Graph Matching in 3D Space for Structural Seismic Damage Assessment

Markus Gerke
University of Twente, Faculty ITC
Department of Earth Observation Science,
P.O. Box 217,
7500AE Enschede, The Netherlands
gerke[dot]itc.nl

Norman Kerle
University of Twente, Faculty ITC
Department of Earth Systems Analysis,
P.O. Box 217,
7500AE Enschede, The Netherlands
kerle[dot]itc.nl

Abstract

One common objective in computer vision and photogrammetry is to infer higher level object structure which is not directly observable in images or other sensing data. A practical problem field for such research is seismic building damage assessment. It is possible to observe objects such as façades, roofs, or rubble piles in oblique airborne images, but whether they are part of an actually intact or destroyed building is not observable directly: only the spatial relation between those directly observable objects allows conclusions about the structural integrity of a building. In this paper we present an approach to seismic building damage assessment, where a graph-based learning technique is employed to detect and to classify building damage levels, given instances of four object classes derived by supervised classification in object space. Results show that the vague building damage level description leads to relatively low classification score (52%), when a pre-defined building outline is assumed. However, if one is independent from such a pre-segmentation, the detection and classification rate is higher (70%).

1. Introduction and Related Work

One ambitious objective in photogrammetric engineering and computer vision science is to teach the computer to interpret a scene fully automatically, i.e. to infer object structure from iconic (image or range) observations. Already some 15 years ago the community was quite active, see for instance the proceedings of the SMATI workshops 1997 and 1999 [4, 3]. The idea behind semantic networks is to code relations between observable entities, e.g. using some expert knowledge, and to use those relations in a classification step to group a given set of objects into higher level object structures. For example, it is difficult to detect a car park automatically directly in airborne images. In a semantic network approach the context and relations to other – observable – objects and clues is represented, for instance that a parking is close to roads, and has a certain shape, minimum size and homogeneous cover. The disadvantage of such models is that they are quite inflexible: the relations are normally hard coded, which makes the models robust for some selected applications and test areas, but prevents them from being generic. An interesting study is presented in [16], where the idea of semantic networks is followed as well, though embedded in a probabilistic framework. An overview on current approaches is given in [13]. A similar idea to semantic modelling is pursued in so-called object-oriented image analysis (OOA), see [1] for an overview. In OOA relations and rules have traditionally also been hard-coded, but uncertainty in relation description and observed image evidence is accounted for by using advanced modelling techniques, e.g. based on fuzzy logic. In addition, recent work has focused on making the main aspects of OOA image segmentation and classification more objective, e.g. by employing thresholds automatically derived from the image data, or by using machine learning techniques for feature selection [18].

Other authors represent hierarchical relations through graphs. Early work includes the 1998 ICPR paper by Jia and Abe [9], where regions in an image were nodes in a graph, while node and edge labels and weights encoded geometric properties of the adjacent image regions. In a first phase, given some images of objects (taken from multiple directions), graph prototypes were trained, while in the classification step observed graphs were matched against those prototypes. A recent, more advanced approach from

In literature different terms are used, for instance in [13] ”semantic image analysis” refers to supervised classification techniques, and in contrast in papers from the geo-spatial domain, very often semantic modeling means that spatial relations or context in general is modeled to detect complex objects.
this category was presented by Raveaux et al. [17] who focused on optimising the learning phase. The advantage of graph based approaches, compared to semantic nets, is that they can be used in a supervised environment and thus are quite flexible with respect to object representation. However, when it comes to complex spatial relations between observed and target objects, a possibility to incorporate expert knowledge would be desirable.

In [8] a graph-based approach is presented where spatial relations between objects are modelled in a more sophisticated manner. An expert can describe how neighbourhood relations between parts of a complex objects may change, and those possible transitions are then considered during graph learning and classification. Given a number of observations, the approach computes a generalised graph that implicitly models all the possible spatial relations, and because it is embedded into a probabilistic framework, uncertainties and incomplete observations are accounted for, at least to a certain degree.

A practical problem field for such an image understanding approach is seismic building damage assessment. Here it is possible to observe objects such as façades, roofs, or rubble piles in oblique airborne images, but whether they are part of an actually intact or destroyed building is not observable directly: only the (spatial) relation between those directly observable objects allows conclusions about the structural integrity of a building.

In this paper we present results of a study where the general ideas behind the approach in [8] were adopted for a classification problem in the context of seismic damage assessment using oblique airborne images. We first give a brief overview on previous work and then put the mentioned graph-based approach into context.

**Background: Seismic damage assessment using high resolution oblique airborne images**

Following disaster events rapid damage assessment is important in order to provide rescue forces with substantial guidance information, or to support rehabilitation and reconstruction efforts. Optical remote sensing data are very often used as main source of information because of its independence from site access difficulties. In addition, optical images are relatively easy to interpret by operators. Traditionally, those remote sensing images have been acquired from a near-nadir perspective, leading to minimal occlusion effects. However, despite ever increasing spatial image resolution such vertical data continue to lead to relatively poor damage map accuracies, largely due to the complexity of structural damage and the limitation of mono-perspective imagery (see [7, 10]). However, with the increasing availability of multiple-view, oblique airborne images, such as captured by the Pictometry system [15, 14] a very interesting new data source has become available. In particular for seismic damage assessment the structural integrity of façades is an important indicator for building damage, and vertical structures such as façades are very well visible in oblique airborne images. In [7] a study is presented where Pictometry images captured over Port-au-Prince (Haiti) after the earthquake in January 2010 were used to assess individual building damage. Two main steps were involved: individual image classification and a per-building damage assessment. In the first step, the individual images were classified into the four classes Tree/Vegetation, Façade, Roof intact and Roof destroyed/rubble using a supervised approach (AdaBoost [5] combined with Conditional Random Fields [12, 11]). Image features used for the classification include spectral texture measures, but also from disparity images obtained thought dense image matching, 3D plane and colour features, as well as straight line information. Results show that an accuracy of up to 90% (Tree/Vegetation) can be obtained, while Façade was classified correctly by up to 60%. The second step presented in [7] consists in determining the damage level of a certain building, given the building footprint and the image-based classification results, i.e. without further direct use of image information. The motivation is that for the building damage assessment the knowledge about relations between façades, intact or destroyed roofs is important, and it seems reasonable to first extract those objects directly from the images and then infer the actual damage. Damage levels are defined in the so called European Macroseismic Scale (EMS98) and range from D1 (negligible to slight damage) to D5 (destruction). The description of damage indicators is quite vague and this vagueness is also reflected in the examples reported in [7]: even two human operators only agreed to 60% when using the oblique images to produce a per-building reference map of the 3 damage classes. The AdaBoost classifier was again used to decide on the damage level and an accuracy of approximately 60% was reached.

In order to exploit better the special geometry offered by oblique airborne images that show a tilt of 45° with respect to the horizon, in [6] an extension to the previous work is presented. The basic idea was to do the classification of the 4 observable classes Tree/Vegetation, Façade, Roof intact and Roof destroyed/rubble not in the individual images, but to use the three-dimensional voxel space: features as computed from single images were projected into object space and assigned to regular 3D-cubes (voxels). In addition, features from the dense-matching point cloud were computed directly in object space and assigned to the particular voxel as well. Especially the class Façade got detected much more accurately when the information from multiple images was merged in that way: the classification accuracy
reached more than 80%.

Graph matching in 3D space for building damage assessment  The results from [6] show that object representation in 3D space helps to increase automatic scene analysis results when the sensor data provide effective 3D information. Given the four classes being directly observable in the data, the interesting research question is now: how can we infer higher level object structures from them, i.e. in this case building damage level? An ideal technique to solve this problem would be able to

- train the algorithm on the (spatial) relations between input and target class, ideally using some statistic reasoning,
- detect and classify object instances given a set of observations,
- classify a pre-segmented set of objects into the target class.

The approach presented by Hartz in [8] seems to be interesting for this problem, because of the possible incorporation of knowledge about the spatial relations between input classes and their generalisation on the one hand, and the ability to learn the actual structure on the other hand. Components from [8] adopted here include: definition of possible spatial relations, including a pre-defined generalisation hierarchy, and derivation of matching costs. Instead of defining generalised graphs for each target class, we used a simpler matching approach, because the observed diversity in building damage classification and our relatively small sample size would severely limit any overall generalisation.

2. Method

We require a set of entities in object space, represented as voxels and being classified using evidence from sensor observations, 4 classes in our case. We call those classes ImageClasses to emphasise their origin, see [6] for details. Further we have a set of (reference) buildings or groups of buildings, and each instance of those buildings has a class label according to the building damage level: NoDamage, PartlyAffected and TotalDamage. Those levels are approximately equivalent to the EMS levels D(1-3), D4 and D5, respectively, see [7] for details. These classes are entitled BuildingDamage. The upper image in Figure 1 outlines of three building groups, captured from a pre-disaster map, are overlaid on a North-looking and orthorectified Pictometry image. The image is annotated with the respective labels indicating the damage level (BuildingDamage). The lower image shows the voxels of the same area, where the colour codes the respective class they belong to.

Given these data we attempted the following: 1) LEARNING: train a graph structure that represents spatial relations between the single ImageClasses instances for a given instance of a BuildingDamage; 2) DETECTION: for an area where only ImageClasses are given we wanted to detect and classify BuildingDamage structures; and 3) CLASSIFICATION: only for an area where ImageClasses and segmented buildings are given we wanted to assign one BuildingDamage to that segment.

2.1. Graph definition, spatial relations, and matching

A Graph $G(N, E)$ is defined as a directed graph where a node $N$ represents the connected component\(^4\) of voxels from a certain class within ImageClasses. The node label is equivalent to the particular class name. An Edge $E$ weight

\(^4\)See [19] for connected component segmentation using k-D tree data structure.
Fig. 2. Spatial relations, generalisation hierarchy and examples (colours for ImageClasses as in Fig. 1, colours of the graph edges as indicated in the spatial relations hierarchy tree)

...: the former relations can never be generalised to the latter one. Three example graphs are sketched in the bottom part of Fig. 2. For every class in BuildingDamage (same as in Fig. 1) the respective graphs are shown. The node position equals approximately the centre of gravity of the corresponding connected component of voxels. The colour of the edges encode the relation, and correspond to those used in the generalisation hierarchy diagram.

We use the concept of Subgraph-Isomorphism to define whether two Graphs $G_1$ and $G_2$ match:

"A mapping $M \subset N_1 \times N_2$ is said to be an isomorphism if it is a bijective function that preserves the branch structure of the two graphs, that is, $M$ maps each branch of $G_1$ onto a branch of $G_2$ and vice versa. $M$ is said to be a graph-subgraph isomorphism if $M$ is an isomorphism between $G_2$ and a subgraph of $G_1"$[2].

Nodes can only match if they have the same node type, i.e. they belong to the same class within ImageClasses, and the directed edges can only match if they are generalisable, i.e. they belong to the same generalisation branch. The matching cost per edge is derived from the smallest common generalisation depth and the generalisation depth of the original relations. See Equation 1[8], where $d_i$ are the depths in the input matching relations from both graphs within the generalisation hierarchy, $d_g$ is the depth of generalised relation, i.e. the closest common generalisation, and $d_{max}$ is the largest generalisation depth available and only being used to normalise the cost.

$$cost = \frac{(d_1 - d_g) + (d_2 - d_g)}{2 \cdot d_{max}}$$

The total matching cost $cost_{tot}$ is the sum of all edge matching costs involved and normalised by the product of matching nodes and matching edges.

2.2. Learning, detection, and classification

The learning phase aims at building trained graphs: given the reference map, first per-building instance all ImageClasses components are selected covering that particular instance. So far, we define the building outline only as a polygon in the XY-plane, hence we collect among all the ImageClasses instances the ones having a major overlap in 2D with a particular building outline. In a second step, a graph is computed following the graph definition explained...
above. Thus, per class in BuildingDamage we obtain a set of corresponding graphs, composed from the respective ImageClasses clusters.

For the detection we do not assume to have a map of building footprints, hence have an opportunity also to identify building groups or building clusters as belonging to a certain BuildingDamage instance. Given the trained graphs from the learning phase the task is to find sub-graphs in a graph which is built from all ImageClasses voxels to be clustered and classified (this graph is called Testgraph T below):

Detection algorithm
begin
foreach trained class do
  foreach trained graph do
    proc Graphmatching with Testgraph T.
    foreach matched subgraph ∈ T do
      foreach matched node do
        accu(node(class)) = m + 2^{-cost_{tot}}
      od
    od
  od
  foreach node in T do
    class(node) = max(accu(node(class)));
end

Whenever a node of the Testgraph is part of a matching sub-graph, an internal accumulation counter (accu in the algorithm) for this node and the particular target class is updated. This score depends on the number of nodes in the matching sub-graph (m) and also on the matching cost: the more expensive the match, the less the contribution to the accumulation counter. After all the classes have been processed, every node is assigned the final class label according to the maximum counter score.

In case a classification is desired we require a map indicating building outlines. The algorithm for classification is:

Classification algorithm
begin
foreach Building do
  proc Graph from ImageClasses: T.
  foreach trained class do
    foreach trained graph do
      proc Graphmatching with T.
      foreach matched subgraph ∈ T do
        accu(costs(class)) = cost_{tot}
        accu(# match_edges(class)) = M_{E}
        accu(# match_nodes(class)) = M_{N}
      od
    od
    proc Final normalised cost for this class.
  od
  proc Final class for building: min cost.
end

In contrast to the detection we set up a Testgraph for all ImageClasses instances within the outline of a building to be classified. We then search for the class which produces the least overall matching cost. The final normalised matching cost is computed:

$$
\text{cost}_{\text{final}} = \frac{\sum_{\text{costs}}}{\sum_{M_{E}} \times \sum_{M_{N}}}
$$

(2)

Not only the cost is considered, but also the total number of matching edges (M_{E}) and nodes (M_{N}). Very often no matching costs are generated, because the matching edges already represent the same spatial relation. In order to account for those cases as well, a very small constant artificial cost of 0.1 is added, in order to have a means to consider the number of matching edges and nodes. Otherwise, such a good match would not be taken into account.

3. Results

In our experiments we concentrated on the same test area as described in [7, 6]. Pictometry images were acquired over Port-au-Prince, Haiti, in January 2010 after the earthquake. For the analysis of the graph-based damage assessment we required as reference individual building outlines, and each building being classified into one of the classes in BuildingDamage. With respect to ImageClasses two sets of classified voxels were created: a) the reference classification, compiled by an operator, and b) the classification result as obtained through RandomTrees classification, see [6] for details.

In Figure 3 an overview of the data is presented. Interesting is the mismatch between the reference labelling of BuildingDamage from the two operators. As discussed in [7] this is mainly caused by the vague definitions given in the EMS98. Another interesting question in this context concerns the actual definition of Building. In this case the building outlines were manually digitized in pre-disaster satellite images, but the question is how to separate buildings within a city-block – or whether the whole city-block should be regarded as one building. In order to also have an
impression of the impact of those definitions, several building sizes were realised, see upper left image in Figure 3.

3.1. Detection and Classification given no pre-segmentation

In this section we present results from the detection approach, i.e. to identify building structures in the given ImageClasses instances complying with the known and trained BuildingDamage classes. Thus, apart from the training phase no information about the location and shape of (former) buildings was used here. The Southern part of the reference BuildingDamage classification of operator 1, see Figure 3, upper row centre, was used in the learning step to collect the respective graphs as constructed from the reference ImageClasses instances, see Figure 3, bottom row centre. Then the Northern part of the BuildingDamage reference classification was used to validate the automatic detection. The detected BuildingDamage structures are shown in the upper left image of Figure 4, while in the lower left image the assessment in the form of a red-green visualisation is shown: voxels from the input ImageClasses that were classified correctly are shown in green, the wrong ones in red and if no structure was identified, they are shown in white. The respective confusion matrix is displayed in the upper part of Figure 5. These results were obtained by using the reference ImageClasses classes for detection, while the lower confusion matrix in Figure 5 shows the assessment when the automatically extracted ImageClasses classes were used for the validation (training still done as above with the reference ImageClasses).

Figure 3. Top row: individual buildings where each building is displayed in a random colour, reference BuildingDamage: operator 1 and operator 2. Bottom row: ImageClasses: reference and extraction result, see [6]

Figure 4. Detection experiment: using reference buildings from operator 1, southern half for training. ImageClasses from reference, also for validation. Classes from validation shown in the upper part, assessment in the lower part. Right hand image shows a part of an east looking image, rotated to fit the left hand images. Image ©Pictometry, Inc.
When the automatically classified ImageClasses were used for the detection instead of the reference data (see confusion matrix at the bottom of Figure 5) the overall accuracy dropped to 60%, and the major difference compared to the result just described was in the correct detection of TotalDamage clusters. For example, using the reference ImageClasses, 88% of the detected TotalDamage clusters were actually correct, but when the automatically classified data were used, only 22% were correct. The reason for that is probably that from the supervised classification a tendency to classify clusters of voxels as Façade and Roof intact was observed, and those do not correspond to the building damage class TotalDamage. Some very interesting observations

Figure 5. Detection experiment: confusion matrices, upper one from experiment shown in Figure 4, lower one from experiment with the same training set up as above, but validation using the automatically extracted ImageClasses instances.

3.2. Classification given building outlines

The classification experiment was conducted in a similar configuration to the detection experiment described before, i.e. using the same BuildingDamage data and ImageClasses classes for training the graphs. The validation was then done twice as well: first, with the reference ImageClasses and subsequently with the automatically classified data. Figure 6 shows the results per building: upper part red-green visualisation and confusion matrix using the reference ImageClasses and likewise in the bottom part, but using the classified ImageClasses. The red-green visualisation of the upper part in Figure 6 and the detection result, displayed in Figure 4, correspond quite well: the large complexes detected and classified correctly in the first experiment were mainly also correctly classified here, when the building outlines were pre-defined. However, the over-

Figure 6. Classification experiment: using reference buildings from operator 1, southern half for training. ImageClasses from reference, also for validation (assessment in the upper part), validation using ImageClasses from actual classification see lower part. Below the corresponding confusion matrices are shown.
rence we only had two buildings classified as TotalDamage, whereas for the detection, we had more than 1000 voxels with that label. This means the assessment was much less robust in the classification case. Compare also the classification accuracy from the classification using the automatically extracted ImageClasses: it was only 37%, while the accuracy from detection with that data was 60%.

4. Conclusions and Outlook

The graph-based approach to structural damage assessment as introduced in this work is a flexible means to train the algorithm on the spatial hierarchy of building elements. In contrast to methods for classification where the object entity must be predefined, a detection of objects is possible in this approach as well. The results showed that the vague BuildingDamage definition has a significant effect on detection and classification accuracy. Already the (human) operators were not able to produce a homogenous reference map of building damage. However, the results were nevertheless promising: despite the fuzzy input of training data, the detection of building clusters led to quite homogenous and reasonable results. The classification was in addition biased by an unclear building outline definition. In dense urban areas individual buildings are difficult to identify, and in fact one building area may show multiple damage types.

In the future we will test the approach in a setup where we are less dependent on the given EMS98 damage levels. One idea is to pre-define possible spatial arrangements of façades and roofs, instead of training them, for instance related to the "survival potential": at least one or more Roof intact must be attached to one or more façades. This way areas can be identified where the chances to find survivors is high.

Acknowledgments

The graph matching was performed using the implementation released by the authors of [2].

References


