Automatic avalanche mapping using texture classification of optical satellite imagery

Siri Øyen Larsen¹, Arnt-Børre Salberg¹ and Rune Solberg¹

Norwegian Computing Center, Section for Earth Observation, Oslo, Norway; siri@nr.no

Abstract. Detection and characterization of an avalanche is important for making avalanche inventories as well as for the management of emergency situations. Problems related to field measurements include remote or inaccessible terrain, poor weather, and avalanche danger. Earth observation satellites therefore represent a potentially important source of information. We present a framework for automatic detection of avalanches in very-high resolution optical satellite imagery. The approach builds upon an initial texture segmentation stage using directional filters to enhance avalanche snow in order to separate it from other relevant area cover types, such as rough and smooth snow surfaces, trees and rock. The directional filters are oriented in the same direction as the terrain aspect, which is estimated from a digital elevation model. In the training stage, filter responses corresponding to the same texture class are used to form so-called textons. We then perform pixel based classification of the image based on the distribution of textons within a sliding window. In the second stage of the detection algorithm, pixels are grouped according to texture class to form image objects, which are finally post-classified based on extracted object features. We have assessed the mapping abilities of our algorithm on a set of QuickBird images of Norwegian mountain areas. The automatically derived avalanche maps have then been compared to manually drawn avalanche outlines made by experts. The preliminary results show that we are able to locate most of the fresh avalanches in the image with few false detections, but that the outline of the avalanche is not always adequately determined.

Keywords. Please include several keywords which characterize the content of your manuscript.

1. Introduction

Each year, snow avalanches hit populated areas and parts of the transport network in the Norwegian mountain regions, leading to loss of lives and the damaging of buildings and infrastructure. Detection and characterization of avalanches are important for making avalanche inventories as well as for the management of emergency situations. The state-of-practise routine for mapping snow avalanches relies currently on field measurements mapping the extent and location of avalanche-release zones and deposits by hand, by amateur photographs, or with a GPS device. Field measurements are typically conducted during the peak avalanche period of the year as snow depth and layers are also recorded. Problems related to this method are poor accessibility of the terrain due to avalanche danger. Only small areas can be surveyed, and surveys can only be conducted in good weather. Quite often, only avalanches causing accidents or resulting in heavy damages are mapped. Large parts of the Norwegian mountains are inaccessible to observers, especially during periods of high avalanche danger level. Earth observation satellites therefore represent a potentially important source of information.

Current approaches for remote-sensing based avalanche detection cover quite a broad range of applications both in view of applied techniques, covered scale ranges and used data. Small-scale applications are mostly focused on the exploitation of the capabilities of laser scanning or ground-based radar (e.g., [1]; [2]). These approaches are, however, not adequate to cover large areas or for the operational mapping of more than just a few selected targets. Previous research regarding automated detection of avalanche deposits with the help of optical aerial imagery has been carried out
and published by Bühler et al. [3]. However, in a commercial setting, airborne data would be inappropriate due to acquisition costs. Lato et al. [4] present an object-oriented approach for segmentation and classification of avalanches in both aerial and very-high resolution (VHR) satellite images. This study was carried out using the software package Definiens eCognition, which is commercial off-the-shelf image analysis software. It demonstrates the capability within the usage of VHR optical data for avalanche mapping, although it is too sensitive towards noise when the image contain other textures than avalanche snow and smooth snow (such as areas characterized by snow and sparse deciduous trees, i.e., sparse forest). Our goal is to develop an algorithm that is robust even in complex terrain, including other textures than just smooth snow and avalanches, such as sparse forest, rocks, and open water. We will not discuss the potential use of radar or SAR data here, as we limit our scope to optical sensors in this paper. An earlier stage of this work was presented in [5] and [6].

Avalanches are visually recognized in the satellite images as rough snow surface characterized by a line-like pattern, where the lines are oriented in the aspect direction of the terrain. Other relevant surface types that appear in avalanche prone terrain include smooth snow, wind disturbed snow, rock, and vegetation. Our hypothesis is that avalanches may be automatically detected by the use of texture analysis. The basis for our method is therefore a texture segmentation step, where the textures are defined using the image response to directional filters, in combination with a digital elevation model (DEM) that can be used to estimate the local aspect direction across the image. Extracting textural features by convolving the image with a given filter is often applied in texture segmentation and classification (see e.g., [7] – [10]). Typically, a set of convolved images are created by applying a bank of filters, each with given characteristics (e.g., scale, orientation, frequency, etc.). Then each filtered image is combined into a multi-dimensional image, which is further analyzed to stratify the image into segments with similar texture patterns.

2. Methods

2.1. Texture classification

The first (and main) part of our approach involves texture classification, and is based on the work by Varma and Zisserman [7], who classify single texture images using the image response to different filter banks, among which we find the Maximum Response (MR) sets. The algorithm is divided into a training stage and a classification stage. In the training stage, exemplar filter response vectors are extracted via clustering from the filtered training images. These vectors – called textons – are next used to label every pixel in the training images. The histogram of the texton image is then used as a model for the corresponding texture class. In the classification stage, the same procedure can be followed to generate a filter response image, a texton map, and finally a histogram model, and the image can then be classified by comparing its model to the models of the training images.

In our application, we do not have single texture images, but large images that each includes many different textures. We therefore have to adjust the setup for texture classification accordingly. Furthermore, we have adjusted the filter bank to be more suitable for our specific application. This will all become clear as we describe our algorithm in detail in the following text.

2.1.1. Filter bank

The filter bank we apply is based on the so-called MR8 filter bank [7]. The MR8 filter bank consists of 38 filters: Two rotationally symmetric filters (a Gaussian and a Laplacian of Gaussian), and 36 anisotropic filters (an edge and a bar filter, both at six orientations and three scales). However, the output from the oriented filters are collapsed by recording only the maximum response
across all orientations (for each filter at each scale), reducing the number of responses to eight (three scales for two filters, plus two isotropic).

For the filter bank, we use the 36 oriented filters as above (Figure 1). However, instead of recording the maximum responses, we select the filter response corresponding to the aspect of the terrain. This choice is based on the appearance of avalanches (which is the main texture that we want to discriminate) in the image. An avalanche can typically be seen in the image as a texture pattern in the snow surface characterized by linear structures in the same direction as the aspect of the terrain. Furthermore, since trees and tree shadows are oriented vertically in the image, sparse forest (trees scattered on a snow covered ground) is often characterized by vertical lines in the snow. Therefore, we always include the response from the vertically oriented filters in addition to the aspect orientation, yielding a recorded filter response in 12 dimensional image space (Figure 2). The selected scale parameters for the bar and edge filters at three scales are \((\sigma_x, \sigma_y) = \{(0.75, 0.25), (1.5, 0.5), (3.0, 1.0)\}\).

**Figure 1.** Oriented bar and edge filters at six orientations and three scales.

**Figure 2.** Comparison of filter responses from two texture classes (avalanche and sparse forest) at both aspect and vertical orientation. Avalanche snow is enhanced in the aspect direction, and not in the vertical direction, while sparse forest is more enhanced by filtering in the vertical direction.
2.1.2. Training stage

In order to produce a ground truth for classification, we first select texture samples for each of the chosen texture types or content classes: avalanche, smooth snow, rugged snow, sparse trees, and rock. All the samples are taken from the same satellite image. The texture samples are found by drawing a few regions of interest (ROIs) in the image. (We had between six and twelve ROIs for each texture class, each ROI consisting of 1–9000 pixels, from an image of totally 6480×11576 pixels.) We do not select ROIs corresponding to dense forest and agriculture areas since forest and agriculture masks are available.

In the training stage, each ROI is convolved with a filter bank to generate filter responses. The filter responses corresponding to the same class are then clustered using a K-means clustering algorithm, and the resulting cluster means are chosen as textons (Figure 3). For each class, ten textons are generated, resulting in a dictionary of a total of 50 textons. The corresponding cluster covariances are also estimated, and each pixel in the training ROIs are labelled by classifying each pixel to one of the textons using a maximum likelihood classifier based on a Gaussian distribution. The histogram of texton frequencies is then used to form models corresponding to the training ROIs (Figure 4). The histograms are normalized to sum to unity.

2.1.3. Classification stage

In the classification stage, a new image is classified pixel by pixel by treating the local neighbourhood around each pixel as a single texture sample image. To do this, we first perform filtering and texton mapping for the entire image, and then we classify each pixel based on the histogram model created from a neighbourhood around the pixel. This strategy will naturally not be optimal at the border pixels between two image regions with different textures. However, since the pixel size is very small compared to the typical (single) texture regions in our application, this is not a problem in practice. The classification itself is performed by selecting the nearest neighbour among the models computed in the learning stage. Distances between histograms are measured using the $\chi^2$ statistic.

![Figure 3. Generating the texton dictionary: The texture samples (from the original image) are filtered, and the multi-dimensional filter response image is then clustered to extract textons, representing the corresponding texture. Finally, all the textons are collected in a texton dictionary.](image)
Figure 4. Generating the model database: The texture samples (from the original image) are filtered, and then each pixel location in the multi-dimensional filter response image is mapped to the closest texton from the texton dictionary (see Figure 3), producing a texton map, from which the image histogram is extracted and put into the model database as a representative for the corresponding texture class.

2.2. Object extraction and classification

The texture classification approach described in Section 2.1 gives us a pixel-based classification of the image. The purpose of object classification is to locate individual avalanches and remove outliers and false alarms.

First of all, each connected group of pixels classified as avalanche snow during texture classification is considered as a potential avalanche object. We then compute a set of features describing the objects, and use these features to build a rule-based classification scheme that separates avalanche objects from false alarms (non-avalanche objects). The features that were chosen for classification are listed and described in Table 1.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>Number of pixels</td>
</tr>
<tr>
<td>Aspect direction difference</td>
<td>Absolute difference btw. mean aspect orientation and object orientation</td>
</tr>
<tr>
<td>Area-perimeter ratio</td>
<td>Object area (number of pixels) divided by object perimeter (number of border pixels)</td>
</tr>
<tr>
<td>Bounding box width</td>
<td>The bounding box is the smallest rectangle that covers the object</td>
</tr>
<tr>
<td>Area-bounding box area ratio</td>
<td>Object area divided by bounding box area</td>
</tr>
<tr>
<td>Co-occurrence mean</td>
<td>Mean value of the co-occurrence mean image*</td>
</tr>
<tr>
<td>Co-occurrence correlation</td>
<td>Mean value of the co-occurrence correlation image*</td>
</tr>
<tr>
<td>Co-occurrence entropy</td>
<td>Mean value of the co-occurrence entropy image*</td>
</tr>
</tbody>
</table>

* These texture co-occurrence measures are calculated from the image based on gray-tone spatial dependencies as described in [11].
2.3. Image data

Our experimental data set consists of three QuickBird images from different mountain regions of Norway: Hellesylt, acquired April 16, 2005 (Figure 5), Dalsfjorden, acquired April 3, 2005, and Eikesdalen, acquired April 13, 2011 (Figure 6). All of the images were acquired during the peak season for avalanches, and contain several fresh avalanches, in addition to some older traces of avalanche activity. All of the texture samples used for the training stage were extracted from the Hellesylt image. Nevertheless, we find it useful to assess the overall performance of our algorithm on this image, since only small parts of it has been used for training.

![Figure 5. Hellesylt panchromatic QuickBird image. Overlay: forest mask (dark green), agriculture mask (brown), and selected samples for the texture classification training stage (see own table for the meaning of the different colors).]
3. Results

Classification results from both the texture classification and the object classification are shown in Figure 7, 8 and 9, for the three scenes Hellesylt, Dalsfjorden, and Eikesdalen, respectively. Some close-up results are shown in Figure 10 and 11.

The results have first of all been visually assessed, as the manually drawn avalanche maps are not suitable for quantitative assessment at this point, due to: 1) both old and fresh avalanches have been mapped, and the algorithm have not been developed to recognize old (and much less visible) avalanche texture, and 2) for one of the images (Dalsfjorden) the avalanche map was made for the image before orthorectification, thus a pixel-to-pixel comparison is meaningless, since perfect match between the mask and image is not available. (The images must be orthorectified before processing, in order to calculate the correct aspect direction from the DEM).

As can be seen in the images, the overall results of texture classification are good. All the fresh avalanches contain pixels classified as avalanche snow, and in most cases, the fraction of pixels classified as avalanche within the mapped avalanche is high enough for the avalanche to be no-
noticeable in the result. The object classification stage successfully removes many false detections, although it also makes a few mistakes (e.g., the large avalanche deposit area in the upper left hand section of Figure 7).

Figure 7. Results, Hellesylt QB image. Left: Original image, manually mapped avalanches in red (Source: Norwegian Geotechnical Institute). Middle: Texture classification result. Right: Object classification result.

Figure 8. Results, Dalsfjorden QB image. Left: Original image, manually mapped avalanches in red (Source: Norwegian Geotechnical Institute). Middle: Texture classification result. Right: Object classification result.
Figure 9. Results, Eikesdalsvatnet QB image. Left: Original image, manually mapped avalanches in red (Source: Norwegian Geotechnical Institute). Middle: Texture classification result. Right: Object classification result.

Figure 10. Results, sub section of the Hellesylt image. The background image is the original image. Overlays from left: manually drawn avalanche map, texture classification result (only showing what was classified as avalanche texture), and object classification result (avalanche objects in red, non-avalanche objects in orange – small objects have been removed pre-classification).
4. Conclusions

We have presented an approach for automatic detection of avalanches in VHR optical satellite images. The approach is based on texture classification of the image, where avalanche texture has been enhanced using edge and bar filters oriented in directions corresponding to the aspect of the local terrain, as it varies across the image. An object-based post-classification step was introduced to remove false detections. The results show that we are able to detect most of the fresh avalanches in the image, while keeping the number of false detections low. However, the texture segmentation tends to produce fragmented objects, so that the shape of the avalanche is not properly captured.

Figure 10 illustrates a fragmented avalanche object. When we turn to object classification, dealing with fragmented objects is challenging, as many of the properties that are used to recognize an avalanche visually (such as orientation and length in aspect direction) rely on well-defined objects. Still, we see how the object classification stage is useful for cleaning up the texture classification result. An example of this is shown in Figure 11, where all the false detections (sparse forest classified as avalanche snow) from the texture classification stage, are removed in the object classification stage. Areas of high entropy (such as sparse forest) may be classified as avalanche snow during texture classification, but then, looking at geometrical and other features related to the context, the object classification is able to recognize that these areas are not avalanches. The results shown in Figure 11 indicate a clear improvement compared to the results presented in [4], where the same image was analyzed.

We should mention that there is no unique answer to how the outline of an avalanche should be defined, as the avalanche cannot be seen as a well-defined structure in nature. There are sliding transitions between different textures in any natural surface, and even two different avalanche experts will not draw the exact same outline of an avalanche event. However, post-processing the texture classification result in order to make the avalanche objects less fragmented, will likely improve the object classification. We also suggest applying some sort of merging algorithm in order to connect avalanche objects that belong to the same avalanche path. Here, typical context-based rules should be applied. For instance, localizing the most likely avalanche trajectories based on a terrain model could assist the merging process, in that objects in the same probable path and close in distance should likely be assigned to the same avalanche. Furthermore, a re-estimation of the avalanche boundaries may be performed by morphological operations on the image.
References


