Automatic Auroral Detection in Color All-Sky Camera Images

Jayasimha Rao, Noora Partamies, Olga Amariutei, Mikko Syrjäsuo, and Koen E. A. van de Sande

Abstract—**Every winter, the all-sky cameras (ASCs) in the MIR-ACLE network take images of the night sky at regular intervals of 10–20 s. This amounts to millions of images that not only need to be pruned, but there is also a need for efficient auroral activity detection techniques. In this paper, we describe a method for performing automated classification of ASC images into three mutually exclusive classes: *aurora*, *no aurora*, and *cloudy*. This not only reduces the amount of data to be processed, but also facilitates in building statistical models linking the magnetic fluctuations and auroral activity helping us to get a step closer to forecasting auroral activity. We experimented with different feature extraction techniques coupled with Support Vector Machines classification. Color variants of Scale Invariant Feature Transform (SIFT) features, specifically Opponent SIFT features, were found to perform better than other feature extraction techniques. With Opponent SIFT features, we were able to build a classification model with a cross-validation accuracy of 91%, which was further improved using temporal information and elimination of outliers which makes it accurate enough for operational data pruning purposes. Since the problem is essentially similar to scene detection, local point description features perform better than global- and texture-based feature descriptors.**

Index Terms—**Aurora, classification, scene detection, vision.**

I. INTRODUCTION

AURORAL LIGHT is a result of charged particles interacting with the Earth's upper atmosphere. Guided by the Earth's magnetic field, particles of magnetospheric and solar origin deposit their excess energy at the altitudes of about 100–200 km on circular zones around the magnetic poles. Analysis of the dynamics of the aurora thus provides an opportunity for ground-based monitoring of processes in the magnetospheric source region as well as the near-Earth responses to space weather events.

MIRACLE network of all-sky cameras (ASCs) in Fennoscandia and Svalbard [13] takes pictures of the sky in regular intervals throughout the winter months (October–March). The camera network in 2014 includes nine stations and

three different imager types [23]. Five stations are equipped with camera systems with optical filters and filter wheels whose technical details can be found in [24]. Four camera stations use full-color imagers [1]. We use these color image data that are described in Section II. The common field-of-view of the ASC network covers the entire land area of Finland, most of Lapland, and Svalbard archipelago.

The camera network captures a few million images annually, and manual analysis of these images is extremely impractical. Traditionally, keograms have been used to visually identify auroral events. Keograms are created by extracting vertical pixel columns from individual all-sky images and placing the columns side by side. The horizontal axis is the time and the vertical axis is the geographical latitude [12]. Fig. 1 is a sample keogram constructed from the images taken during the night of February 26/27, 2012. Keograms can be used to approximate auroral activity during an entire night without having to analyze individual images. However, it is inaccurate and time-consuming, as the original images may still need to be perused. Thus, there is a need to prune the data that is captured, and also automate the process of detecting auroral events in the image data for further analysis.

The primary question for the auroral image data providers is to determine the periods of aurora in the data. Methods for automated classification of auroral images have been reported by Syrjäsuo and Donovan [14]. More recently, Syrjäsuo and Partamies [15] reported the results of a systematic comparison of different numeric features extracted from gray-scale (optically filtered) auroral images. The best performance was achieved using supervised learning and local and brightness invariant features. They concluded that the achieved classification errors in the range 6–8% are acceptable for operational data pruning. In contrast, Wang *et al.* [16] applied manual preprocessing of the data first and then used local features for classifying images of the dayside aurora. However, this approach is not optimal for large-scale operational use, where manual pruning of new images would still be required. To the best of our knowledge, color auroral image classification attempts have not been reported, although the potential has been highlighted earlier (see, e.g., [1]).

An overview of different stages involved in the classification of images into the three different classes is depicted in Fig. 2. The following sections will describe each of the involved steps in more detail, and finally, we present the results of the performed experiments.

II. DATA

The input images are captured by an upward-looking CCD camera with wide-angle optics for a circular field-of-view of

Fig. 1. Sample keogram from February 26/27, 2012. Horizontal axis is Universal Time (UT) and vertical axis is geographical latitude, zenith angle, or north–south distance from the camera station. One can detect auroral activity at any given hour without the necessity of any additional data.

180°. A computer controls the image acquisition and captures images of the night sky.

The raw data are 16-bit in 4 channels (cyan, magenta, yellow, and green). In operational use, our nominal data product is 8-bit JPEG images, which is what is used for classification. This allows us to classify all color-converted and archived data. In auroral imaging, the typical image cadence is 10–20 s and the exposure time is 3–4.5 s. The aperture of the cameras is f0.95. Because these instruments capture the whole visual wavelength band, any additional illumination affects the brightness and the color balance of the images [25]. The camera settings have been optimized for auroral imaging and will not change during the winter. The exposure may be slightly adjusted when the imaging starts in the autumn but is mostly unchanged over several years. This camera type does not have campaign modes but is continuously imaging throughout the winter season for auroral monitoring purposes.

The CCD employs a color filter array, which is integrated on the actual image sensor. As a result, neighboring pixels are captured through optical passband filters corresponding to cyan, magenta, yellow, and green colors. The knowledge of the filters' passbands and respective arrangement is used in color synthesis, which is required to convert a raw image into a final color image. The final image comprises red, green, and blue color channels. Syrjasuo et al. [1] describe the method used for image capture and conversion to RGB space in detail. Of the strongest auroral spectral lines, the CCD is most sensitive to green wavelengths ($\lambda \approx 558$ nm).

Fig. 3 shows some sample images from the stations at Kevo (69.76N, 27.01E) and Muonio (68.02N, 23.53E) in Northern Finland in varying environmental conditions. Due to the wideband color channels, the captured colors depend strongly on the background illumination, which often determines the resulting color balance. Any image with the moon may become challenging because the moonlight scatters from the clouds and is often brighter than aurora diluting the colors. Furthermore, due to wide-band color channels, any artificial light source (e.g., street or car lights) can make the faint auroral light difficult to detect. Insensitivity to uncontrollable color tints in images as well as absolute brightness is clearly required. Features are further
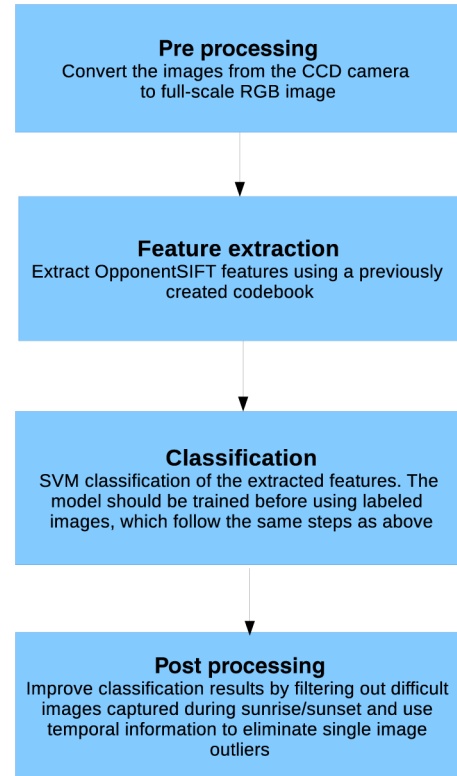

Fig. 2. Schematic representation of the overall process involved in automated classification of ASC images.

required to be insensitive to the unchanging obstacles in the images, such as tree tops or buildings close to the horizon [e.g., Fig. 3(d)]. Additionally, images required for building the supervised model are manually labeled, and hence are prone to inaccuracies in labeling.

III. FEATURE EXTRACTION

Images are normally represented as three-dimensional matrices, namely, height, width, and color channels. In order to lessen the computational effort, the first step in building a statistical classifier is usually to reduce the data dimensionality. This can be accomplished by representing each image with a feature vector that closely represents the interesting characteristics of the image. There are a large number of feature extraction techniques that have been extensively used over the years for a wide range of applications. The main objective in choosing a particular feature extraction technique over the others is its ability to preserve the differentiability of the images with the least number of variables. van de Sande et al. [2] provide a comparative survey on various color features that help in the task of classification.

Image features can be broadly classified into two kinds: 1) *Global features*, which encode the global visual content of an image, e.g., histograms, moments, as well as simple statistics such as mean brightness; 2) *Local features* or descriptors, on the other hand, which describe an image using local properties centered around some sparse set of key points, e.g., texture-based features, neighborhood descriptors, and any of the global features determined using a selected area in the image.
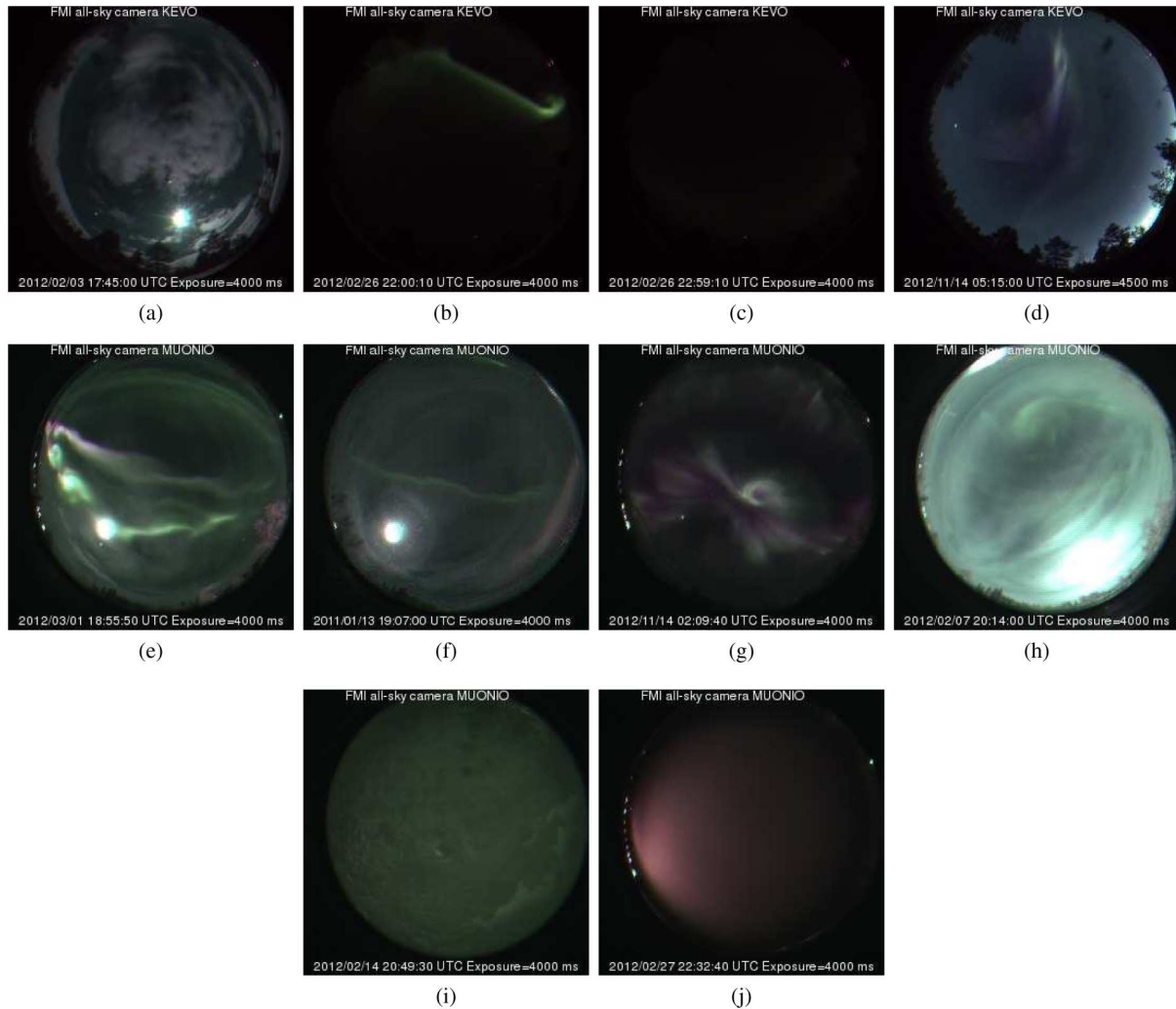
Fig. 3. Sample images from stations at Kevo and Muonio. (a) has no auroral activity but bright full moon and patches of cloud, (b) has a prominent auroral arc, and (c) has a very faint auroral arc, while (d) has a dispersed aurora but not just green, but with a red tinge. (e) and (f) show prominent and faint auroral arcs, respectively, with a bright full moon in the background. (g) shows a very prominent red-tinged aurora. (h)–(j) show examples of particularly difficult images, with high clouds and full moon, snow occlusion, and light pollution with dense cloudy conditions.

Global features are computationally efficient. For example, in auroral keograms, each individual image is represented using a simple feature one pixel column in the middle of the image. As already noted, this is not sufficient for accurate (visual) classification into aurora and no aurora classes. Clutter, occlusion, viewpoint changes, and illumination often necessitate re-examining the original images. However, when using local features, one can construct a more accurate representation of the image. More computational effort is required to identify salient properties of the scene but one often gains robustness. Usually, several local features are combined into an aggregate feature, which then represents the whole image.

In this section, we discuss the local features we used for auroral image classification. We also highlight selected results from the literature.

A. Local Binary Patterns

Texture-based feature descriptors are one of the most popular methods in recent years. Local binary patterns (LBP) is one such texture-based method which has gained widespread approval.

Due to its computational simplicity, LBP has been applied in many applications including texture classification, face recognition, facial expression recognition, as well as daytime auroral classification.

1) Original LBP Operator: The LBP texture analysis operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. The basic idea is to build a binary code that describes the local texture pattern by applying a threshold to the neighborhood by the gray value of its center. Using a circular neighborhood with P neighbors at a radius R [17], the LBP code of a pixel at (x_c, y_c) with a gray value g_c is calculated as

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c) * 2^p \qquad (1)$$

where g_p is the gray value of the neighboring pixel p and $S(x) = 0$ if $x < 0$ else $S(x) = 1$. Since the LBP operator is defined for a gray-scale image, using it as is would discard all color information, which is vital in detecting auroral activity in images.
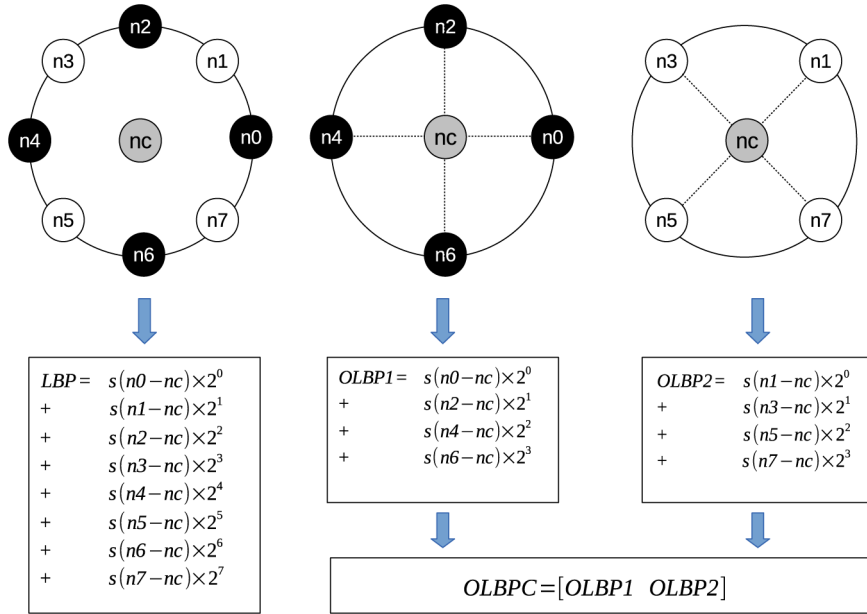
Fig. 4. Calculation of OLBPC with eight neighboring pixels (adapted from [3]).

2) LBP for Color Images: One of the most straightforward ways to incorporate color into the LBP descriptors is to apply the LBP operator to individual color channels and use the combination as the feature vector. However, this creates another problem. With the neighborhood radius $R = 1$ and $P = 8$, the histogram would have 256 elements per channel, which would lead to a final feature vector with a length of 768. In order to train a statistical model to account for this huge number of variables, the amount of training images needed would be enormous. We can try and reduce the dimensionality by choosing only $P = 4$ neighbors; however, this results in loss of data. A solution suggested by Zhu *et al.* [3] is to use *orthogonal LBP combination* (OLBPC), which preserves the differentiability and reduces the number of dimensions.

OLBPC works by computing the LBP histograms twice, with four neighbors each time. Here, we first compute the LBP for $P = 4$ and then apply LBP again, but this time to the diagonal neighbors which were missed by the first operator. Fig. 4 gives a schematic representation of the OLBPC operator. This technique will result in a much shorter histogram (32 per channel) while still accounting for all the neighborhood pixels, hence keeping the discriminative power. Zhu *et al.* [3] claim that OLBPC performs better than other LBP dimensionality reduction techniques.

B. Scale Invariant Feature Transform

Point description techniques work by describing local neighborhood relations around representative keypoints. These points can be selected based on image features or by other sampling methods. Scale Invariant Feature Transform (SIFT) is a point description technique that is widely in use because of its high discriminative property.

1) Classical SIFT Features: SIFT features were first proposed by Lowe [4]. They capture the local shape around a point using edge orientation histograms. The SIFT feature extraction process follows four key stages:

1) scale-space extrema detection—achieved by constructing a scale-space using a Gaussian kernel;
2) keypoint localization—apply contrast and edge response elimination to reduce the number of salient key points;
3) orientation assignment—assign a consistent orientation to the keypoints based on local image properties;
4) keypoint descriptor—set of histograms over a window centered on the keypoint.

SIFT features are scale, rotation, and translation invariants. Partial illumination invariance can be achieved using normalization of the extracted features. Additionally, the extracted features are also shift invariant with respect to light intensity.

2) Color SIFT Features: The original SIFT features work with single channel images as they only consider intensity values. However, there have been many extensions suggested for extracting SIFT features for color images. In addition to using red, green, and blue (R, G, B) values, a color can be represented by hue, saturation, and value (HSV; intensity). In this HSV color space, the color content is described by the first two channels (hue and saturation), while the intensity is separated into the third channel (value). *HSV-SIFT* [5] is one of the earlier approaches for using SIFT for color images. It computes features over all three channels of the HSV color space. HSV-SIFT features are scale invariant and shift invariant with respect to light intensity. However, they are only partially invariant to light color changes.

Opponent SIFT describes all the channels in the opponent color space (2) using SIFT features from the original color components R, G, and B

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}. \tag{2}$$

The O_3 channel contains the intensity information, while the other channels describe the color information in the image.

However, the channels O_1 and O_2 also have intensity information; hence, they take relative brightness changes into account. However, they are insensitive to absolute intensity, hence suitable for use with auroral images.

Transformed color SIFT or *rgbSIFT* extracts SIFT descriptors on a normalized RGB space as follows:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix} \tag{3}$$

where μ and σ represent the mean and variance of the channel intensities, respectively. The normalization of the pixel value distributions independently of each other makes the features extracted scale invariant, and also invariant to changes in light, color, and arbitrary effects.

C. Salient Point Sampling

Local feature extraction involves describing local features around specific salient points or points sampled using a grid. Salient or interesting point sampling is usually used for object recognition related applications, as the points can be used to describe the shape of an object. Grid-based point sampling is more suited for scene-type classification. Salient point extraction techniques include Harris-Laplace and Laplacian detectors [18].

As auroras do not have specific shapes, extracting salient points would not help classification. Thus, we use dense sampling with a fixed interval of six pixels with a sampling scale of $\sigma = 1.2$. Dense sampling is an example of uniform sampling over the image grid with a fixed pixel interval between points [11].

D. Codebook-Based Classification

Point description techniques such as SIFT work by describing interesting points in the images. As there are multiple salient points in each image—a regular point grid in case of auroral images, the resultant is still a two-dimensional matrix. However, to perform classification, having a single vector represent an image is more convenient. Codebooks are used to accomplish such purpose. Codebooks are constructed by clustering the features from the set of training images using methods such as k-means [11]. Other clustering techniques could also be used; however, k-means is widely preferred because of simplicity and is also effective [20].

E. Selective Weighting of Channels

Auroral events are predominantly green. Hence, it is intuitive to apply different weights for the color channels. This weighting changes the contribution of each of the color channels to the feature vector. Intuitively, assigning more weight to the green channel could increase the differentiability of the feature vectors and hence improve the classification accuracy.

During the extraction of the color SIFT features, we can assign a higher weight to the green channel. The simplest approach would be to consider just the green channel. An alternative would be to assign a higher weight to the green channel, while using the
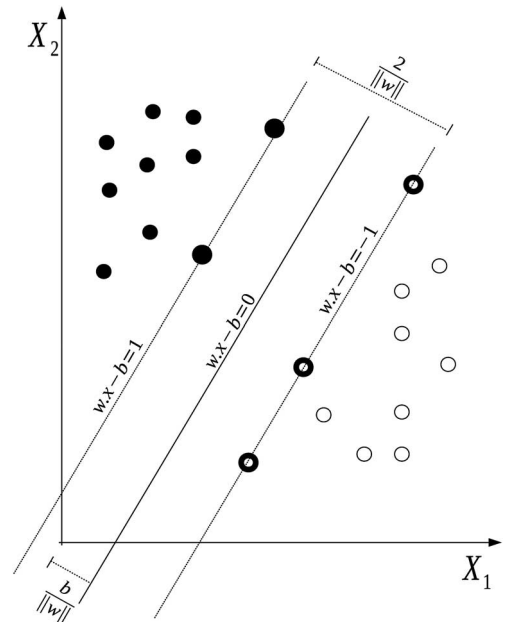

Fig. 5. Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

presence of the other two channels as counter evidence. For example, the tuning could be $2G - R - B$, where G, R, and B are the SIFT features for the green, red, and blue channels, respectively. In our experiments, we use an undocumented option from the color descriptors software package released as part of [2] to extract the selectively weighted features.

IV. SUPPORT VECTOR MACHINES

Support vector machines (SVM), specifically kernel SVM, is the classification algorithm that is used by most of the state-of-the-art systems. It is a binary classifier that models the difference in the two classes. It can easily be extended to multiclass classification by performing classification between pairs of classes and consolidating the results.

A. Linear SVM

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Fig. 5 gives the schematic representation of an SVM model.

Given a training set D, a set of n points of the form

$$D = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in -1, 1\}$$

where y_i is either 1 or -1, indicating the class to which the point x_i belongs. Each point is represented by a p-dimensional real vector. We want to find the maximum-margin hyperplane that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyperplane can be written as the set of points x satisfying

$$w.x - b = 0$$

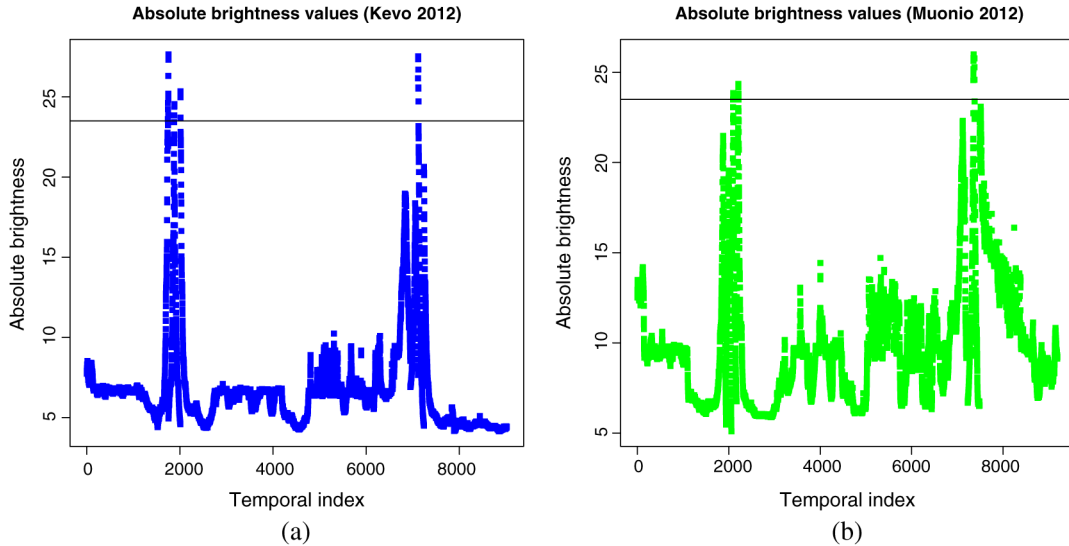where w is the normal vector to the hyperplane and b is an offset.

Fig. 6. Average color histogram values for two stations over a period of two nights (November 13 and 14, 2012). Two clear peaks per station clearly seen corresponding to the sunset and sunrise events. The low brightness values before and after the sunrises/sunsets are due to the shorter exposure times at the very beginning and end of the imaging night.

B. Kernel Trick

The main drawback with linear SVM is that it can be applied only if the two classes are linearly separable. If the two classes are not linearly separable, we can apply SVM classification efficiently using what is called the kernel trick. The kernel trick is a way of mapping observations from a general set S into an inner product space V, without ever having to compute the mapping explicitly, in the hope that the observations will gain meaningful linear structure in V. Linear classifications in V are equivalent to generic classifications in S.

The trick to avoid having to compute the explicit mapping is to use learning algorithms that only require dot products between the vectors in V, and choose the mapping such that these high-dimensional dot products can be computed within the original space, by means of a kernel function. Some of the common kernel functions include the following:
1) polynomial—$k(x_i, x_j) = (x_i.x_j + c)^d$,
2) Gaussian radial basis function—$k(x_i, x_j) = exp$ $(-\gamma\|x_i - x_j\|^2)$, for $\gamma > 0$, and
3) hyperbolic tangent—$k(x_i, x_j) = tanh(\kappa x_i.x_j + c)$ for some $\kappa > 0$ and $c < 0$.

We use a polynomial kernel in our experiments for the sake of simplicity. All the experiments were performed using *svm-light* [21] implementation.

V. POSTPROCESSING

The auroral images are taken periodically and hence we can exploit this temporal information to improve the performance of the classification. It is highly unlikely that in a consecutive image series covering several minutes only one of the images actually contains aurora. Simple median filtering of classifications or other nonlinear approaches can be put to effective use. In a case where there is aurora on the background and fast changing cloudiness between the aurora and the observer, single images containing a little bit of aurora even if correctly classified are not

valuable in the studies of auroral dynamics. Single images containing a little bit of aurora within a cloudy or foggy time period are equally challenging to detect by a professional human observer.

Common sources for misclassification errors are sunrise and sunset that cause severe background illumination gradients. This can be easily rectified by ignoring images with higher absolute brightness values than an empirically determined threshold. Application of a brightness threshold on the mean of the color histogram values helps us to get rid of the images captured during sunrise and sunset. This is not a severe loss as the background luminosity is so high that detecting auroral activity can be challenging even for a trained eye. Fig. 6 shows the average color histogram values for data captured from the two stations over a period of two nights comprising about 10 000 images (index on the x-axis in Fig. 6). Given an image, the brightness values are calculated as

$$\mathbf{I}_{hist}(i) = \frac{\mathbf{R}_{hist}(i) + \mathbf{G}_{hist}(i) + \mathbf{B}_{hist}(i)}{3}$$
$$b = \frac{1}{N} \sum \mathbf{I}_{hist}(i)$$

where $\mathbf{I}$ is the cumulative histogram values, b is the absolute brightness level to which the threshold is applied, and $N(= 256)$ is the number of bins used to compute the histograms.

VI. EXPERIMENTS AND RESULTS

In this section, we apply the methods described earlier, to select a model which best performs the classification task. The labeled data used to build the supervised classification model consists of 32 918 images, which is a combination of randomly selected images from Muonio and Kevo over the years 2011 and 2012, and also randomly ordered images from 4 days (February 26–29, 2012). All data originate from the stations at Muonio and Kevo. There were 37% images with auroras and 63% images
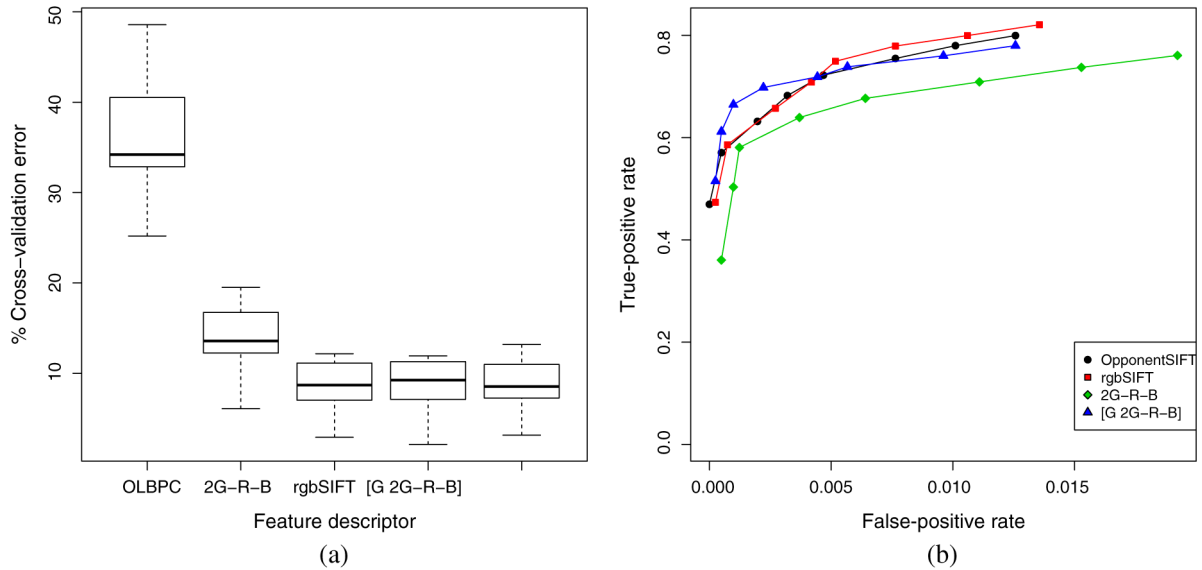
Fig. 7. (a) Cross-validation errors and (b) ROC curve for different local feature descriptors. The line inside the rectangle in the box plots represents the mean of the cross-validation errors, the height of the rectangles represent inter-quartile range, and the bars represent the maximum and minimum errors. In the ROC curve, the closer the points are to the top-left corner, the better is the performance.

TABLE I
MEAN CROSS-VALIDATION ERRORS AND MEAN TRUE-POSITIVE RATES AND FALSE-POSITIVE RATES FOR DIFFERENT LOCAL FEATURE DESCRIPTORS

	Performance analysis of different features		
	Mean cross validation error (%)	Mean true positive rate	Mean false positive rate
LBP	35.59	0.56	0.31
2G-R-B	13.67	0.84	0.095
rgbSIFT	8.67	0.89	0.054
[G 2G-R-B]	8.63	0.89	0.056
OpponentSIFT	8.51	0.90	0.054

without auroras with an approximately same contribution from both stations.

We perform SVM classification on features extracted using OLBPC and different color SIFT features namely OpponentSIFT and rgbSIFT. We also use the features extracted using selective weighting of the color channels. First, we use a weight of 2 for the green channel and subtract the red and blue channel values $(2G - R - B)$, as described in Section III-E. Another set of feature vectors are extracted, which are a combination of just the green channel and the $2G - R - B$. Thus, the resultant feature vector is $[G\ 2G - R - B]$.

Each of the extracted features—32 918 vectors, were divided into 13 parts of 1266 each (except for the last one) and perform k-fold cross-validation with $k = 13$. Performance metrics for SVM classification using the five different feature vectors are shown in Fig. 7. We performed an error analysis using k-fold cross-validation. The mean cross-validation errors for each of the features is reported in Table I.

In a multiclass classification problem, when different classes are not symmetric, i.e., one of the classes is more populous than the others, accuracy as a performance metric could be misleading. Hence, we compute true-positive rates—ratio of images with auroral activity classified correctly to the total number of images with aurora ($TPR = true\ positives/total\ positive\ examples$) and false positive rates—ratio of images without

any auroral activity being wrongly classified as having an aurora to the total number of images without any aurora ($FPR = false\ positives/total\ negative\ examples$) and plot them on an receiver operating characteristics (ROC) graph. The closer the point is to the upper left corner, the better the performance. Fig. 7(b) shows the ROC curve obtained by varying the threshold on the posterior probablities for each of the different features evaluated during our experiments. When an image with aurora is missed, it is a false negative. Our aim is to capture all auroral events, i.e., to reduce false negatives, even if it increases the number of false positives. Hence, we choose a feature extraction technique with highest true-positive rates.

OpponentSIFT, rgbSIFT, and the latter of the weighted feature vectors [G 2G-R-B] have performances that are comparable, while OLBPC performs the worst of the extracted features. As is evident from Fig. 7, OpponentSIFT, rgbSIFT, and [G 2G-R-B] have accuracies $(100 - classification\ error)$ of 91%, while LBP performs the worst with an accuracy of 64%. The ROC curve for OLBPC has been omitted from Fig. 7(b) in order to represent the other methods more clearly.

Weighting the green channel intuitively makes sense, although experimentally, we found out that it performs worse than OpponentSIFT features. Features extracted with the green channel having twice the weight as the other two channels and the other two channels used as counter evidence has a classification
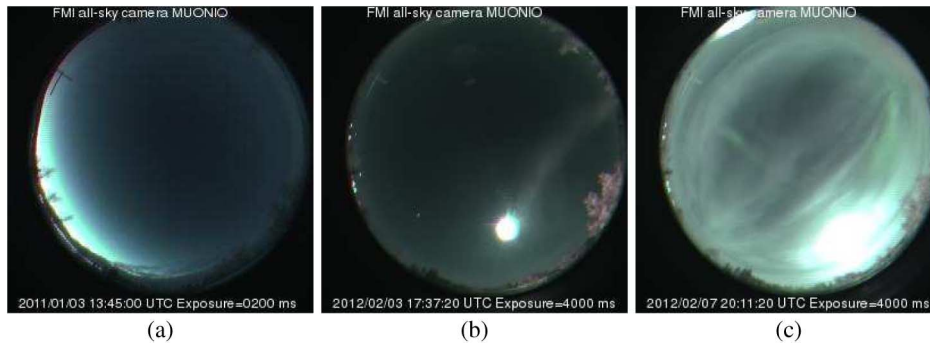
Fig. 8. Example of images that were wrongly classified. In (a), the sunrise event throws out a green arc on the edge of the image that gets mistaken for an aurora. (b) shows a moonlit smoke trail which was confused to aurora by the method. (c) is an example of thin layer of high clouds illuminated by the full moon with some faint aurora behind which was missed by the method.

accuracy of 86%, while a combination of this feature with just the green channel performs almost equally as well as OpponentSIFT. However, this weighted combination of green and other colors includes more descriptors than OpponentSIFT and is thus computationally more expensive.

After the classification, we can perform postprocessing described earlier to improve the classification result. First, we use a threshold on histogram values to filter images captured during sunrises and sunsets, which have a high absolute illumination. Any analysis on these images is difficult and hence are considered as outliers. A threshold of 23.5 was empirically found to filter out most of the images captured during sunrises and sunsets.

As shown in Fig. 6, three of the four bright periods at the ends of imaging exceed the threshold and end-up discarded. The Kevo (green) sunset (the first peak) with lower brightness is due to partial cloud cover and will be automatically correctly classified as *No Aurora*. The most challenging conditions are clear sky in dawn/dusk, where auroral light looks faint and colorless.

The next step is to use the temporal information to improve the classification performance. We used the time information in the images to determine the time interval between change in classification labels. If the interval is smaller than the threshold interval, we ignore the change and assign the same label as the majority of the images in the sequence. This filters out single images being mislabeled during an auroral event. A threshold of 180 s was empirically determined to perform satisfactorily.

The main objective of this work was to detect the auroral events of all image data. Although the classification errors reported in Table I are for the three-class classification problem, for all further analyses, class *cloudy* was included with the class *no aurora*. The class *cloudy* was found highly dependent on the light conditions at each station and least accurately detected in the manually labeled data. The information about this class has been saved for future studies.

All wrongly classified data have been visually examined. They primarily include sunrise/sunset images where the bright horizon has been interpreted as aurora (a reason why the brightness threshold was used to cut out the brightest frames), full moon behind thin high clouds and thin aurora where the thin colorless auroral arc is being confused to clouds, and some cases with artificial light which may incorrectly seem like aurora. These cases have also been very difficult to correctly classify for the professional eye. Fig. 8 shows some of the misclassified images.

VII. CONCLUSION

We performed experiments with different types of local feature descriptors to perform classification of color auroral all-sky images. Of the different features, OpponentSIFT features were found to be the most effective out of all the feature descriptors used. Since these features are independent of absolute brightness, we were able to use the same model to classify images captured at different camera stations. We also observed this method to work reasonably well with different exposure values. However, the exposure values remained same for a majority of the images. We were able to perform the classification with a cross-validation accuracy of 91% and a test accuracy of around 80%. These values are further improved by postprocessing to result in an accuracy higher than 90%, which was earlier concluded to be sufficient for operational purposes [15].

As a part of the process, we have been able to create a comprehensive data set of labeled ASC images that can be used in future experiments. After running the automated classification of auroral images for all the exisiting image data we will eventually be able to build statistical models which will aid in the prediction of auroral occurrence.

The method has potential for future use for data from different stations, but its performance can only be assessed by providing a set of labeled data from these new stations. Even though the training of the classifier is computationally expensive, a trained model can be run at a modern station computer during the daytime after the nights imaging.

We have shared animations generated from images that were classified to contain auroral events and images that were classified into class *no aurora* and *cloudy* at http://www.space.fmi.fi/MIRACLE/ASC/colour_classification.html and http://www.space.fmi.fi/MIRACLE/ASC/colour_classification.html as an example output of the classification process.

References

[1] M. T. Syrjasuo *et al.*, "Low-cost multi-band ground-based imaging of the aurora," in *Proc. Int. Soc. Opt. Photonics*, 2005, pp. 59010F–59010F.

[2] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[3] C. Zhu, C.-E. Bichot, and L. Chen, "*Colour orthogonal local binary patterns combination for image description*," Ecole Centrale de Lyon, Écully, France, Tech. Rep. LIRIS UMR5205 CNRS, 2011.

[4] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[5] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. ACM Int. Conf. Image Video Retr.*, 2007, pp. 401–408.

[6] P. Morguet and M. Lang, "Feature extraction methods for consistent spatio-temporal image sequence classification using hidden markov model," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 1997, vol. 4, pp. 2893–2896.

[7] F. Melgani and S. B. Serpico, "A Markov random field approach to spatio-temporal contextual image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 11, pp. 2478–2487, Nov. 2003.

[8] Y. Liu, D. Zhang, G. Lu, and W. Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognit.*, vol. 40, no. 1, pp. 262–282, 2007.

[9] M. T. Syrjäsuo and E. F. Donovan, "Using relevance feedback in retrieving auroral images," in *Proc. 4th IASTED Int. Conf. Comput. Intell.*, Jul. 2005, pp. 273–297.

[10] F. Tang, S. H. Lim, N. L. Chang, and H. Tao, "A novel feature descriptor invariant to complex brightness changes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 2631–2638.

[11] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 604–610.

[12] Finnish Meteorological Institute. (2013, Dec. 17). "Creating Keograms". Available: http://www.space.fmi.fi/MIRACLE/ASC/asc_keo_00.shtml

[13] M. T. Syrjäsuo *et al.*, "Observations of substorm electrodynamics using the MIRACLE network," in *Substorms-4*, S. Kokubun, Y. Kamide, Eds. Tokyo, Japan: Terra Scientific, 1998, pp. 111–114.

[14] M. Syrjäsuo and E. Donovan, "Analysis of auroral images: Detection and tracking," *Geophysica*, vol. 38, pp. 3–14, 2002.

[15] M. Syrjäsuo and N. Partamies, "Numeric image features for detection of aurora," *Geosci. Remote Sens. Lett.*, vol. 9, pp. 176–179, 2012.

[16] Q. Wang *et al.*, "Spatial texture based automatic classification of dayside aurora in all-sky image," *J. Atmos. Solar-Terr. Phys.*, vol. 72, pp. 498–508, 2010.

[17] T. Ojala *et al.*, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[18] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 420–425, 1995.

[20] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retr.*, vol. 4, no. 2, pp. 215–322, 2009.

[21] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. Burges and A. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999.

[22] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[23] Finnish Meteorological Institute. (2014, Mar. 4). "FMI All-Sky Cameras - Imaging the Auroras". Available: http://www.space.fmi.fi/MIRACLE/ASC/index.html

[24] L. Sangalli *et al.*, "Performance study of the new EMCCD-based all-sky cameras for auroral imaging," *Int. J. Remote Sens.*, vol. 32, p. 11, 2011.

[25] N. Partamies, M Syrjäsuo, and E. Donovan, "Using colour in auroral imaging," *Can. J. Phys.*, vol. 85, no. 2, pp. 101–109, 2007.

Jayasimha Rao received the B.E. degree in computer science and engineering from Visvesvaraya Technological University, Belgaum, Karnataka, India, in 2007. Currently, he is pursuing the M.Sc.(tech.) degree at School of Science, Aalto University, Helsinki, Finland.

His research interests include augmented reality, content-based image retrieval, image completion, multispectral imaging, and information visualization.

Noora Partamies received the Ph.D. degree in physics from the University of Helsinki, Helsinki, Finland, in 2004.

Currently, she is with the PI with the MIRACLE auroral all-sky camera network at the Finnish Meteorological Institute, Helsinki, Finland. Her research interests include auroral imaging and ionospheric physics to magnetosphere–ionosphere coupling and space weather.

Olga Amariutei received the B.Eng. degree in automatic, control, and industrial system engineering from the Gheorghe Asachi University, Iași, Romania, in 2007, after conducting her final diploma project at University of Sheffield, South Yorkshire, U.K. In 2007, she joined the Automatic Control and System Engineering at the University of Sheffield to pursue the Ph.D. degree in plasma physics.

In 2011, she started her Post-Doctoral Researcher appointment at the Finnish Meteorological Institute, Helsinki, Finland, in space weather. Her research interests include nonlinear modeling, control, prediction, and forecasting applied to plasma physics and image processing.

Mikko Syrjäsuo received the doctoral degree of science in technology from the Helsinki University of Technology, Helsinki, Finland, in 2001.

Currently, he is working as a Technology Manager with the School of Electrical Engineering, Aalto University, Helsinki, Finland. His interests include automated image analysis and spaceflight instrumentation.

Koen E. A. van de Sande received the B.Sc. degrees in computer science and artificial intelligence, both in 2004; the M.Sc. degree in computer science, in 2007; and the Ph.D. degree in computer science, in 2011, all from the University of Amsterdam, Amsterdam, The Netherlands.

Currently, he is a Guest Researcher with the University of Amsterdam and works in R&D at Euvision Technologies, Amsterdam, The Netherlands. His research interests include computer vision, object recognition and localization, machine learning, parallel computing, and large-scale benchmark evaluations (PASCAL VOC, TRECVID, and ImageNet).