article.

## 3 Low-Level Feature Extraction

In the first step of the detection process, we calculate a feature vector field for the source image. Each feature vector describes the neighborhood of one pixel using textural, radiometric and geometric attributes.

The original image, the original image with histogram equalization, and a denoised version are included into the feature vector field. The feature vector contains, in addition, the following parameters:

1. Intensity gradient (module and argument) calculated using Deriche's method for several values of $\alpha$. Using different values of $\alpha$ gives us an edge detection at different levels of detail.
2. Entropic structure: This parameter [1] gives the level of structuration on the neighborhood of a pixel. It allows us to distinguish among homogeneous areas, areas structured by man-made constructions, and unstructured areas (see appendix A).
3. Shadow predicate: This parameter (adapted from [1]) indicates which pixels belong to "shadow" areas. This is done by finding a "shadow threshold" from the position of the first local minimum in the intensity histogram.
4. First-order texture parameters: The mean $\mu$, entropy, energy, variance $\sigma^2$, skewness and kurtosis of that histogram, and a variation coefficient $c_v = |\sigma|/\mu$, of the first-order intensity histograms.
5. Texture signal activity.
6. Local histograms.
7. Fourier transform texture parameters: The maximum, mean, root-mean-square and variance, of the amplitude of the complex Fourier transform on a square neighborhood of each pixel.
8. Gray-level difference texture parameters (see appendix A).
9. Region size and compactness. For four different sets of parameters, each giving different levels of detail, we segment the original image into regions. We use, as image parameters, the area and compactness of the region each pixel belongs to.

## 4 Terrain Classification

After the feature vector field for an image has been calculated, we use a neural network to try to determine which type of terrain each pixel belongs to. For this particular application, we chose the following terrain types: water, vegetation, building, railroad, road, bridge and roundabout.[1] Of course we do not expect to detect bridges at this step, we just want to detect pixels that locally, look like belonging to a bridge. Each feature vector for an image is fed to a neural network in succession. This neural network has 8 outputs, one for each terrain type (there are two classes for roundabout terrain). The network sets each output to a real value in $[0, 1]$, higher values meaning higher "confidence" that the pixel belongs to that terrain type.

The Stuttgart Neural Network Simulator (SNNS) has been used to train and run a 4-layer feed-forward neural network with 107 input neurons, 39 hidden neurons on the first hidden layer, 20 on the second, and 8 output neurons, fully connected. The parameters of the RPROP learning algorithm (resilient back-propagation) have been set to $\Delta_0 = 0.2$, $\Delta_{\max} = 50$ and no weight decay. Neurons had logistic activation functions; weights and biases were randomly initialized with a $\mathcal{U}(-1, 1)$ uniform distribution, and training patterns were randomly permuted. These parameters were determined empirically.

## 4.1 Training the Neural Network

A set of images containing bridges, roundabouts and counterexamples, provided by CNES (the French space agency) has been manually labeled. For cross-validation, we divided our images into three homogeneous groups, $a$, $b$ and $c$.

Then training on three networks has been performed using two groups as training sets and the remaining group as validation set (on one network we used $a$ and $b$ for training, on another we used $a$ and $c$, and on the last one we used $b$ and $c$ as training data). After 200 training iterations, we obtained validation error rates of 0.222, 0.212 and 0.223. Error rates are given in root-mean-square error (RMSE) per output neuron (see appendix A). For the rest of this project, we selected one of the three networks.

These error rates look large, but are acceptable because we do not need to obtain a perfect detection at this step and because errors are distributed in a way that makes the results satisfactory. See some examples in Fig. 2.

## 4.2 Multiple Resolutions

This system is able to work with images taken at different resolutions, if all of them are present in the training set: The neural network module should be capable of learning characteristic features for different resolutions without modification. We then scale the results of the neural network so that the following phases operate on images all at the same resolution. Scaling must be done at this point and not before, because images at different resolutions have different texture characteristics.

[1] In the first stages of this project we also wanted to automatically detect roundabouts. However, we were not supplied with enough training images for that.

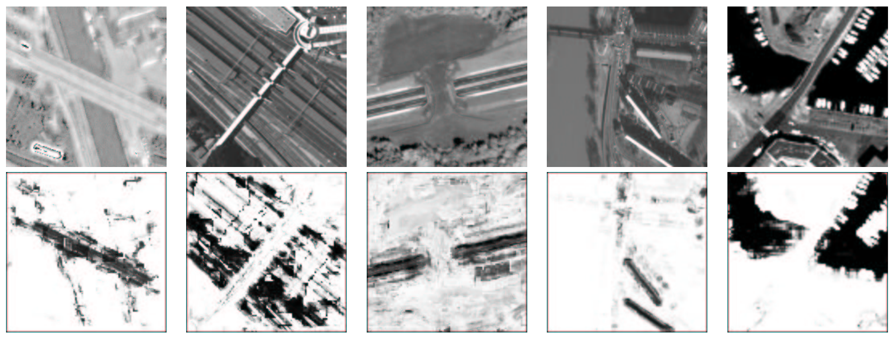

**Fig. 2.** sample neural network output; top: source images, bottom: network output (darker pixels represent stronger responses); from left to right, example of *bridge*, *railroad*, *road*, *building* and *water* output channels

### 4.3 Results Improvement and Voting Phase

The results of the neural network are noisy, fuzzy and full of holes and other artifacts. To improve them,

1. each channel (corresponding to one terrain type) is smoothed by convolution with a Gaussian mask, and then thresholded;
2. in a neighborhood of each pixel, we calculate a weighted histogram of terrain types. We weight each pixel in the neighborhood based on its distance from the base pixel and its terrain type. For each pixel, the terrain type with higher histogram count wins the "voting"; finally
3. we further regularize the resulting regions by mathematical morphology opening and closing operations, and by removing small regions.

## 5 Rule-Based Detection

The final step towards bridge detection, once we have a good classification of pixels into terrain types as given by the neural network and the vote procedure, is to apply a certain number of "detection rules" to that terrain classification.

These manually-produced rules match particular combinations of regions of a certain type and geometry, returning a possible bridge location, dimensions and orientation. We give here an informal description of some of them. Expressions such as "large" or "near" are in fact translated into hard thresholds, but this would be a good place to put fuzzy logic in. See [10] for the formal definition of these rules and for the specific threshold values.

1. Two large regions of water or rail terrain (same type for both regions) are separated by a narrow and long strip. This strip is a bridge.
2. One large and narrow region of bridge terrain is a bridge.

3. There is a narrow and long region of bridge or road terrain separating two large regions of water or rail terrain (same type for both regions). This strip is a bridge.
4. Two regions of road terrain, long and narrow, are separated by less than a certain distance. Additionally, they are *aligned.* There is a bridge at the middle of the separation between the two regions.
5. We apply rule (4) not to road terrain, but to the terrain channel resulting of taking all road and bridge terrain and removing any water, vegetation or rail intersecting it.
6. One long and narrow region of road or bridge terrain intersects a very narrow strip of road or bridge terrain —both regions of *different* type. Both regions are roughly orthogonal. Then there is a bridge at the intersection.

See Fig. 3 for some examples.

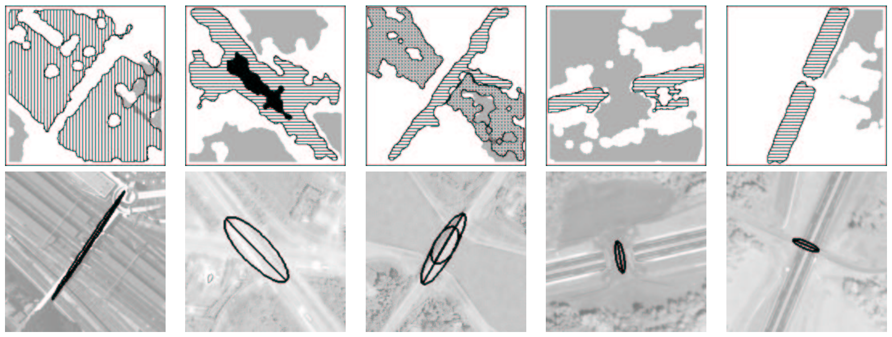

**Fig. 3.** detection rules; top: input to the detection rules (black=bridge, gray=green, vertical=railroad, horizontal=road, crosses=water), bottom: corresponding source images and detected bridges; these bridges are detected by (from left to right) rule 1, rule 2, rules 1 and 3, rule 4, rules 4 and 5

## 6 Evaluation

Evaluation is performed on a set of small (100 × 100 and 200 × 200 pixels), gray-level, high-resolution satellite images (Ikonos-2 images at 1 m$^2$ or 16 m$^2$ per pixel) containing bridges, roundabouts and counterexamples (objects that look like bridges or roundabouts, but are not). The complete system using these techniques processes input images at one to two minutes per image on a 900 MHz Pentium III computer.

Performance of the whole system has been evaluated by running the detection process for each image in our database.

Table 1 gives the number of images with scores of correct detections (real bridges that the system detects) and false alarms for a typical set of parameters.

A large part (43%) of our images were of low quality (fuzzy images, images where not even a human observer could decide on the presence of a bridge, bad lightning or sensor saturation) or of significantly lower resolution than the others (see Fig. 4). Bad classification by the neural network module caused 29% of the errors. Low image quality caused 41% of the errors. Because of the strong effect of abnormally low image quality on the system's performances, we also give results taking into account only good-quality images.

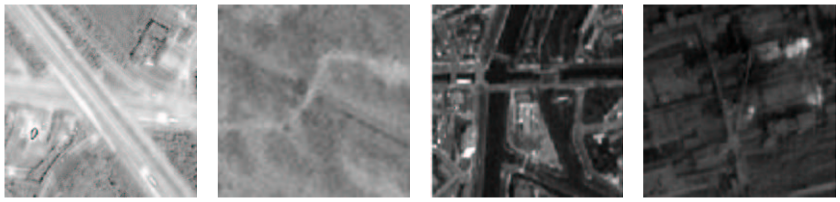

**Fig. 4.** Low quality images

**Table 1.** system evaluation with the final image set

| | whole set | good quality |
|---|---|---|
| total images | 254 | 146 |
| images with bridges | 99 | 41 |
| correct bridge detection (over img. with bridges) | 21% | 41.5% |
| false bridge detection (over all images) | 7.1% | 5.5% |

The system correctly processes 70.5% of all images. It correctly processes 21% of images containing bridges, and 85% of images containing false bridges (objects resembling bridges). This performance is satisfactory given the experimental nature of this system and the difficulty to model such a complex concept as bridge. Besides, all validation test have been made on real images, often of very poor quality.

The system is clearly biased towards under-detection. If this is not the desired behavior, we can change that by modifying the weights in the vote process and the thresholds in the detection rules.

## 7 Suggestions for Future Research

We present in this section some suggestions for future research on the area of this project.

### 7.1 Automatic Rule Construction

A technically more interesting approach to the top-down detection part would be to use, there too, learning methods instead of fixed, human-programmed rules.

We tried to use mobile, situated agents endowed with a learning mechanism. This novel approach (we could not find references on similar work; but see [3] for a system using non-mobile, non-situated multi-agent systems) is inspired by reinforcement learning methods used in the Animat Approach to robotics [7, and others]. Robots move and act in their environment, of which they have a partial knowledge given by their sensors. Their actions have an effect on the environment, which in turn produces *reward* or *punishment* stimuli for the robots.

We programmed software agents that could move on an artificial environment which was in fact the feature vector field described in Sect. 3. Agents are "located" on pixels on the image. Their "sensors" perceive the feature vector at that position only. Agents may move or classify the current pixel. During learning, the environment reacts to a classification decision by rewarding or punishing the agent, depending on its correctness and other criteria, such as coverage or speed.

As a proof-of-concept experiment, we gave these agents the task of improving the results of the neural networks, given these results and the feature vector field for an image. However, the experiments were a failure. Analysis of the system's output suggests that there may be problems in the fitness function used in the genetic programming-based learning mechanism.

We believe there is still plenty of room for further research into this direction. Automatic rule construction using data mining techniques, such as decision trees, should also be explored.

### 7.2 Integration with Geometry-Based System

Related work at the SIP-CRIP5 laboratory [6] was made about using geometric features and models to detect bridges and roundabouts in satellite images, with some success. Both systems can be combined by using the output of [6] could be used as an additional channel to the feature vector field of this system. In that way, our terrain classification would be based not only on radiometry attributes, but also on higher-level geometric properties.

### 7.3 Larger-Scale Descriptors

This system relies on image descriptors calculated on small neighborhoods of each pixel. We found that it was difficult, even for a human, to detect bridges and roundabouts using local information only, as our system was requested to do. Combining the system with larger-scale detectors such as detectors for road networks [9] may help. The output of such detectors can be used as an additional component of the feature vector field.

## 8 Conclusion

Our initial goal was to develop a set of techniques and methods to automatically detect bridges in high-resolution satellite images. We have presented several such techniques: terrain classification by neural networks operating on textural parameters, learning mobile agents, and static detection rules. We have implemented and integrated them in a running system which can easily cooperate with other approaches such as a geometry-based approach [6]. We have evaluated its performance, with satisfactory results. We believe that our techniques are easily generalizable to other kinds of objects; however we have not conducted experiments to show it.

We have shown that using texture information to classify terrain areas, and combining the resulting regions using geometrical properties, is enough to detect some kinds of bridges. We have experimented, unsuccessfully, with learning mobile agents as a tool for object detection and image enhancement, and we have pointed to future experiments to be done on that area.

## 9 Acknowledgments

This work was partly supported by the French National Center for Space Studies (CNES). We would like to thank the reviewers for their constructive remarks.

## References

[1] C. Baillard. *Analyse d'images aériennes stéréo pour la restitution des milieux urbains.* PhD thesis, Institut Géographique National, 2-4 Av. Pasteur, 94165 Saint-Mandé, France, 1997.

[2] J. Desachy. A knowledge-based system for satellite image interpretation. In *Proc. 11th IAPR Intl. Conf. on Pattern Recognition (ICPR '92)*, volume 1, pages 198–201, The Hague, The Netherlands, August 1992. IAPR, IEEE.

[3] R. J. Gallimore et al. 3D scientific data interpretation using cooperating agents. In *Proc. 3rd Intl. Conf. on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98)*, pages 47–65, London, UK, 1998.

[4] S. Kamata et al. A neural network classifier for LANDSAT image data. In *Proc. 11th IAPR Intl. Conf. on Pattern Recognition (ICPR '92)*, volume 2, pages 573–576, The Hague, The Netherlands, August 1992. IAPR, IEEE.

[5] A. Ketterlin, D. Blamont, and J. J. Korczak. Unsupervised learning of spatial regularities. http://citeseer.nj.nec.com/1418.html.

[6] N. Loménie, J. Barbeau, and F. Cloppet-Oliva. Détection de carrefours routiers et de ponts dans les images satellitales à haute résolution. Technical report, CRIP5-SIP, Université de Paris 5, Paris, France, November 2001.

[7] J.-A. Meyer and S. W. Wilson, editors. *From Animals to Animats: Proc. 1st Intl. Conf. on Simulation of Adaptive Behavior.* MIT Press, February 1991.

[8] P. Robertson. Grava – a corpus based approach to the interpretation of aerial images. http://citeseer.nj.nec.com/25230.html.

[9] R. Stoica, X. Descombes, and J. Zerubia. A Markov point process for road extraction in remote sensed images. Technical Report RR-3923, INRIA, Sophia-Antipolis, France, April 2000.

[10] R. Trias-Sanz. Automatically detecting geographical objects in high-resolution satellite images. Master's thesis, Laboratoire SIP, CRIP5, Université René Descartes-Paris 5, 45 rue des Saints-Pères; F-75006 Paris; France, September 2002.

[11] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6:269–285, 1976.

[12] K. Yanai and K. Deguchi. An architecture of object recognition system for various images based on multi-agent. In *Proc. 14th Intl. Conf. on Pattern Recognition (ICPR '98)*, volume 1, pages 278–281, Brisbane, Australia, August 1998. IEEE.

## A Some Definitions

**Entropic Structure.** Let $G$ be the gradient vector field of the image, in our case calculated using Deriche's method with $\alpha = 0.65$, $r$ a neighborhood radius and $t_1, t_2$ two thresholds. Let $D$ be the domain of $G$. For a pixel $p$, we find $N_p := \{z : z \in D, \|z - p\| \leq r, \|G(z)\| > t_1\}$. If card $N_p < t_2$, $p$ is deemed to belong to a homogeneous area. Otherwise, its entropic structuration is the entropy of the histogram $h_p(i)$ (see [1]). We used $r = 8$, $t_1 = 4000$ and $t_2 = 25$.

**Gray-Level Difference Texture Parameters.** For each pixel $p$ we take a square neighborhood $N$ (of size $15 \times 15$ in our case) centered on $p$. For each one of the eight offset vectors $\{v_{ij} = (i, j) : i, j \in \{0, 1, 2\}, i + j > 0\}$, we select the pairs of pixels in $N$ separated by $v_{ij}$, $S_{ij} = \{(a, b) : a \in N, b \in N, a + v_{ij} = b\}$, and calculate, for each such set of pairs of pixels, the histogram $h_{ij}$ of the differences between the intensity value at $a$ and at $b$. For each $h_{ij}$ we compute the mean, energy (also called second angular moment in this context), entropy, variance, skewness, kurtosis, contrast, and inverse differential moment (see [11]).

**RMSE for Neural Networks.** This error metric is computed as follows: for a validation set of $N$ samples, and a network of $K$ output neurons, let $R(n, k)$ be the output of the $k$-th output neuron on the $n$-th input pattern, and let $D(n, k)$ be the desired output in the same conditions. Then,

$$RMSE^2 := \frac{1}{N}\frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}\|R(n, k) - D(n, k)\|^2 \ . \tag{1}$$

Since neurons have an output in $[0, 1]$, the minimum RMSE is 0 (perfect learning), and the theoretically worst RMSE is 1.