# Automatic updating of urban vector maps

S. Ceresola, A. Fusiello, M. Bicego[⋆], A. Belussi, and V. Murino[⋆⋆]

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15, 37134 Verona, Italy

**Abstract.** In this paper we propose an automatic updating system for urban vector maps that is able to detect changes between the old dataset (consisting of both vector and raster maps) and the present time situation represented in a raster map. In order to automatically detect as much changes as possible and to extract vector data for new buildings we present a system composed of three main parts: the first part detects changes between the input vector map and the new raster map (based on edge matching), the second part locates new objects (based on color segmentation), and the third part extracts new objects boundaries to be used for updating the vector map (based on edge detection, color segmentation and adaptive edge linking). Experiments on real datasets illustrate the approach.

## 1 Introduction

A significant amount of work has been done in the field of aerial image processing in particular regarding the detection of buildings or other man-made structures such as roads, railways, and others [1–5]. These results could be of great help, if they were applied to updating data stored in a Geographical Information System (GIS).

Geographical data in a GIS are usually structured in different logical layers, each one containing a specific type of geographical features [6] (for examples, there could be a layer for roads, one for buildings and one for rivers). We call this set of layers as *GIS database*. Data in layers have been obtained from survey processes that required: a) the acquisition of aerial images of the territory of interest and b) the processing of these images by human experts and using specific optical tools in order to extract the vector representation of the features of interest. GIS applications requires up to date information in order to be effective, and this cannot be achieved without time consuming and expensive operations, if the above approach is applied; indeed, it requires to repeat the process from the beginning creating a completely new layer of vector data.

As suggested by some previous papers in the areas of GIS and remote sensing [7, 8], a layer of a GIS database, containing the vector representation of a set of

[⋆] Current Address: DEIR, Università di Sassari - via Torre Tonda 34, 07100 Sassari, Italy.

[⋆⋆] Corresponding author: Tel +39 045 802 7996, Fax +39 045 802 7068, E-mail `vittorio.murino@univr.it` features, can be updated by comparing the original aerial image (from which the vector representation was derived) against some new images of the same area. Now, the problem is: how can we reduce the cost of this operation? Obviously the cost for the realization of the new ortophotos cannot be avoided, however we can reduce the cost of the comparison operation between old and new data by adopting automatic tools that are able to automatically detect changes and update datasets avoiding human work. In this paper an automatic approach to GIS updating is proposed based on image processing techniques, namely edge matching, color segmentation and edge detection and linking.

### 1.1 Overview of the proposed approach

Our approach is divided in three phases. In the following, we call $V_{old}(L)$ the old vector representation of the layer $L$, $R_{old}$ the old ortophoto, or old raster map, and $R_{new}$ the new ortophoto, or new raster map. Moreover, we focus on features with a polygonal representation in vector format.

1. *Detection of feature boundary changes*: for each new raster map $R_{new}$ and for each layer $L$, we identify the known features of $L$ on $R_{new}$, by superimposing $V_{old}$ onto $R_{new}$. By comparing the edges of known features, and the ones obtained by a gradient analysis on $R_{new}$, we detect changes of old features. These changes are further validated by comparing $R_{new}$ against $R_{old}$ with a robust change detection technique. As a byproduct, we obtain a color range for each feature type by considering all the areas of $R_{new}$ where changes have not been detected.
2. *Finding location of new features*: for each new raster map $R_{new}$ and for each layer $L$ we identify new features of the type represented in layer $L$, by classifying $R_{new}$ using the color ranges stored for layer $L$.
3. *Recovery of vector data for new features and update of vector map*: for each new raster map $R_{new}$ and for each layer $L$ the edges of the new features identified by the previous phase are computed and stored in $V_{new}(L)$.

In this paper a direct application of this general approach to the buildings layer updating is proposed: preliminary experimental evaluation on real datasets shows promising results.

In the paper we analyze in detail each part of the system. In particular, Section 2 presents the proposed approach for automatic detection of changes and contains a subsection for each phase of the approach as above described. Section 3 illustrates the results obtained by apply this approach to real aerial images and GIS databases. Finally, Section 4 outlines conclusions and future works.

## 2 Method

### 2.1 Detection of buildings boundary changes

The method we used is inspired by [9]. The input data of the system are: the vector map at time $t$ ($V_{old}$), the raster image at time $t$ ($R_{old}$) and the raster image at time $t + \Delta t$ ($R_{new}$), where $\Delta t$ is the time lag between the old and the new image.

We need to compare old information with the newer one and to identify changes that occurred in the meantime. In our approach, this is equivalent to comparing an object as it is represented in an image captured at instance $t + \Delta t$ to the same object represented in a vector format at time $t$. In the case of buildings, the vector data represents the building outline.

The old vector information is retrieved from a GIS database. Using positional information from this database and geo-referencing information from the new image, we transfer the old object information onto the new image. $V_{old}$ produces edge templates that should be validated in $R_{new}$.

To make comparison more precise, we partition our vector polygons in smaller segments, according to the MSE (Minimum Spatial Element) criterion defined in [10], which defines the spatial resolution of changes.

We locate the gradient maximum value in the direction perpendicular to the edge template: we compare the position of this maximum value with the edge template position. If the difference is less than few pixels, vector data is validated, otherwise a possible change is detected. Before a change flag is raised, the detection must be confirmed on the raster image with a robust method for change detector based on the following ratio:

$$\lambda = \frac{\left[\frac{\sigma_1 + \sigma_2}{2} + \left(\frac{\mu_1 - \mu_2}{2}\right)^2\right]^2}{\sigma_1 * \sigma_2} \tag{1}$$

where $\mu$ and $\sigma$ are the mean gray value and the variance for the region being compared. Only if $\lambda$ is greater than a threshold, a change is detected.

This copes with the case when an edge of $V_{old}$ does not match with $R_{new}$, but no changes happened. Consider for example the case when vegetation occludes edges both in $R_{new}$ and $R_{old}$.

In short, we can resume step-by-step our method as follow:

1. Transferring the old vector information onto the new image.
2. Decomposition of the outline (old vector data) according to the defined MSE.
3. Calculating sum of gradient in the direction perpendicular to the edge.
4. Calculating peaks of gradient and their position.
5. Select the greater peak and compare its position with that of edge.
6. Apply the robust change detection method.

Please note that this part of our system is only able to detect changes in building already present in $V_{old}$. Namely, the following situations are catered for: buildings that have been enlarged/reduced, or building that have been pulled down.

The following section describe our strategy for locating *new* buildings, i.e., buildings constructed after time $t$.

## 2.2 Finding new buildings' location

Observing lots of aerial images, we can notice that buildings' rooftop in the same area have a good probability to have a similar color. We can therefore define a color range for buildings by computing the average color value of areas where no changes have been detected. This is particularly important since it allows an automatic definition of the range based on the present time image $R_{new}$.

The method for detecting new buildings is based on simple color image segmentation. As customary, we work in the HSV color space and use the chrominance (hue and saturation) information only.

The image is segmented by thresholding the H and S image channels. The threshold values $\theta_{\min}$ and $\theta_{\max}$ (one for each channel) are computed as follows:

$$\theta_{\min} = \mu - c, \qquad \theta_{\max} = \mu + c \tag{2}$$

where $\mu$ and $c$ represents the color range obtained from examples.

The result is then refined with morphological operations [11]. We can notice that buildings have generally a regular, and compact shape. By analyzing ratio between perimeter and area of blobs in the output binary image, we can select the areas that are reasonably roofs.

## 2.3 Recovery of vector data for new buildings and update of vector map

In this last part of the system, the vector map $V_{old}$ is updated to obtain the new map $V_{new}$. To this end we take into account the parts of $V_{old}$ that have not been validated (Sec 2.1), corresponding to changes to old buildings, and the new buildings detected in Sec. 2.2.

Let us start with the processing applied in the case of new buildings. The output of the previous stage is the geo-referenced location of areas containing new buildings. The goal is to obtain a polygonal representation for each building. The boundary of the building are located using both edge detection (Canny [12] operator) and the result of color segmentation from the previous phase.

The edges and regions found are then integrated using an adaptive method that fits a closed polygon. This has some advantages:

- it is a very natural way of combining edge detection and color segmentation results;
- it enforces the type of result that we are looking for, namely closed polygons;
- it does not impose unnecessary constraints on the shape a building could have.

Our fitting method starts with a bounding box, strictly containing the segmented region, decomposed according to the method used in Section 2.1. Then, the first segment is moved toward the center of the region until it reaches the boundary of the building. This condition is expressed by requiring that the segment should coincide with as much edge points as possible and should lie on a color boundary. In formulae, the following two conditions must be satisfied. The first is

$$\frac{1}{n}\sum_{i=1}^{n} E(x_i, y_i) > \lambda_1 \tag{3}$$

where $i \in [0, n]$ with $n$ number of points in the segment, $x_i$ is the abscissa of i-*th* segment point and $y_i$ its ordinate, $E$ the edge map produced by edge detection process and $\lambda_1$ is a threshold. The measure above express the rate of segment points that coincide with edge points. The second writes:

$$s_{i+1} \cong 0 \quad \text{and} \quad s_{i-1} \geq \lambda_2 \tag{4}$$

where $s_{i+1}$ is the rate of points just one pixel above the segment that belong to the segmented region, and $s_{i-1}$ is the rate of points just one pixel under the segment that belong to the segmented region.

When such a position is reached, the next segment is placed with the additional constraint that one endpoint must coincide with the previous, so as to form a closed polyline.

In the case of updating the representation of an already exiting building, only the region where the changing took place is considered, and only segments that have not been validated are moved.

## 3 Results

The proposed method has been tested using aerial images of the Davies city, California[1]. The images were digitized and orthorectified to remove distortion introduced from varying camera angles and distances. These images are taken as $R_{old}$, whereas the present images $R_{new}$, are obtained by manually editing the old ones.

An example of two images taken in input is shown in Figure 1(a) and (b), representing the old image $R_{old}$ and the present image $R_{new}$, respectively. Comparing the two images one can see that a house located in the bottom left of the image disappears in the present image, while another house appears in the bottom left of the central agglomerate of houses. A littler difference could also be noticed in the first and the fifth house of the second row from the top, which change their configuration. The vector data taken in input is relative to the historical image, and the goal is to update it basing on the new image, *i.e.* to find the disagreements described below. The result obtained with the application of our method is presented in Figure 2: yellows parts are confirmed vector data, the blue parts are new buildings not present in the vector data (i.e. not present in the historical images) and the red parts are the parts described in the vector data but that do not match with the present image.

One can note that obtained results are really satisfactory, all disagreements between older and new images have been found and coded in polygonal form, suitable for vector data updating.

[1] Available at `http://www2.dcn.org/orgs/orthophotos`.

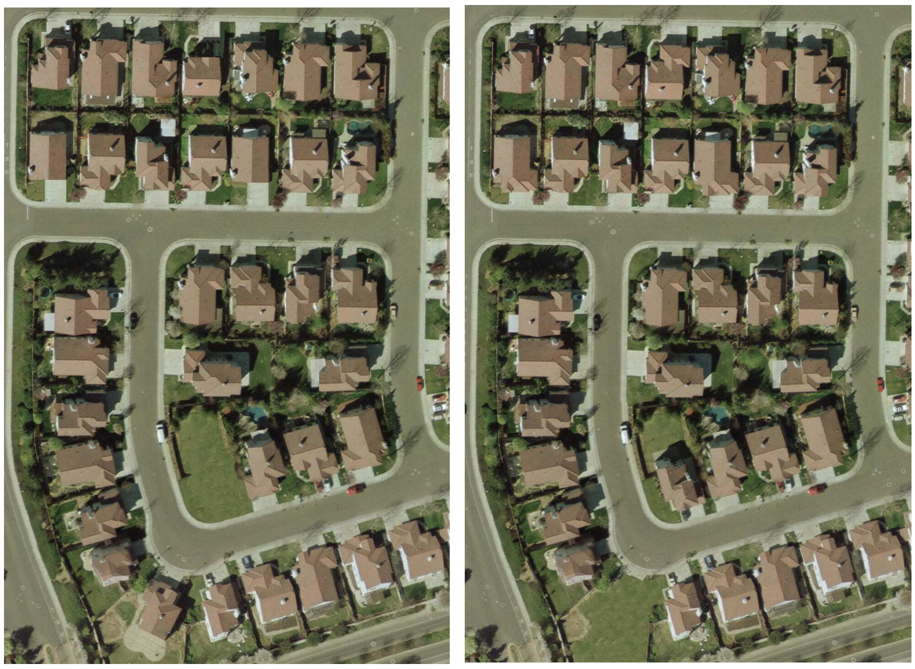
(a) (b)

**Fig. 1.** (a) Historical image; (b) present time image.

## 4 Conclusions

In this paper we proposed a complete semi automatic method to update vector maps of a GIS database. Detection of changes regarding a given feature type (i.e., roads, buildings, etc.) is performed by integrating vector and raster data. Features are located based on color segmentation and shape; moreover, vector outline of new features are extracted through an new adaptive edge linking method. The proposed method has been applied to a layer containing buildings and it has been tested on real high resolution images, presenting encouraging results.

## References

1. Meisels, A., Bergman, S.: Finding objects in aerial photographs: a rule-based low level system. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition. (1988) 118–122
2. Mayer, H.: Automatic object extraction from aerial imagery: A survey focusing on buildings. Computer Vision and Image Understanding **74** (1999) 138–149

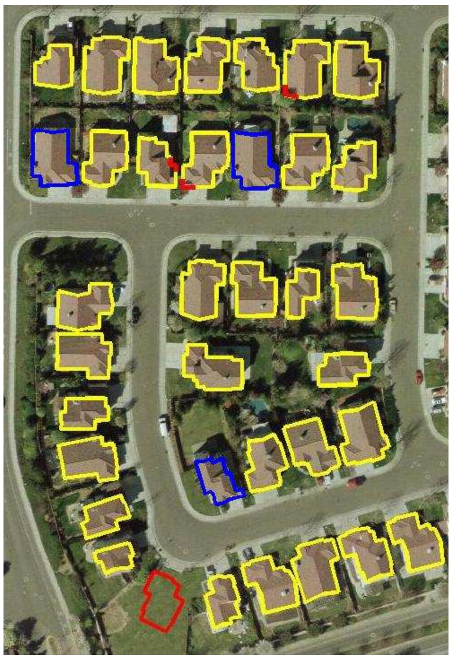

**Fig. 2.** Result.

3. Auclair-Fortier, M.F., Ziou, D., Armenakis, C., Wang, S.: Survey of Work on Road Extraction in Aerial and Satellite Images. Technical Report 247, Département de mathématiques et d'informatique, Université de Sherbrooke (2000)
4. Bicego, M., Dalfini, S., Murino, V.: Extraction of geographical entities from aerial images. In: Proc. of IEEE Workshop on Remote Sensing and Data fusion over Urban Areas (URBAN03). (2003) 125–128
5. Bicego, M., Dalfini, S., Vernazza, G., Murino, V.: Automatic road extraction from aerial images by probabilistic contour tracking. In: Proc. of IEEE Int. Conf. on Image Processing. Volume 3. (2003) 585–588
6. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufmann Publishers (2002)
7. Walter, V., Fritsch, D.: Automated revision of gis databases. In: Proc. of the ACM GIS 2000. (2000) 129–134
8. Walter, V.: Automatic classification of remote sensing data for gis database revision. International Archives of Photogrammetry and Remote Sensing **XXXII** (1998) 641–648
9. Agouris, P., Monuntrakis, G., Stefanidis, A.: Automated spatiotemporal change detection in digital aerial imagery. In: Proc. of SPIE Aerosense2000, Orlando, FL. (2000)
10. Mountrakis, G., Agouris, P., Stefanidis, A.: Navigating through hierarchical change propagation in spatiotemporal queries. In: Proc. of the IEEE Time 2000 Workshop. (2000) 123–131
11. Castleman, K.: Digital Imgage Processing. Prentice Hall (1996)

12. Canny, J.: A computational approach to edge detection. IEEE Trans. on Pattern Analysis Machine Intelligence **8** (1986) 679–698