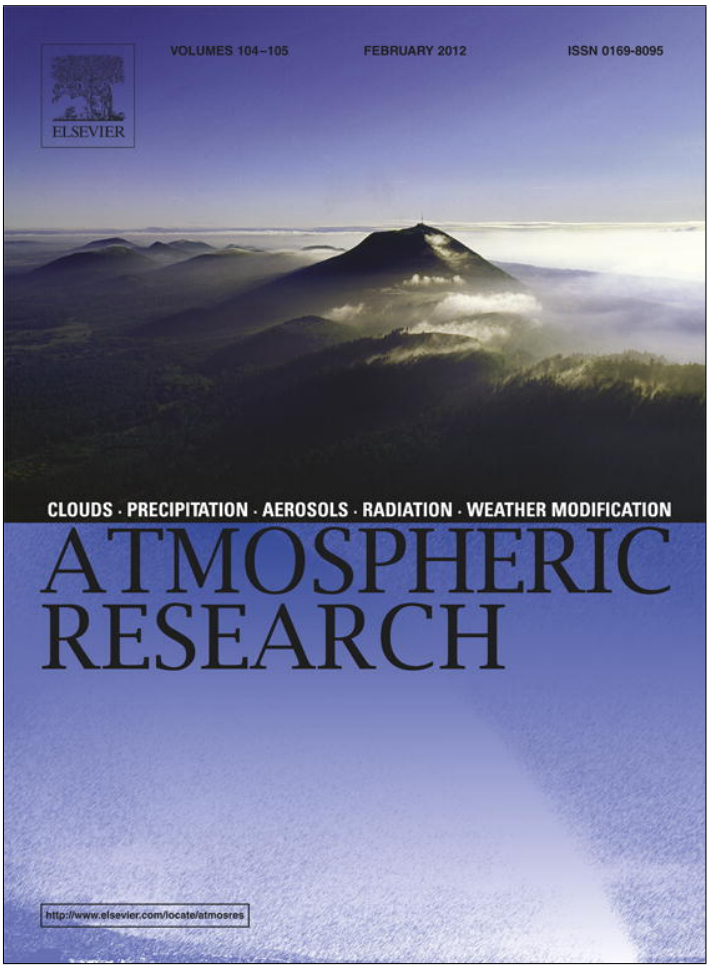

(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Atmospheric Research 118 (2012) 121–132

Contents lists available at SciVerse ScienceDirect

Atmospheric Research

journal homepage: www.elsevier.com/locate/atmos

Texture operator for snow particle classification into snowflake and graupel

Karolina Nurzyńska [a,*], Mamoru Kubo [a], Ken-ichiro Muramoto [a,b]

[a] School of Electrical and Computer Engineering, Kanazawa University, Kanazawa, Japan
[b] Ishikawa National College of Technology, Tsubata, Japan

ARTICLE INFO

Article history:
Received 4 August 2011
Received in revised form 7 June 2012
Accepted 7 June 2012
Available online 16 June 2012

Keywords:
Image processing
Classification
Texture operators
Meteorological radar

ABSTRACT

In order to improve the estimation of precipitation, the coefficients of Z–R relation should be determined for each snow type. Therefore, it is necessary to identify the type of falling snow. Consequently, this research addresses a problem of snow particle classification into snowflake and graupel in an automatic manner (as these types are the most common in the study region). Having correctly classified precipitation events, it is believed that it will be possible to estimate the related parameters accurately.

The automatic classification system presented here describes the images with texture operators. Some of them are well-known from the literature: first order features, co-occurrence matrix, grey-tone difference matrix, run length matrix, and local binary pattern, but also a novel approach to design simple local statistic operators is introduced. In this work the following texture operators are defined: mean histogram, min–max histogram, and mean–variance histogram. Moreover, building a feature vector, which is based on the structure created in many from mentioned algorithms is also suggested.

For classification, the k-nearest neighbourhood classifier was applied. The results showed that it is possible to achieve correct classification accuracy above 80% by most of the techniques. The best result of 86.06%, was achieved for operator built from a structure achieved in the middle stage of the co-occurrence matrix calculation. Next, it was noticed that describing an image with two texture operators does not improve the classification results considerably. In the best case the correct classification efficiency was 87.89% for a pair of texture operators created from local binary pattern and structure build in a middle stage of grey-tone difference matrix calculation. This also suggests that the information gathered by each texture operator is redundant. Therefore, the principal component analysis was applied in order to remove the unnecessary information and additionally reduce the length of the feature vectors. The improvement of the correct classification efficiency for up to 100% is possible for methods: min–max histogram, texture operator built from structure achieved in a middle stage of co-occurrence matrix calculation, texture operator built from a structure achieved in a middle stage of grey-tone difference matrix creation, and texture operator based on a histogram, when the feature vector stores 99% of initial information.

1. Introduction

Hokkuriku District on Honshu Island, Japan, is affected by many monsoon winds coming from the Northeast Asian continent. These winds move towards the land clouds, which form over the Sea of Japan. Due to the high mountains in the region, with elevation of 3000 m, the clouds change its stage from developing to mature one. This combination of factors is responsible for very high snowfall rate in this area (up to 2 m per day). In consequence, there are many events of floods, landslides, traffic problems, etc.

In order to prevent and diminish the influence of the weather on life, computer systems are developed as an aid in accurate weather forecasting. The base of these systems is weather prediction algorithms from the data gathered by weather radar. These data estimate the precipitation rate (R) on the basis of the radar reflectivity factor (Z). The relation

* Corresponding author.
E-mail address: Karolina.Olga.Nurzynska@gmail.com (K. Nurzyńska).

0169-8095/$ – see front matter © 2012 Published by Elsevier B.V.
doi:10.1016/j.atmosres.2012.06.013

between these two parameters is given by (Marshall and Gunn, 1952) by the formula:

$$Z = BR^{\beta}, \tag{1}$$

where B and β are coefficients which considerably depend on the precipitation type. It was also noticed that these parameters are strongly related to the falling particle size (Loffler-Man and Blahak, 2001; Matrosov, 1992).

It is also known, from the work of (Mizuno, 1992), that the predominant type of snow particle precipitation, on the coast of the Sea of Japan, is graupel. (Matsuo et al., 1994) state that its formation is influenced by the updraft. Additionally, (Harimaya et al., 2004) noticed, that also the snowflake size distribution varies within this region. In conclusion they state that the snowflake size and density are related to the distance from the coast. It reflects the processes responsible for snowflake creation. As a consequence, this paper addresses the problem of snow particle classification into snowflake and graupel. The classification of the precipitation events would improve the accuracy of precipitation estimation, because it would be possible to find separate Z–R relation coefficients for each snowfall type.

This paper presents part of research concerning application of image processing techniques, which can be used to describe snow particle images. In previous works the statistical and shape features have been exploited to describe the images. Firstly, many existing statistical techniques with shape features were tested. The results are presented in (Nurzyńska et al., 2010b). Then, in order to improve the object description some novel shape features have been developed. Their performance is described in the paper (Nurzyńska et al., 2011a). Finally, the possibility to apply texture operators has been researched. The initial results were presented in (Nurzyńska et al., 2011b). In this work the full range of conveyed experiments with image processing technique basing on texture operators is presented.

The development of technology allows using a digital camera to obtain snow particle images with satisfactory resolution. Yet, applying image data for precipitation research is quite an old idea. For instance, at the University of Maintz the ground-based holographic droplet and aerosol recording system was developed in late 80s (Vössing et al., 1998). The system gathered the information about aerosol particles and droplets from a volume of approximately $1l$ on holographic images. It permitted measuring the raindrop and hydrometeor size distribution as well as its shapes. On the other hand, in order to find characteristic properties of precipitation (Frank et al., 1994) described the pluviospectrometer, which is an optical device also utilizing image processing. Recently, a new equipment is exploited for precipitation recognition like Pludix, X-band disdrometer and other, which performance have been compared by (Prodi et al., 2011).

The structure of this paper is the following. Firstly, the imaging system used for data acquisition is presented in Section 2. This section also contains a description of image pre-processing algorithms applied for image database creation. Next, Section 3, describes texture operators well-known from the literature. Furthermore, new approaches are suggested for texture operator definition. Section 4 presents the chosen classification algorithm. The results with discussion are given in Section 5. Finally, Section 6 draws the conclusions.

2. Imaging system

2.1. System overview

A laboratory dedicated for weather precipitation measurements was constructed at Kanazawa University. All equipment used for this research is placed in a small area on the laboratory roof. This permits the assumption, that all measurements describe the same precipitation event.

In order to conduct the research, described in this paper, only an imaging system was applied. The imaging system was designed to assure the best conditions for snow particle image acquisition. Firstly, the whole construction is enclosed by a wind breaker. Next, the camera is placed in a horizontal tunnel 2 m from the measurement area. The measurement space itself is placed inside a hollow tube oriented vertically. The aim of the barriers is to diminish the influence of wind on the snow particle speed and trajectory. Additionally, there are four halogen lamps, which illuminate the measuring area and assure similar conditions regardless of daytime and sunshine illumination.

The images recorded by the camera have resolution of 1280×960 pixels. The shutter speed is $1/10{,}000s$. The data gathered by the camera is transmitted to an external PC by an IEEE 1394b cable. The volume of the measuring space is $W128 \times H96 \times D250$ mm.

Dedicated image pre-processing software was developed for this system. The program analyses each grabbed frame and crops snow particle images. Firstly the binary representation of an image is calculated which permits distinguishing snow particles from the background. Next, to separate multiple particles, which were recorded on one image, each binary object is given a label. Each labelled object then is cropped from the image with some area around it (usually 30 pixels in each direction) and stored in the snow particle image database. Some examples of recorded snow particles are presented in Fig. 1.

2.2. Database description

From the images gathered automatically, by the system described in the previous section, a smaller set was selected to create a snow particle database. The only criteria applied when choosing the snow particle image was a visual inspection. A specialist looking through collected images was deciding whether a particle on the image reflects snowflake or graupel and set a corresponding label. In case when it was difficult to make a decision or when the image was of poor quality, it was deleted. However, when the quality was good, the image was allocated to one of the classes.

The snow particle database, used in experiments, contains 4384 snowflake and 4096 graupel grey-scale images. The training dataset consists of 460 images of graupel and 461 of snowflake. This gives in total 7559 images in the testing database. The database is not normalized neither in terms of resolution of images nor in terms of rotation of objects.

The size of each snow particle is calculated as a sum of its pixels, whereas the pixel dimensions are equal to 0.1×0.1 mm^2.

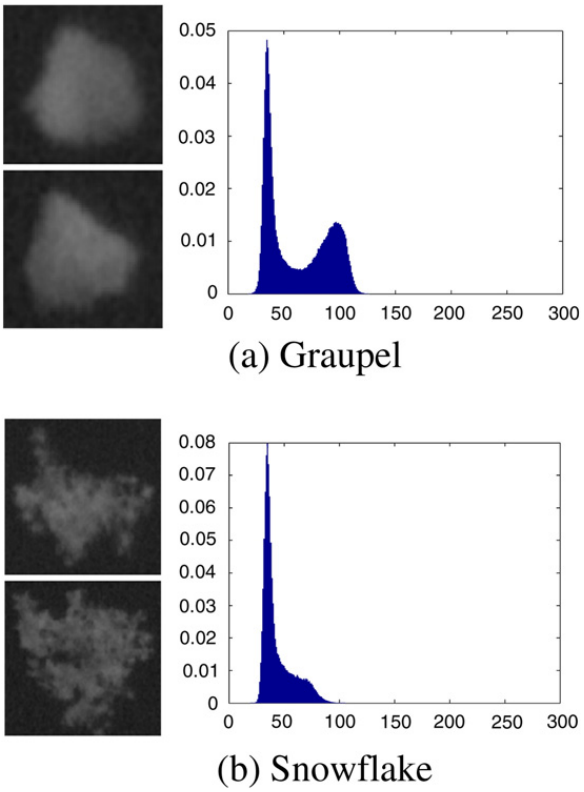

Fig. 1. Recorded snow particles with average class histograms of image intensity.

Fig. 2 presents the size distribution of snowflakes and graupels gathered in whole dataset. Some examples of images with average class histogram are presented in Fig. 1.

3. Texture description

Computer programs, designed for image processing purposes, aim to understand the image in such a manner as human brain does. Therefore, many techniques have been suggested which try to mimic brain behaviour during the recognition process. These methods take under consideration distinctive features of objects like shape (Nurzyńska et al., 2010b, 2011a), texture (Nurzyńska et al., 2010a, 2011b), or both.

This research addresses a comparison of well-known techniques for object classification based on image texture. In other words, for an image a texture operator is calculated, which describes the image content by means of few or even hundreds of values. These values form a feature vector used for classification. Below well-known techniques are presented designed to describe the texture as well as some novel approaches are suggested.

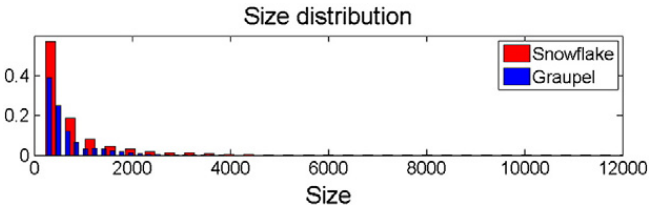

Fig. 2. The size distribution of snowflake and gruapel particles in snow particle image database.

3.1. First order features

The first order features (*FOF*) are based on an image intensity distribution over an image histogram. It describes the general quality of the image. Let $I(x, y)$ represent an image function of two variables x and y, and $x = 0 \ldots M-1$ and $y = 0 \ldots N-1$, where M and N are image resolution. This function takes the values $i = 0 \ldots G-1$, where G represents the total number of intensity levels of an image. In this research grey-scale images are concerned, therefore G equals to 256. The histogram (distribution) of intensity level occurrence is given by the formula

$$H(i) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \begin{cases} 1, & I(x,y) = i \\ 0, & I(x,y) \neq i \end{cases}. \quad (2)$$

Usually, in order to compare histograms of images of different resolutions, they are normalized

$$h(i) = \frac{H(i)}{NM}. \quad (3)$$

Besides, it is worth noticing that $h(i)$ represents also the intensity density distribution function, which describes the probability of each intensity level to occur in the image. For example, an average intensity distribution for snowflake and graupel classes is presented in Fig. 1. It is interesting to notice, that although the objects on the images seem similarly grey, its intensity distribution histograms are different. Therefore, using only a histogram as a texture operator should allow for particle classification.

On the basis of histogram data (Materka and Strzelecki, 1998) compiled formulae for parameters describing the following values: mean, variance, skewness, kurtosis, energy, and entropy, which describe the texture quality in this method. An example of a histogram calculated for an image is depicted in Fig. 3.

3.2. Co-occurrence matrix

The second order features additionally consider the spatial relations between the luminance intensities within an image. As a consequence, it contains information about the spatial properties. This is important in image classification, because usually a value of a single pixel is correlated with values of neighbours. Otherwise, it could be suspected to be a noise.

(Haralick et al., 1973) introduced the spatial relationship between pixel intensities in the definition of spatial dependence matrix, which later became called a co-occurrence matrix by (Zucker and Terzopoulus, 1980); this name is now in common use. Generally speaking, the co-occurrence matrix (*COM*) stores the information about the illumination values co-appearance in the image. For given G luminance levels *COM* has a resolution $G \times G$. Each cell, $com(x, y)$, of *COM* contains the information of the co-occurrence of intensity levels of value x and y in the given direction θ, at the distance d. Fig. 4 presents the neighbourhood defined for each pixel in the image. Table 1 presents four basic directions for which this matrix is calculated. It is necessary for further calculation to normalize the frequency data, R, in this matrix. The normalization factor for each direction is given in the last column of the

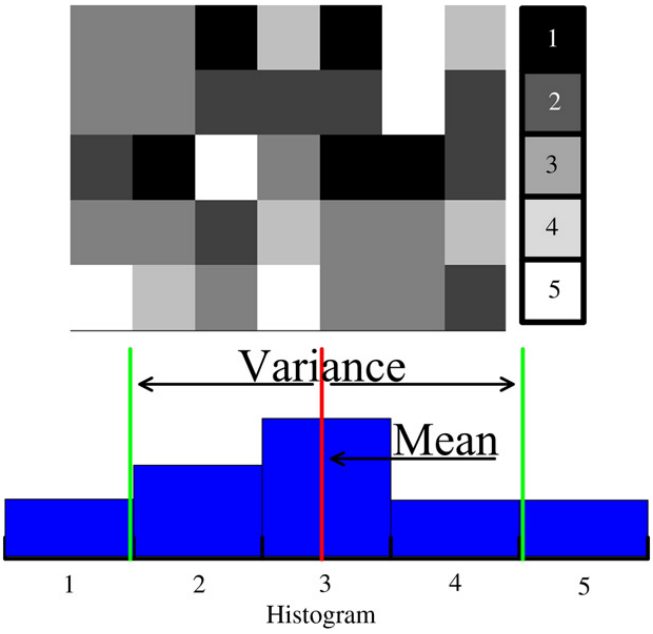

Fig. 3. Histogram calculated for an exemplary image with visualization of two parameters (mean and variance) from the first order features.

Table 1. Fig. 5 presents steps of this algorithm applied to an exemplary image of five intensity levels.

Four co-occurrence matrices are calculated for one image, they correspond to each direction. Then for all of them 14 parameters are calculated, which describe among others: texture contrast, correlation, entropy, and information measure. For the definition of the parameters please refer to (Haralick et al., 1973). The final feature vector contains the average and standard deviation calculated for each feature over these four matrices.

The most problematic issue, when applying the co-occurrence matrix, is its calculation complexity, which depends on the co-occurrence matrix size. Therefore, it is suggested to limit all possible luminance level values to less than original 256, in order to achieve better calculation speed. In this research the image illumination was quantized non-linearly into 16 shades. One shade represented the background and other 15 represented linear scaling for shades encountered in the object image.

3.3. Grey tone difference matrix

Grey-Tone Difference Matrix (*GTDM*) presented by (Amadasun and King, 1998) is an attempt to define texture measures correlated with human perception of textures.

6	7	8
5	o	1
4	3	2

Fig. 4. The neighbourhood of *o* pixel in a co-occurrence matrix.

This matrix is a column vector containing G elements. The entries represent the difference between the intensity levels of a pixel and an average intensity computed over a square window centred at the pixel and sliding over the image. The average intensity for the window centred at pixel (x, y) is

$$\bar{I}_i = \frac{1}{W-1}\sum_{m=-K}^{K}\sum_{n=-K}^{K} I(x+m, y+n), (m,n)\neq(0,0), \quad (4)$$

where K specifies the window size and $W=(2K+1)^2$. Next, the i-th entry of the grey-tone difference matrix is

$$s(i) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |i-\bar{I}_i|, \quad (5)$$

for all pixels having the intensity level i, otherwise $s(i)=0$. Fig. 6 presents a *GTDM* structure contents for the exemplary image.

From this definition five features were developed, which describe the texture coarseness, contrast, busyness, complexity, and strength.

3.4. Local binary pattern

The idea of Local Binary Pattern (*LBP*) presented in works of (Ojala et al., 2000) and (Ojala et al., 2002) is very simple,

Table 1
The direction definition and the normalization factor for co-occurrence matrices based on notation from Fig. 4.

θ	Neighbours	Normalization factor R
0-degree	5 – o – 1	$2I_y(I_x-d)$
45-degree	4 – o – 8	$2(I_x-d)(I_y-d)$
90-degree	3 – o – 7	$2I_x(I_y-d)$
135-degree	6 – o – 2	$2(I_y-d)(I_x-d)$

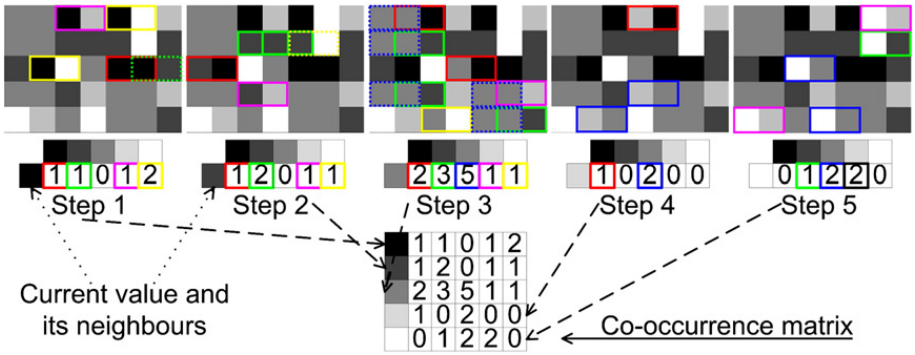

Fig. 5. Co-occurrence matrix achieved for image with 5 intensity levels. Steps 1 to 5 present the algorithm for co-occurrence matrix calculation for distance 1 pixel and angle 0°.

yet powerful. Authors understand the texture as a two-dimensional phenomenon characterized by two orthogonal properties: spatial structure (pattern) and contrast (the "amount" of local image texture). The definition starts from the joint distribution of luminance value on a circularly symmetric neighbour set of pixels in a local neighbourhood. Then an operator invariant against any monotonic transformation of luminance is derived. Rotation invariance is achieved by recognizing that this illumination invariant operator incorporates a fixed set of rotation invariant patterns.

In order to find the illumination rotation invariant texture operator, the texture T, in local neighbourhood of monochrome image, is seen as the joint distribution of the grey levels of $P(P>1)$ image pixels, and is given as

$$T = t(g_c, g_0, \ldots, g_{P-1}), \tag{6}$$

where g_c corresponds to the luminance value of the central pixel of the local neighbourhood and $g_p(p=0, \ldots, P-1)$ describes the luminance values of P equally spaced pixels on a circle of radius $R(R>0)$, that form a circularly symmetrical neighbourhood.

The coordinates for point g_p are given by $\left(-Rsin\frac{2\Pi p}{P}, Rcos\frac{2\Pi p}{P}\right)$ and for $g_c=(0, 0)$. Next, to achieve the illumination invariance, the luminance value of central pixel g_c is subtracted

$$T = t(g_c, g_0 - g_c, \ldots, g_{P-1} - g_c). \tag{7}$$

Authors assume that the difference $g_p - g_c$ is independent of g_c, which allows for factorization

$$T \approx t(g_c)t(g_0 - g_c, \ldots, g_{P-1} - g_c). \tag{8}$$

Finally, they assume that the information conveyed in the value of the central point, g_c, contains only luminance information, whereas whole texture is described by the other factors, therefore they remove it from the definition without large loss of information for further calculation

$$T \approx t(g_0 - g_c, \ldots, g_{P-1} - g_c). \tag{9}$$

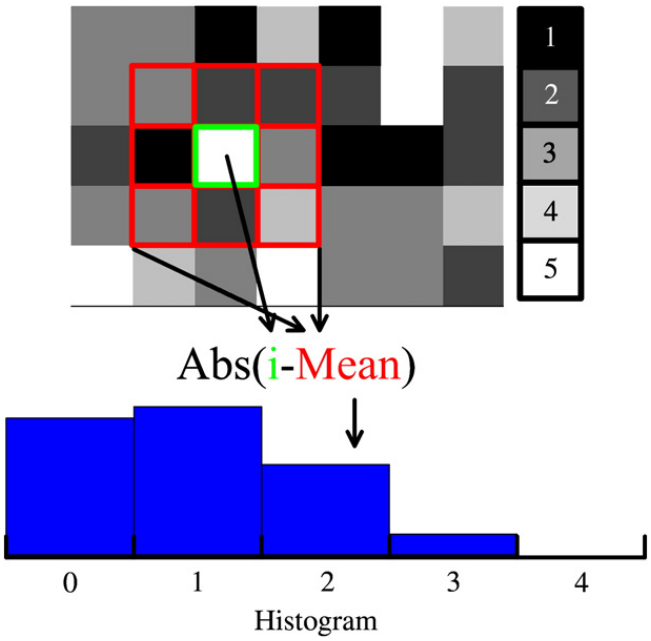

Fig. 6. Grey tone difference matrix calculated for image with 5 illuminance levels.

This texture operator should be very discriminative, yet some further processing is necessary. Firstly, it is binarized according to the formula

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{10}$$

and then the value is calculated as

$$LBP_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s\left(g_p - g_c\right) & if\ U\left(LBP_{P,R}\right) \leq 2 \\ P+1 & \text{otherwise,} \end{cases} \tag{11}$$

where

$$U\left(LBP_{P,R}\right) = \sum_{p=0}^{P-1} |s\left(g_p - g_c\right) - s\left(g_{p-1} - g_c\right)|. \tag{12}$$

Additionally, to incorporate the information of the illumination in the image the authors suggest the new definition of the variance operator as

$$VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{P-1} \left(g_p - \mu\right)^2, \quad \text{where } \mu = \frac{1}{P}\sum_{p=0}^{P-1} g_p. \tag{13}$$

Finally, they calculate the histogram over the image from the joint distribution of $LBP_{P,R}/VAR_{P,R}$. They also notice that the coefficients P and R must not be similar in case of calculation for both parameters. Fig. 7 presents the idea of *LBP* calculated for an exemplary image with $R=2$ and $P=12$.

3.5. Run length matrix

Run length matrix (*RLM*) was proposed by (Galloway, 1975) for texture classification. It is based on the assumption that the texture of good quality is characterized by small number of consecutive pixels in a given direction with similar luminance (short runs). On the contrary, the coarse textures are characterized with longer runs.

For an image the run length matrix cell, $r(i,j)$, is defined as the number of runs of pixel of luminance i and length j. Where the maximal pixel value is G and the maximal run depends on the image resolution $K = max(N,M)$. Fig. 8 presents steps of the algorithm on an exemplary image. In further research it was noticed that such definition makes the matrix dependable on the object rotation, therefore matrices for each direction are calculated; similarly as in case of co-occurrence matrix method.

In the next step the average matrix is calculated from which several parameters are derived according to formulae presented by (Galloway, 1975) and reported also by (Tang, 1998), (Albregtsen et al., 2000). Moreover, additional features were developed by (Chu et al., 1990).

3.6. Novel approach

This section presents novel approaches to texture operator calculation. Some of the techniques are popular in image processing, although its application for texture descriptors has not been known to the authors. From one point of view, it is suggested to design simple local texture statistics, which in general, recollects the local binary pattern idea, however its definition as well as the calculation is less complicated. On the other hand, using structures which are designed for feature calculation as a feature vector is also suggested.

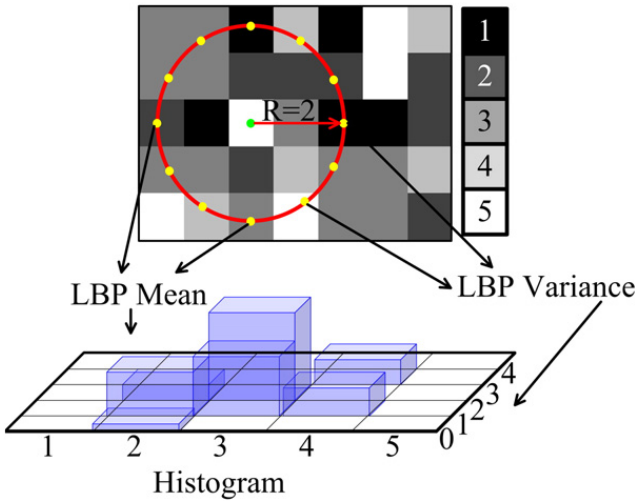

Fig. 7. Local binary pattern matrix achieved for image with 5 luminance levels. LBP is calculated for radius equal to 2 pixels and for 12 points.

3.6.1. Local texture statistic operators

This section presents simple texture operators based on local texture statistics. All represent joint distribution of local statistical features calculated for an image. The locality is defined by a sliding window with side equal to 7 pixels. Application of square neighbourhood allows also to assume the rotation invariance of the achieved operators. Following operators are introduced:

- mean histogram (*MH*) — for each pixel the mean value in the window is calculated and the results are presented as a histogram of 64 bins. Fig. 9(a) shows an idea of this texture operator;
- min–max histogram (*MMH*) — for each pixel the minimal and maximal value in the window are found and its absolute difference is the value stored in the histogram of 64 bins (see Fig. 9(b));
- mean–variance histogram (*MVH*) — for each pixel the mean and variance in the window are calculated and the results index two-dimensional histogram (16×16 bins) (see Fig. 9(c)).

3.6.2. Structure based texture operators

Most of the described texture operators, e.g. *FOF*, *COM*, *RLM*, and *GTDM*, in order to calculate a descriptive feature, create firstly a special data structure, which is used for parameter calculation. Here the idea of (Albregtsen et al., 2000) is followed, where the data structure is utilized as a feature vector and gives better results than the features itself. It is proposed to create a feature vectors which are based on the data structures corresponding to the other mentioned techniques. In consequence, additional texture operators are defined as feature vectors built from data structures of:

- normalized histogram (*FOF*) of 64 bins,
- co-occurrence matrix (*COM*) calculated for images with quantized luminance into 16 values,

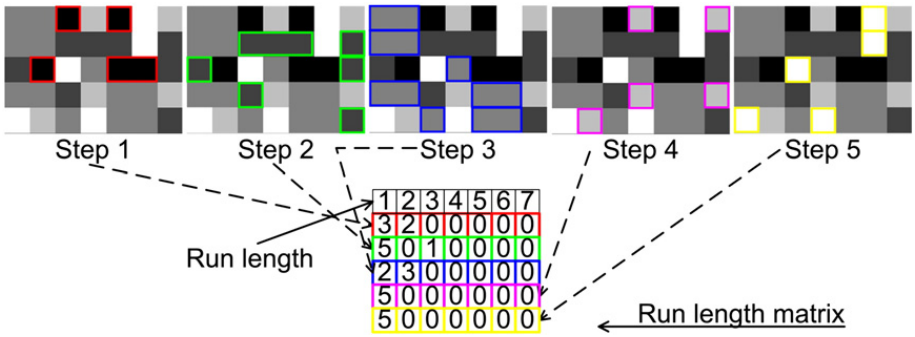

Fig. 8. Run length matrix calculated for image, with resolution 5×7 pixels and 5 grey-scale levels. Steps 1 to 5 present the calculation of run lengths for each grey-scale level.

- grey-tone difference matrix (*GTDM*) achieved for original images of 256 grey shades.

In order to distinguish between the standard method using feature vector based on features (*FVBF*) and the suggested approach with feature vector based on data structure (*FVBDS*) a letter is inserted in a subscript — it is *F*, for all based on features, and *M*, for data structures. Fig. 10 depicts the difference in data processing of these two approaches for feature vector design.

4. Classification technique

The k-nearest neighbour (*kNN*) classifier does not assume any data distribution pattern. It is an example of a supervised classification method. It means that a labelled training dataset is necessary. The preparation of the labelled dataset bases on the visual inspection of chosen images.

Each object, from testing dataset, represents a point in *P*-dimensional space (where *P* corresponds to the feature

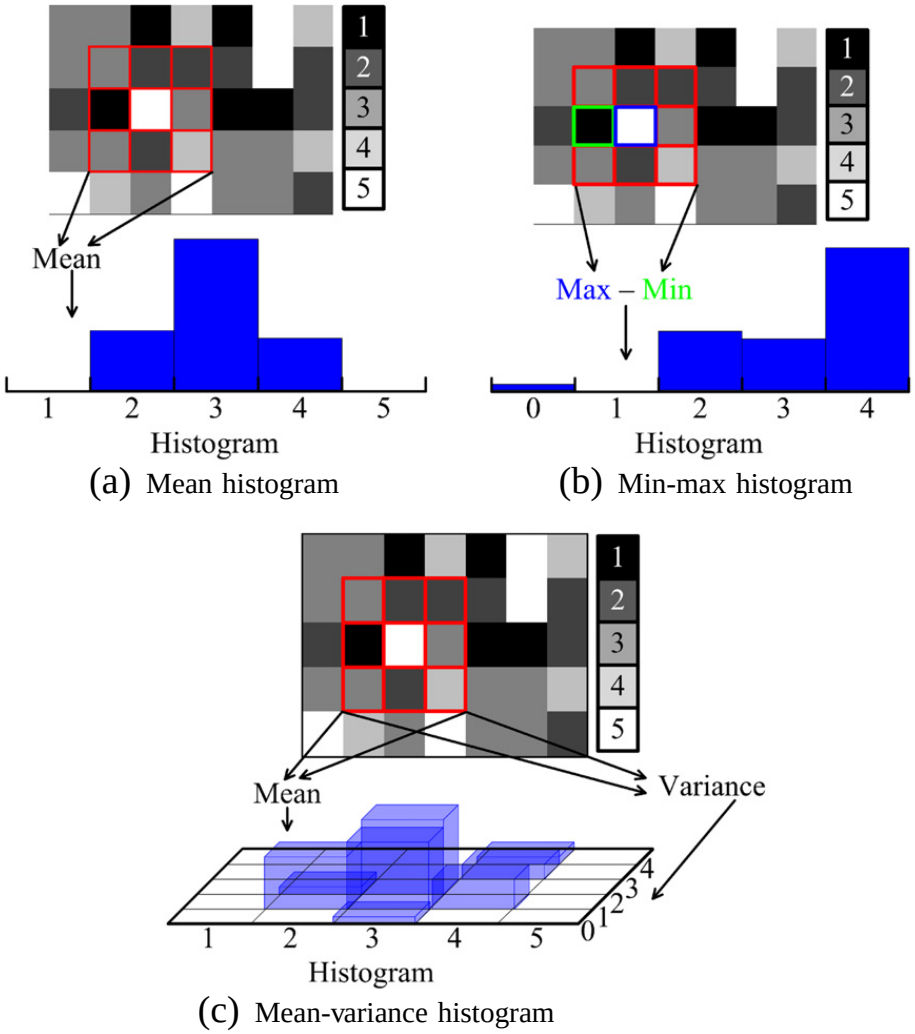

Fig. 9. Histograms calculated for local texture statistic operators.

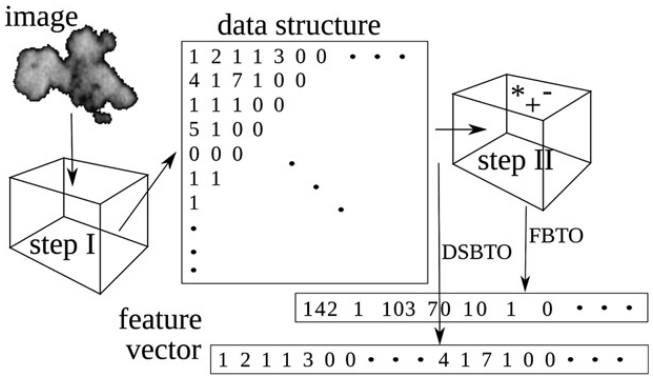

Fig. 10. Two approaches of data processing for feature vector design.

vector length). A virtual hyper-sphere (e.g. for $P=2$ features it is a circle, for $P=3$ - a sphere) is created which encloses k training objects only. The tested object is classified to this class, which is represented by the majority of training objects enclosed in the hyper-sphere. Therefore, it is important to choose the k parameter to be an odd number in case of two-class problem, otherwise there might be the same number of neighbours belonging to both classes, what makes the classification impossible. Example of *kNN* classification (for $k=3$) in two-dimensional case is presented in Fig. 11.

In the case of presented experiments the classifier was tuned up to assure that k parameter is set to an optimal value. Therefore, firstly classification tests for all techniques were conveyed for k in range 15–95, where the values changed with step equal to 10. After comparing the results, it occurred that for $k=25$ most methods achieved the best efficiency and this value was used in experiments. It is not surprising that smaller value of the parameter gave the best results. Generally, the bigger the k parameter becomes the higher is the probability that the element will be miss-classified. It is a consequence of considering only a local area, in the case of small k value, and interpret data globally, when k value is getting bigger.

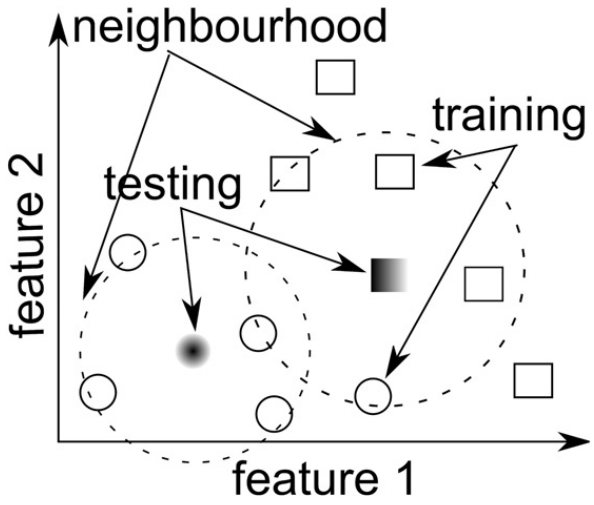

Fig. 11. K-nearest neighbour classification.

5. Results and discussion

This section presents all experiments performed in order to evaluate properties of mentioned before texture operators. Considering classification of snow particles into snowflake and graupel, it is possible to apply the k-nearest neighbour classifier. The goal is to achieve the highest classification efficiency for images gathered in the snow particle image database, which represents the data collected by the image acquisition system.

The creation of labelled dataset of snowflake and graupel demanded a co-operation with a person, who could appoint each image to one of the classes. The visual inspection assumed that only images with low quality could be removed from the database. Otherwise, the experience of the specialist was the only criteria used for image classification.

5.1. Texture operator comparison

This experiment aimed to evaluate the classification efficiency of all presented texture operators. In order to describe the classification quality, three different parameters were calculated. Firstly, the classification efficiency is given as a ratio of number of snow particle images, which belong to the class to which were classified, to all tested images. Secondly, the probability of detection represents the fraction of correctly classified images from one class (e.g. snowflake), to all images classified as this class (e.g. snowflake). Finally, the false alarm rate corresponds to miss-classified images of one class (e.g. snowflake classified as graupel), to all images classified as this class (e.g. snowflake). The results are given in Table 2.

The best classification efficiency, 86.06%, was achieved for COM_M method. However, there are many methods (*LBP*, *MVH*, FOF_M, *MH*, $GTDM_M$) which result is around 85%. It is worth to point out three things. Firstly, in all cases classification with feature vectors based on structure data (bold font in table), gives better result in comparison to the traditionally calculated features. Probably, using all information gathered in the data structure, instead of only few parameters derived from it, improved the classification result. In times when these techniques were developed, short feature vectors were important due to calculation overload, but nowadays this problem could be neglected. Secondly, it is interesting that the results achieved by *LBP*, and local texture statistic operators *MVH*, and *MH* are almost similar. Finally, considering the additional parameters, it

Table 2
Correct classification efficiency for texture operators.

Method	Efficiency [%]	Probability of detection [%]		False alarm rate[%]	
		Snowflake	Graupel	Snowflake	Graupel
COM_M	**86.06**	84.45	88.00	46.23	38.91
LBP	85.91	85.89	85.93	47.62	47.53
MVH	85.87	83.85	88.40	45.81	36.55
FOF_M	**85.52**	82.55	89.48	44.82	31.13
MH	85.26	83.36	87.61	45.85	37.51
$GTDM_M$	**85.17**	80.19	92.97	42.51	18.46
COM_F	84.75	83.81	85.84	46.70	42.78
RLM_M	**83.15**	80.61	86.50	44.82	34.53
MMH	81.46	78.27	85.97	43.66	31.31
FOF_F	81.27	77.72	86.49	43.13	29.25
$GTDM_F$	80.21	75.77	87.44	41.51	24.17
RLM_F	55.81	57.55	54.09	48.74	52.25

occurred that all techniques better manage with graupel images than with snowflake. In all cases the probability detection is higher for graupel and the false alarm rate is lower. It is a consequence of snow particle structure, which corresponds to image quality. Generally, graupel structure is dense whereas snowflake is fluffy what on images sometimes results in low contrast. Moreover, sometimes in the case of snowflake it is difficult to precisely describe the object outline, which may result in confusing background as a particle part and vice versa.

5.2. Multiple texture operators

Describing the snow particle image by one texture operator is sufficient for classification. However, this experiment checks whether it is possible to improve the results when two texture operators are implemented. In order to create pairs of texture operators only these which efficiency was higher than 80 % in previous experiment are considered. Therefore, 55 pairs of multiple texture operators were created. Due to the lack of space, Table 3 presents results for these pairs where class efficiency is higher than the best score in a previous experiment (86.06 %).

This experiment shows, that exploring information concerning an image texture from two texture operators, can improve the classification. In the best case — $LBP - GTDM_M$ pair — the classification efficiency is 87.89%, which is almost 2% higher than in previous experiment. The results are a little bit disappointing, because the improvement in classification is not considerable. This result suggests that, however, the texture operators are calculated using different approaches, the information stored in them is redundant. Therefore, combining more of them will not improve the classification efficiency, because no new information will be added.

Table 3
Classification efficiency for multiple texture operators.

Method A	Method B	Efficiency [%]
LBP	$GTDM_M$	87.89
LBP	COM_M	87.60
LBP	FOF_M	87.56
FOF_F	RLM_M	87.41
LBP	*MVH*	87.11
FOF_F	COM_F	87.04
COM_M	FOF_M	86.58
LBP	*MH*	86.95

5.3. Dimension reduction

The length of feature vectors of most of the presented texture operators is considerably long. Detailed lengths are presented in Table 4. Moreover, the results from previous experiment suggest that probably the information stored in the feature vector is redundant. Therefore, this experiment checks the influence of reduction of feature vector length on the classification performance.

While reducing the data dimensions, it is important to remove the redundant data, but preserve the information, which permit to distinguish between objects. In case of dataset with high-dimensionality, when it is impossible to visualize data to find patterns governing the dataset, the statistic tools become useful. Principal component analysis (*PCA*) is a statistic tool used to solve this kind of problems. This mathematical procedure uses Karhunen–Loève transform to find a set of principal components (eigen vectors), which describe the data. The set of the principal components can be understood as a new set of axes, which follows some rules. First of all, the axes are defined in this direction where the highest variability of the data is noticed. Therefore, the number of principal components can be smaller or equal to the number of original features. Next, for each principal component a value (eigen value) is given. It corresponds to the data variability in this direction. It is important to remember that the eigen vectors are orthogonal. In order to apply *PCA* on the dataset it should be assured that its mean is zero. In other words, the mean value for each feature should be subtracted from the dataset. Fig. 12 presents an exemplary two-dimensional data distribution and its eigen vectors.

Let $E = [e_1, e_2, \ldots, e_l]$ to be a matrix of eigen vectors, e_i, and $EE = [ee_1, ee_2, \ldots, ee_l]$ represents a vector of eigen values, ee_i, corresponding to the eigen vectors. Eigen vectors, e_i, in E should be ordered according to the decreasing value of its eigen value, ee_i. As it was mentioned, the eigen value describes the variability of data in direction defined by the eigen vector. From other point of view, this value corresponds to the information which is described by this eigen vector. However,

Table 4
Feature vector length after PCA transformation with assurance that at least N percent of information is stored in new data representation.

Method	Data dimension vs. information			
	Original	99%	95%	90%
RLM_M	640	65	42	30
LBP	416	101	33	5
$GTDM_M$	256	79	53	34
MVH	256	19	8	6
COM_M	256	8	6	4
FOF_M	64	15	7	5
MH	64	12	6	4
MMH	64	9	6	4
COM_F	28	1	1	1
RLM_F	12	1	1	1
FOF_F	6	2	1	1
$GTDM_F$	5	1	1	1

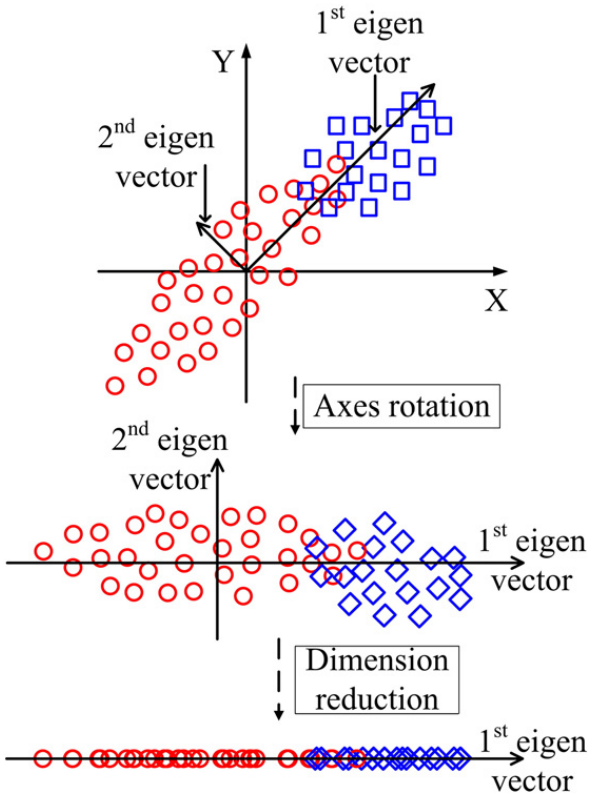

Fig. 12. Exemplary data distribution on Cartesian plane. Additional axes correspond to eigen vectors direction. The length of each axis reflects the eigen value.

it is more convenient to present ee_i in percents. Then each eigen value is recalculated according to the following formula:

$$ee_i = \frac{ee_i}{\sum_{k=1}^{l} ee_k}. \tag{14}$$

Then the decision concerning the number of eigen vectors necessary to describe the data is dependent upon equation

$$\sum_{k=1}^{m} ee_i > PE, \tag{15}$$

where *PE* corresponds to the percent of information that should be available in a new set with diminished dimensions and m is the number of chosen eigen vectors. Hence, for 100% $m = l$. Otherwise, a new matrix $E_{new} = [e_1, \ldots, e_m]$ is created. Finally, the dataset with reduced dimensions, O_{new}, is achieved according to the equation:

$$O_{\text{new}} = E_{\text{new}}^{T} O, \tag{16}$$

where O denotes original data, and o is an object feature vector (which has p dimensions, and $p \geq m$).

Table 4 concludes the application of *PCA* in regard of feature vector length. In the second column, the length of a feature vector is given before the dimension reduction. The columns from three to five contain the feature vector lengths achieved after application of PCA. The difference is the percentage of how many features from original information remained. Generally, the bigger reduction of the feature vector length means, that the original data is characterized by high redundancy. For instance, in case of RLM_M method using 99% of original data information, reduces the feature vector length to 10%. Similarly, after reduction, the feature vector calculated for COM_M contains only 8 elements from 256 at the beginning (that is only 3% of the original length). On the other hand, when one decides to lose more information from the original dataset, it is possible to compress the feature vector even more (please see columns 4 and 5). However, in these cases the amount of lost data may become crucial, and diminish the descriptive properties of the feature vector. It is also interesting to notice, that in case of feature vectors based on features, the data spread only in one direction. Therefore, the length of the feature vector after dimension reduction is one.

Next, Table 5 gathers correct classification efficiency calculated for data of reduced dimensionality. This table does not present results for all methods which feature vector length, due to *PCA* application, with its size reduced to one dimension. Unfortunately, this reduction made the correct classification impossible in all these cases. For texture operators, with very long feature vectors, the dimension reduction improves the classification considerably, as in the best cases 100% accuracy was achieved for *MMH*, COM_M, $GTDM_M$, and FOF_M. It seems that removing the redundant data permit the *kNN* classifier to describe between class boundary more precisely.

5.4. Examples of snow particle classification

This section presents some examples of snow particle and its classification results. For consideration, the *MH* method was chosen, because its feature vector could be easily visualized as a histogram. This method also returned high correct classification ratio: in the case of one dimensional classification it was 85.26%, and after application of *PCA*, 95% of original information, the result increased to 99.24%, while reducing the feature vector length from 64to 6 elements.

Fig. 13 presents examples of snow particles with calculated feature vectors (Fig. 13b, c, e, and f) and additionally average histograms of feature vectors achieved for each class: graupel (Fig. 13a) and snowflake (Fig. 13d). It is not surprising that the shape of the average class histograms of feature vectors corresponds to the average class histogram of image intensity presented in Fig. 1. It is a consequence of titMH method definition, which can be seen as a smoothed version of the intensity histogram, where additionally the number of bins

Table 5
Classification efficiency achieved for datasets created after applying PCA with various information reductions.

Method	Classification efficiency [%] vs information [%]		
	90	95	99
MMH	97.17	100.00	100.00
COM_M	79.78	100.00	100.00
$GTDM_M$	99.87	99.86	100.00
FOF_M	99.83	99.96	100.00
MVH	97.84	99.50	99.99
LBP	99.90	99.96	99.97
RLM_M	99.36	99.24	99.01
MH	96.11	99.24	97.42
FOF_F	-	-	79.66

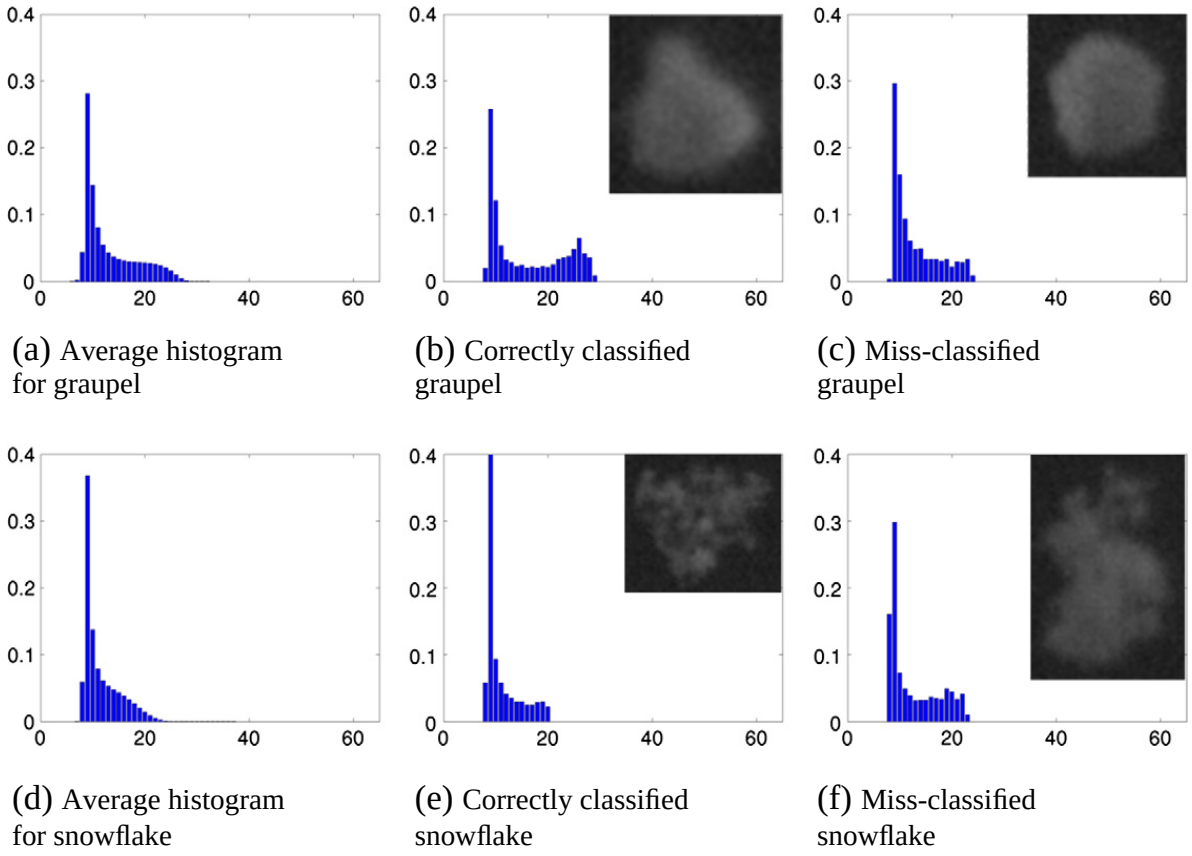

Fig. 13. Visualization of feature vector achieved for *MH* method.

was reduced four times. In both cases, the slope of the histograms for the snowflake class is straight whereas the shape of the histograms achieved for the graupel class resembles a saddle. Particles presented in Fig. 13 are examples of correct and miss-classified objects when one dimensional classifier was applied. However, the *kNN* decision is based on the Euclidean distance calculated in 64 dimensional space, one can notice that the particles which feature vector shape reflecting the saddle was classified as graupel (see Fig. 13b and Fig. 13f), whereas these described by feature vector which histogram has straight slope was classified as snowflake (see Fig. 13c and Fig. 13e).

Applying *PCA* before classification allowed for correct classification of all particles. Unfortunately, the applied software does not return the feature vectors achieved after the dimension reduction. Therefore, the shape of the reduced feature vector is unknown. However, looking at the exemplary feature vectors, it is worth to point out that in all cases only features between 10 and 30 bin vary, others are always set to 0. Next, from the range $10-30$, one can see that the biggest variation is in the end of the set, and probably this part is represented in the reduced feature vector.

6. Conclusions

This work presents an application of texture operators for snow particle images classification into the snowflake and graupel. Many texture operators were considered, namely: first order features, co-occurrence matrix, run length matrix, grey-tone difference matrix, and local binary pattern. Moreover, a novel approach which design simple local statistic operators was introduced. Here, three new texture operators were presented: mean histogram, min–max histogram, and mean–variance histogram. Finally, it was suggested to use feature vector based on the data structure, which is created in a middle stage by many of methods, instead of using a feature vector based on features only.

Calculating the correct classification efficiency with k-nearest neighbourhood classifier shows that all these techniques except RLM_F describe the object well and allow to classify with accuracy above 80%. The best result was achieved for COM_M texture operator and its correct classification efficiency is 86.06%. It was noticed, that utilizing feature vectors based on structures, in comparison with feature vectors based on features, gives better results. A next experiment checked whether it is possible to improve the correct classification accuracy by using more texture operators in order to describe an image. Although, the correct classification efficiency increased up to 87.89% for $LBP-GTDM_M$ pair, generally the improvement is not satisfactory. Moreover, it was found, that the redundancy in the data is so high, that applying more texture operators for object description does not increase the information. Therefore, principal component analysis was applied in order to remove the redundant data (and also to reduce the feature vector length). In consequence, when keeping 99% of original information describing the object, it was possible to reduce the feature vector length even down to 10% of its original length in case of RLM_M texture operator. Next, using reduced feature vectors for class description, it improved

the classification considerably, as in case of *MMH*, COM_M, $GTDM_M$, FOF_M the correct classification efficiency was 100%.

Using a snow particle classification system of high accuracy allows to easily distinguishing between the precipitation types. Combining the information about the precipitation type with precipitation rate and radar readouts is a next step in the research which aims to find better coefficients describing the Z–*R* relation. It is hoped, that finding accurate parameters, which bound the radar reflectivity factor with measured precipitation rate, allow for good precipitation forecasting. On the other hand, applying the snow particle classification system in future research is also possible. As the snow particle type depends on the developing stage of cloud, it is believed that this system could help in research concerning cloud growth and development.

References

Albregtsen, F., Nielsen, B., Danielsen, H.E., 2000. Adaptive grey level run length features from class distance matrices. Proc 3, 3746 (ICPR'00).

Amadasun, M., King, R., 1998. Textural features corresponding to textural properties. Trans. Syst. Man. Cybern. 19 (5), 1264–1274.

Chu, A., Seghal, C.M., Greenleaf, J.F., 1990. Use of grey value distribution of run lengths for texture analysis. Pattern Recognit. Lett. 11, 415–450.

Frank, G., Härtl, T., Tschiersch, J., 1994. The pluviospectrometer: classification of falling hydrometeors via digital image processing. Atmos. Res. 34, 367–378.

Galloway, M.M., 1975. Texture analysis using grey level run lengths. Comput. Graphics Image Process. 4, 172–179.

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. Trans. Syst. Man Cybern. 610–621 (SMC-3).

Harimaya, T., Kodama, H., Muramoto, K., 2004. Regional differences in snowflake size distributions. J. Meteorol. Soc. Jpn. 82 (3), 895–903.

Loffler-Man, M., Blahak, U., 2001. Estimation of equivalent radar reflectivity factor from measured snow size spectra. J. Appl. Meteorol. Climatol. 40, 843–849.

Marshall, J.S., Gunn, K.L.S., 1952. Measurement of snow parameters by radar. J. Atmos. Sci. 9, 322–327.

Materka, A., Strzelecki, M., 1998. Texture analysis methods — a review. CPST B11 report. Technical University of Lodz, Institute of Electronics, Brusselss.

Matrosov, S.Y., 1992. Radar reflectivity in snowfall. Trans. Geosci. Remote Sens. 30, 454–461.

Matsuo, T., Mizuno, H., Murakami, M., Yamada, Y., 1994. Requisites of graupel formation in snow clouds over the Sea of Japan. Atmos. Res. 32, 55–74.

Mizuno, H., 1992. Statistical characteristics of graupel precipitation over the Japan Islands. J. Meteorol. Soc. Jpn. 70, 115–121.

Nurzyńska, K., Kubo, M., Muramoto, K., 2010a. Grey scale texture classification method comparison considering object and lighting rotation. Proc. ICMV 2010, Hong Kong, 28–30 December 2010. .

Nurzyńska, K., Kubo, M., Muramoto, K., 2010b. 2D feature space for snow particle classification into snowflake and graupel. IEICE Trans. 3344–3351.

Nurzyńska, K., Kubo, M., Muramoto, K., 2011a. Shape parameters for automatic classification of snow particles into snowflake and graupel. Meteorological Applications, Oct. 4, 2011. , http://dx.doi.org/10.1002/met.299.

Nurzyńska, K., Kubo, M., Muramoto, K., 2011b. Snow particle automatic classification with texture operators. Proc. IGARSS'2011. 24–29 July, Vancouver, Canada. .

Ojala, T., Pietikäinen, M., Mäenpää, T., 2000. Grey scale and rotation invariant texture classification with local binary patterns. Lect. Notes Comput. Sci. 1842, 404–420.

Ojala, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution grey-scale and rotation invariant texture classification with local binary patterns. Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987.

Prodi, F., Caracciolo, C., D'Adderio, L.P., Gnuffi, M., Lanzinger, E., 2011. Comparative investigation of Pludix disdrometer capability as Present Weather Sensor (PWS) during the Wasserkuppe campaign. Atmos. Res. 99 (1), 162–173.

Tang, X., 1998. Texture information in run-length matrices. Trans. Image Process. 7 (11), 1602–1609.

Vössing, H-J., Borrmann, S., Jaenicke, R., 1998. In-line holography of cloud volumes applied to the measurement of raindrops and snowflakes. 49, 199–212.

Zucker, S.W., Terzopoulus, D., 1980. Finding structure in co-occurrence matrices for texture analysis. Comput. Graphics Image Process. 12, 286–308.

Karolina Nurzyńska received her M.E. and Ph.D. degree in the field of computer science from the Silesian University of Technology, Poland in 2005 and 2009, respectively. She was a researcher in the School of Electrical and Computer Engineering, Kanazawa University. Her research interests include image processing and 3D surface reconstruction.

Mamoru Kubo received B.E. and M.E. degrees from Nagoya Institute of Technology, Japan in 1990 and 1992, respectively. He is an assistant professor in the School of Electrical and Computer Engineering at Kanazawa University, Japan. His research interests include photogrammetry and remote sensing. He is a member of IEEE.

Ken-ichiro Muramoto received B.E. and M.E. degrees from Toyama University, Japan in 1971 and 1973, respectively. He received the Ph.D. degree in Engineering from Kyoto University, Japan in the field of image information science. He has also received the doctor of medical science degree from Toyama Medical and Pharmaceutical University in the field of neurophysiology. He was a professor in the Faculty of Electrical and Computer Engineering, Institute of Science and Engineering at Kanazawa University, Japan. He is currently the president in the Ishikawa National College of Technology, Japan. His research interests include image processing, pattern recognition, human vision and remote sensing. He is a member of IEEE.