AIRBORNE CROWD DENSITY ESTIMATION

Oliver Meynberg*, Georg Kuschk

Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Wessling, Germany
(oliver.meynberg, georg.kuschk)@dlr.de

Commission III/VII

KEY WORDS: Crowd detection, Aerial imagery, Gabor filters, Texture features

ABSTRACT:

This paper proposes a new method for estimating human crowd densities from aerial imagery. Applications benefiting from an accurate crowd monitoring system are mainly found in the security sector. Normally crowd density estimation is done through in-situ camera systems mounted on high locations although this is not appropriate in case of very large crowds with thousands of people. Using airborne camera systems in these scenarios is a new research topic. Our method uses a preliminary filtering of the whole image space by suitable and fast interest point detection resulting in a number of image regions, possibly containing human crowds. Validation of these candidates is done by transforming the corresponding image patches into a low-dimensional and discriminative feature space and classifying the results using a support vector machine (SVM). The feature space is spanned by texture features computed by applying a Gabor filter bank with varying scale and orientation to the image patches. For evaluation, we use 5 different image datasets acquired by the 3K+ aerial camera system of the German Aerospace Center during real mass events like concerts or football games. To evaluate the robustness and generality of our method, these datasets are taken from different flight heights between 800m and 1500m above ground (keeping a fixed focal length) and varying daylight and shadow conditions. The results of our crowd density estimation are evaluated against a reference data set obtained by manually labeling tens of thousands individual persons in the corresponding datasets and show that our method is able to estimate human crowd densities in challenging realistic scenarios.

1 INTRODUCTION

Monitoring large crowds is an important topic in the field of security surveillance as it can provide crucial information for decisions by the local security forces, as for example detecting a hazardous situation which may result in a panic. For mass events with thousands of people, the only feasible method for crowd monitoring is by airborne camera systems, due to sheer scale and the limited field of view from in-situ cameras. Manual crowd estimation by human observers is possible when enough trained experts are at hand, but in general this is not the case and often also a question of costs and time. Therefore it is highly desirable to employ airborne camera systems which are able to monitor human crowds at such large events.

For ground based scenarios, (Lin et al., 2001) train a classification system on head-like contours to detect and count people in indoor scenarios. For this system to work, the camera needs to be roughly on the same height like the observed people, resulting in the same limited field of view as for a human observer in the same situation, and is therefore only applicable for small indoor scenarios. The work (Ghidoni et al., 2012) uses co-occurence matrices of the input images to measure change in image texture. As this method is not invariant in scale, rotation or even intensity, this method only works for stationary cameras as for example in stadiums. A very good work done by (Arandjelovic, 2008) uses a sliding window approach in scale space to create a bag-of-words descriptor for each image patch using clustered SIFT features and classifying them using a SVM. However, their result is just a binary crowd mask and the evaluation is done by comparison w.r.t. a hand-segmented ground truth, and ergo has quite a strong subjective bias towards the labeling persons definition of a crowd. In a different terrestrial setup, (Aswin C et al., 2009) are using a multi camera setup and background subtraction to detect single persons (and classifying their activities using pre-learned linear dynamical systems).

In the field of aerial crowd detection and crowd density estimation, (Hinz, 2009) uses sequences of temporally consecutive and overlapping images to estimate the background by applying a gray-level bounded region-growing approach. Then, a blob detector (in the paper called Laws texture filter) is used on the foreground pixels to filter out non-crowd-like objects. To estimate the crowd density, a Gaussian smoothing kernel with a fixed standard deviation / bandwidth is applied. This is an often used approach for density estimation which we slightly adjust to our needs. Unfortunately, a quantitative evaluation is missing. In the work of (Sirmacek and Reinartz, 2011), the FAST feature detector is applied to detect blob-like and corner-like image structures. To filter out non-crowd / non-people responses in highly cluttered background, in a second step image segmentation is used to remove segmented areas which are too small (crowds are assumed to be placed atop of a uniform looking surface) and which contain less than a certain amount of local features. This approach seems to work well when the parameters are tuned to a given dataset, but it is clearly non scale or color/contrast invariant, as the chosen parameters for the image segmentation algorithm and the thresholds for the number of local features and minimum area size need to be tuned for every dataset anew. However, the herein proposed method of determining the bandwidth for the kernel density estimation via the mean of the minimum nearest distances of the detected people gives promising results for estimating the crowd density.

2 METHOD

As the image space Ω we deal with, is made up by multiple images in a range of $\approx$ 21MPix, in a first step we need to apply a fast method to reduce the search space for the following time consuming feature extraction and classification steps. To that end we apply the FAST interest point detector (Rosten and Drummond, 2006) to extract all corner- and blob-like structures in the image, as a crowd populated by many individual persons is expected to

consist of an agglomeration of such structures.
In a second step we extract image patches around these interest points and extract scale- and rotational invariant texture feature descriptors by Gabor filtering these image patches. The resulting feature descriptors are classified by a support vector machine (Drucker et al., 1997) and the positively classified crowd areas are then used to estimate a crowd density for the whole image, based on kernel density estimation with an automatically computed kernel bandwidth.
For filtering out non-crowd image patches in the second step it is of course also possible to use additionally provided road maps, building footprints or digital elevation models, but as this data usually is not available to the local operators and would restrain the usage of our method, we do not consider these options. Furthermore we decided against a single person detection, as the results would be highly unreliable using typical nowadays aerial imagery taken from a law-regulated flight altitude of 1000-1500m, where a person seen from directly atop covers roughly 5-10 image pixels (see Figure 1 for examples).
The single components of our complete workflow are as follows

1. Detect FAST features $F_k \in \Omega$
2. Extract image patches I_k around F_k
3. Create texture feature descriptors v_k by Gabor filtering I_k
4. Classify v_k using a trained SVM
5. Estimate crowd density based on positive detections

and will be described in detail in the following sections.

2.1 Interest Point Detector

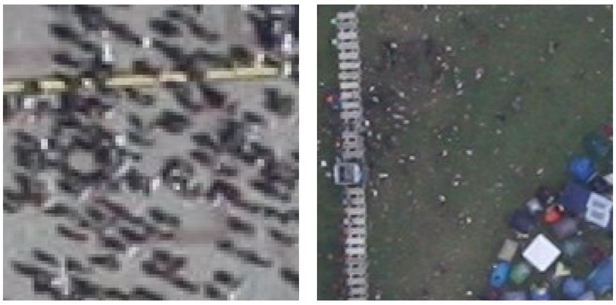
To restrict the search range for image patch based classification from all possible pixel positions (e.g. 21MPix) to a small number of probable candidates (50,000), we first apply an interest point detector as a very rough but fast initial filtering of all image positions into regions possibly containing crowd areas or non-crowd areas. The design of the interest point detector does not need to be overly complicated, as long as it detects local image intensity changes (see the characterization of crowd areas in the following Section 2.2). Further, we do not want to detect intensity changes along regular edges, but to detect blob-like and corner-like small elements. Typical operators for this task are the Harris corner detector (Harris and Stephens, 1988) or the Laplacian of Gaussian. However, the FAST corner detector (Rosten and Drummond, 2006) was specifically designed to detect these points of interest in a minimum of computational time, which is why we use these FAST features for initial filtering of possible crowd areas.

2.2 Feature Extraction

Seen from an aerial observer's side, a human crowd is characterized by an image region containing a number of very small subregions which are differing in brightness or color from their surrounding. These subregions or blobs should be further randomly distributed (ranging from dense to sparse), exhibiting no recognizable pattern (see Figure 1 for typical examples). And as we are interested in crowd detection only, all other image regions are deemed to not contain crowds.
We thus need to transform this intuitive characterization of properties of an image patch $I \in \mathbb{R}^{M\times N}$ into a formal operator F, with which we can automatically compute a distinctive and reliable feature space $\mathbb{R}^d$

$$F : \mathbb{R}^{M\times N} \rightarrow \mathbb{R}^d \tag{1}$$

The resulting d-dimensional feature space should have low intraclass variability (one crowd area should result in a similar feature vector as a completely different crowd area) and a high interclass variability (feature vectors of crowd areas should be distinct from feature vectors from non-crowd areas).



(a) High crowd density plus shadows (b) Low image resolution and low contrast

Figure 1: Examples of 64×64 image patches containing human crowds in a challenging aerial imagery. A reliable estimation of single persons is not possible due to the low image resolution.

Gabor filters are a highly suitable choice for the transformation F, with their frequency and orientation representations being similar to the human visual system. Introduced to computer vision by (Daugman et al., 1985) and (Daugman, 1988) they have been found to be particularly appropriate for texture representation and discrimination. Usage ranges from retrieval of image data, e.g. (Manjunath and Ma, 1996) and (Han and Ma, 2007), to optical character recognition (Wang et al., 2005) and fingerprint feature extraction (Lee and Wang, 1999), just to name a few.
A 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal wave in a specified direction (Figure 2), with the response of a 2D image convolution being highest for image gradients along the filters orientation. Simply speaking, Gabor filters are detecting image gradients of a specific orientation. The convolution of image patches with a number of different scales and orientations allows for extraction and encoding of local texture information into a low dimensional feature vector, usable for generic classification.
The motivation behind our choice of Gabor filters is that a crowd area, having the aforementioned characteristics of randomly distributed blobs, should give a high response in every direction, whereas regular man made structures only have high responses orthogonal to their main orientations and natural structures should give a similar response in every direction, but due to lack of contrast of a lower magnitude than for crowds.

As their definitions are manifold, and in order for this paper to be self-contained, we describe our choice of Gabor filters used throughout the rest of the paper in the following:
The mother wavelet of the two-dimensional complex Gabor func-

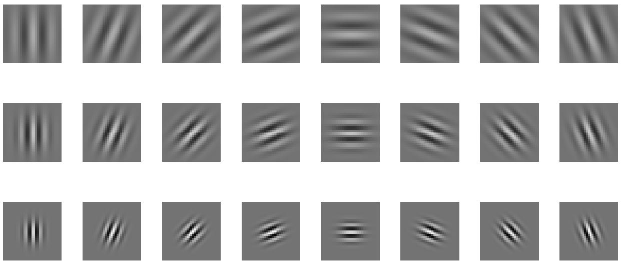


Figure 2: Gabor filter bank for 3 scales and 8 orientations

tions is defined as

$$\begin{aligned} g(x,y) &= \frac{1}{2\pi\sigma_x\sigma_y} \cdot \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi iWx\right) \quad (2)\\ &= G(x,y)\cdot \exp\left(2\pi iWx\right) \end{aligned}$$

with

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \cdot \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right) \quad (3)$$

The real and imaginary parts of this wavelet are computed as

$$\begin{aligned} g_{\text{re}}(x,y) &= G(x,y)\cdot cos\left(2\pi Wx\right)\\ g_{\text{im}}(x,y) &= G(x,y)\cdot sin\left(2\pi Wx\right) \quad (4) \end{aligned}$$

A filter bank of gabor functions $g_{s,k}(x,y)$ is now generated by rotating and scaling the mother wavelet $g(x,y)$ as follows

$$\begin{aligned} g_{s,k}(x,y) &= a^{-s}g(x',y') \quad (5)\\ x' &= a^{-s}(xcos\theta_k + ysin\theta_k)\\ y' &= a^{-s}(-xsin\theta_k + ycos\theta_k) \end{aligned}$$

with angles $\theta_k = k\pi/K$ $(k = 0,..,K-1)$, K being the number of orientations, and the scaling factor a^{-s} assuring that the energy of the filter is independent of the scale $s = 0,..,S-1$ (S being the number of scales). Following the argumentation of (Manjunath and Ma, 1996) to reduce the redundancy of the resulting nonorthogonal Gabor wavelets, we set $a = (U_h/U_l)^{1/S}$, resulting in $W = U_h/a^{S-s}$. The upper and lower center frequency of interest are set to $U_h = 0.4$, $U_l = 0.1$ and σ_x, σ_y are set accordingly.

Given an image patch $I \in \mathbb{R}^{M\times N}$, its Gabor wavelet transform is now defined as

$$\begin{aligned} W_{s,k}(x,y) &= I(x,y) * g_{s,k} \quad (6)\\ &= \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} I(i-M/2, j-N/2)\cdot g_{s,k}(i,j) \end{aligned}$$

The corresponding texture attributes for one such wavelet transform, given a scale s and orientation k, are computed as the mean and standard deviation of the absolute valued filter response

$$\begin{aligned} \mu_{(s,k)} &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} |W_{s,k}| \quad (7)\\ \sigma_{(s,k)} &= \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left(|W_{s,k}| - \mu_{(s,k)}\right)^2 \end{aligned}$$

and the full feature vector f constructed by applying a whole filter bank of Gabor wavelets (e.g. $S = 3$ and $K = 8$) to the image patch I and concatenating the resulting attributes to a 48-dimensional vector

$$f(i) = \left[\mu_{(0,0)}, \sigma_{(0,0)}, \mu_{(0,1)}, \dots \mu_{(2,7)}, \sigma_{(2,7)}\right] \quad (8)$$

encoding, laxly speaking, the amount of edges in a number of different orientations. Note that due to taking the absolute value of the filter response, it does not matter whether a person appears dark on bright background or vice versa.

As a naive 2D convolution of image patches $I \in \mathbb{R}^{N\times N}$ with Gabor filters $g \in \mathbb{R}^{M\times M}$ would result in an intractable complexity of $\mathcal{O}(N^2M^2)$, we transform both the image patch and the filter from spatial domain to frequency domain by the fast Fourier transform, multiply the elements pointwise and transform the result back to obtain our feature attributes, resulting in a complexity of $\mathcal{O}(N^2logN^2 + M^2logM^2)$. For typically sized image patches $I \in \mathbb{R}^{64\times 64}$ and Gabor filters $g \in \mathbb{R}^{32\times 32}$ the speedup factor is around factor 70. We further sped up the algorithm by precomputing the FFT's of the different Gabor filters only once and by making use of multi core parallelization. The border pixels of an image patch need to be treated with care, both in the spatial and frequency domain. Therefore, before computing the Fourier transform, we apply a windowing function to the input image, to reduce the ringing effects. Note that in the spatial domain we would have similar problems.

2.3 Classification

As we cannot assume our feature vectors to be linear separable with respect to their classes, statistical distance measures like the Mahalanobis distance would perform quite bad. In constrast, kNN classifiers (k nearest neighbors) are fully capable to handle multimodal distributions in the feature space, but are quite sensitive to overlapping / mixed distribution class areas. Support vector (regression) machines (SVM) (Drucker et al., 1997) instead are combining both of the aforementioned advantages, as they are specifically designed for non-linear classification and, based on statistical regression, are robust to areas of overlapping class distributions. The only care has to be taken for highly imbalanced training data, where the number of positive samples of one class is much larger than the other classes. Simple over-sampling (synthetically generating positive samples) is taking care of this issue though.

2.4 Crowd Density Estimation

To estimate a crowd density based upon our classification results, we consider the crowd density as a probability density function (pdf) over the image domain. Since the classification results provide only a finite and very sparse set of N sampled data points of this pdf, we need to apply additionally data smoothing inbetween, inferring about the real underlying pdf. A standard method to this end is the kernel density estimation, defined as

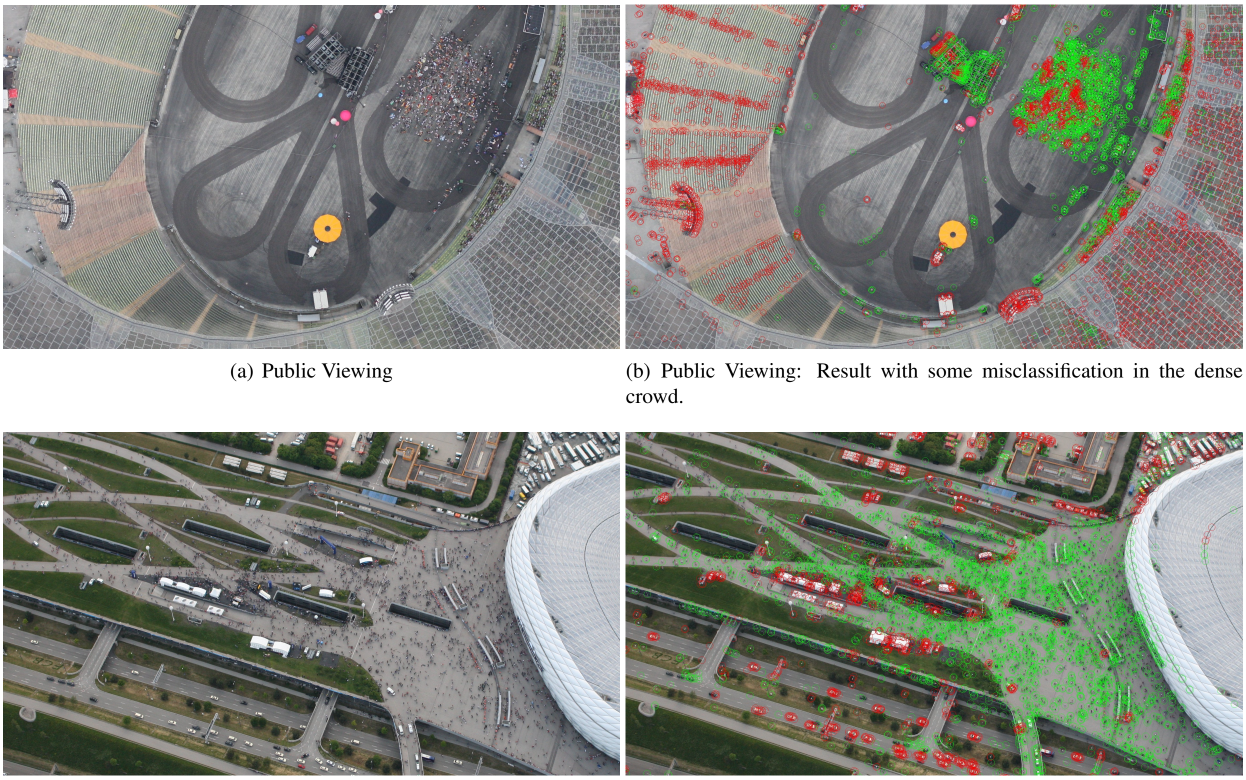
$$f_h(x) = \frac{1}{N\cdot h^d}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right) \quad (9)$$

with the kernel function K typically being a Gaussian distribution. The only problem here is the choice of the smoothing parameter (also called bandwidth) h. As the crowd detection algorithm is required to work on images of different scale, the size and distances between two identical image features can vary in image space, when the images are taken from different distances in world space or with a different image resolution. Therefore, an automatic data-driven bandwidth selection is needed, adapting for every image or scale respectively.

3 EVALUATION

Instead of hand-segmenting human crowds and comparing a computed binary crowd mask, we evaluate the accuracy of the estimated crowd density in a continuous way without applying any thresholds. We choose this approach because in reality there is also no artificial threshold where you distinguish between certain crowd density levels. Our proposed method uses original, not-orthorectified JPEG images and does not rely on additional information like road maps or building plans as our method should work ad-hoc without additional preprocessing steps.

The images are all taken with the DLR 3K camera system which consists of three non-metric Canon EOS 1Ds Mark III cameras.



(a) Public Viewing

(b) Public Viewing: Result with some misclassification in the dense crowd.

(c) Football Stadium

(d) Football Stadium: Result with lowest $RMSE$ and D_{KL} in this test run

Figure 3: Test images with classification results of the detected interest points. green ≙ classified as crowd, red ≙ classified as non-crowd

Each camera has a full frame CMOS sensor with a resolution of 21 MPix. The used lenses are a Zeiss Makro Planar 2/50mm and a Zeiss Distagon T 2/35mm.

3.1 Test Data

Evaluation of the proposed algorithms is done on five different aerial image datasets (see Table 1). The City Centre dataset consists of three images showing the crowded shopping streets around Munich's New Town Hall on a sunny day. While the individual persons and their shadows are clearly visible, other regions in the image consist of small dense crowds where single persons occlude each other. Due to the lower flight altitude the Public Viewing data set has a slightly higher resolution. In some images of this scene one large group sits on the ground of an arena in front of a screen and smaller but also dense groups stand in other areas (Figure 3a). The lighting conditions are mixed with images containing shadows and no shadows respectively. The Southside data set shows images of a rock festival from two different flight altitudes (1000m/1500m) and different angles. The scene has both dense crowds and individuals. The main challenge in this dataset is the huge campground with a lot of small tents. The interest point detector apparently detects all the corners of the tents which does not really lead to a reduction of the search space. Moreover individuals standing between tents can hardly be detected. The RockamRing dataset shows another rock festival with dense and sparse crowds (Figure 6a, details in Section 3.2). It was acquired with an older camera system which is why the GSD is worse than in the other datasets. Finally, the images of the Football Stadium dataset show the crowded area in front of a stadium from a side-view perspective which results in varying GSDs. The reference data is acquired by manually marking each single person in the test images. The "Images" column in Table 1 indicates the number of manually marked images per data set, the "Point of view" column indicates if the data set contains only nadir images or also images from a sideview perspective.

Dataset	GSD [cm]	Images	Point of View
City Centre	13	3	nadir
Public Viewing	10	10	nadir/side
Southside	13/18	5	nadir/side
Rock am Ring	20	4	nadir
Football Stadium	10 (varies)	10	side

Table 1: Overview of the five different data sets.

3.2 Accuracy evaluation

We evaluate the accuracy of the classified images by comparing them with the reference images. Single persons in these reference images were labeled by different human interpreters. Then we apply the same image smoothing kernel as defined in Section 2.4 on these reference images resulting in a Gaussian-filtered image with highest intensities at regions with the highest number of labeled persons. To actually compare the pdfs of the reference and of the classified image we normalize both pdfs and convolve the image with the same Gaussian kernel. Both resulting images are represented by P and Q in the following. Without loss of generality the kernel's radius and standard deviation were selected beforehand, based on the choice of different human interpreters who decided which number of persons per area is sufficient to regard a specific area as crowded.

For similarity measurement we calculate the mean absolute error $MAE = \frac{1}{n}\sum_{i=1}^{N}|p_i - q_i|$, the root mean square error

$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{N}(p_i - q_i)^2}$, and the Kullback-Leibler divergence $D_{KL} = (P, Q) = (P - Q)\log\left(\frac{P}{Q}\right)$ of the difference between the classified P and the respective reference image Q. $D_{KL} \in [0, \infty)$ where $D_{KL} = 0$ means $P = Q$. Apparently, the smaller the errors are the more similar the images are and the better is the classification result.

The classifier training runs on one manually selected image of the dataset with a certain parameter combination which the classifier uses for all images in the dataset. Then, this procedure of training and classifying with the same parameters repeats for many combinations. Concretely, we tried Gabor filters with different scales, orientations, filter radii, and image patch widths. The parameter ranges are listed in Table 2. We tried all possible combinations in the given range with the constraint that the Gabor radius must not be bigger than half the width of an image patch.

Image Patch Width [pixel]	32, 48, 64, 80, 96
Gabor Filter Radius [pixel]	8, 16, 24, 32
Number of scales	2, 3, 4
Number of orientations	8,10,12,14,16,18,20,24

Table 2: Gabor filter parameters we used in the evaluation.

Figure 4 shows the resulting measures for the Publicviewing dataset. The $RMSE$ and the D_{KL} are plotted for forty of the classified images. Images 1-20 have the lowest $RMSE$ whereas Images 21-40 have the highest $RMSE$ of the whole test run which consists of several thousands of images due to the large number of possible parameter combinations. Table 3 lists the used parameters for these images. All three similarity measures often correlate with each other and show that the lower these measures are the better is the classification result. Figure 3 shows the classification results with the lowest (=best) similarity measures of the whole Public Viewing dataset and the whole Football Stadium dataset, respectively.

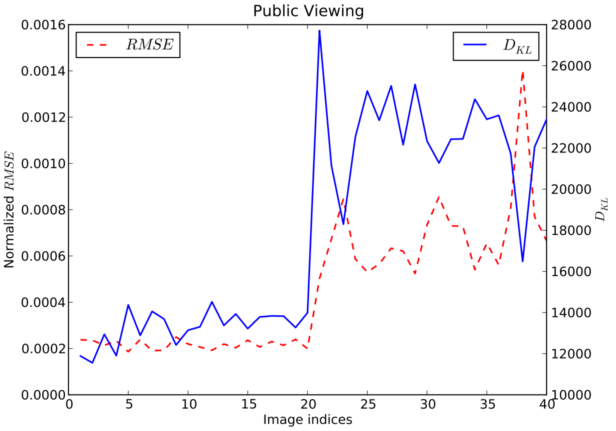


Figure 4: $RMSE$ and D_{KL} for the Publicviewing dataset. The steep slope between images 20 and 21 marks the border between images with lowest and highest errors.

The D_{KL} of the first three test images (No.1-3) for the Rockamring dataset is high although the MAE(not shown) and $RMSE$ are the lowest in this test run (Figure 5 and Table 4). D_{KL} measures the difference between the two probability distributions of the classified and of the reference image. In the case of the test images No. 1-3 the chosen parameters perform badly and classify all patches as ”non-crowd” and no patch as ”crowd” which results in a large difference of the pdfs and a large D_{KL}. Figure 6b shows the bad result. Test Image No.4, however, has a low D_{KL} which corresponds to the visualization in Figure 6c, which is much better. This is an unexpected result because how can a low RMSE be achieved for a misclassified image while for the same images the D_{KL} is high - as expected for a wrong classification.

No.	gr	scl	ori	pw	No.	gr	scl	ori	pw
1	8	3	14	48	21	24	2	8	48
2	16	2	16	48	22	24	2	16	48
3	8	3	24	32	..	..	..	..	..
..	..	..	..	..	38	16	4	24	48
19	8	4	12	48	39	24	2	12	48
20	16	3	16	48	40	8	2	8	48

Table 3: Public Viewing dataset: The images with the lowest and highest similarity measures and the used Gabor filter parameters.(No.= Image index which corresponds to the images in Figure 4, gr=Gabor radius, scl=Number of scales, ori=Number of orientations, pw=Patch width)

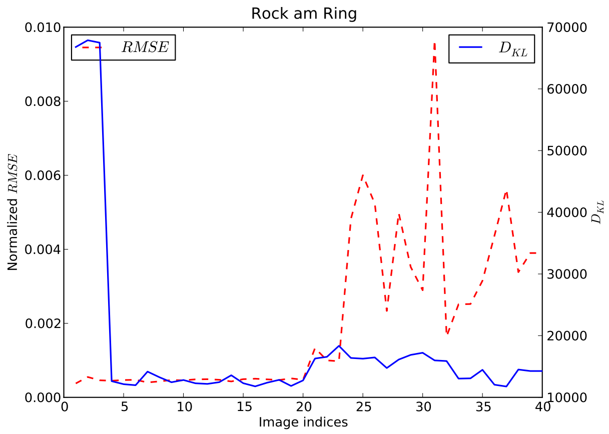


Figure 5: $RMSE$ and D_{KL} for the RockamRing dataset. Note the large D_{KL} difference between images No. 3 and No. 4, but a near constant $RMSE$.

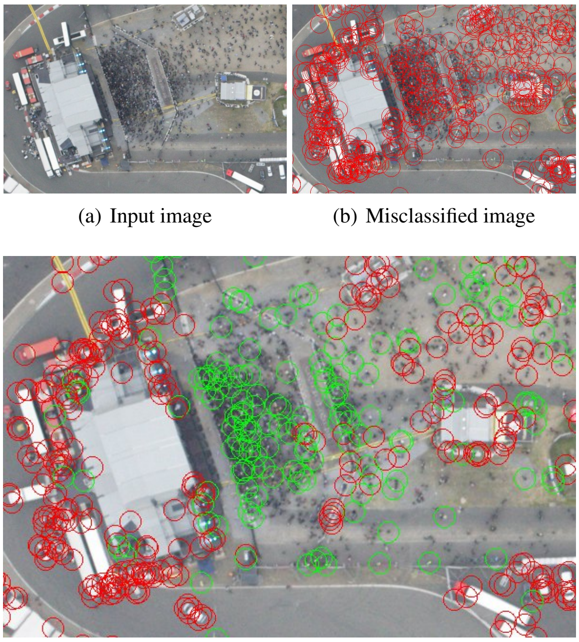
No.	gr	scl	ori	pw	No.	gr	scl	ori	pw
1	8	2	20	80	21	24	3	14	80
2	24	2	24	48	22	16	2	24	32
3	24	2	24	48	..	..	..	..	..
4	16	4	20	48	38	32	3	16	80
5	32	4	8	96	39	8	2	16	80
20	24	2	14	80	40	8	2	20	80

Table 4: Excerpt of Gabor parameters for highest and lowest similarity measure for the Rock am Ring dataset.

This case exemplarily shows the limits of this evaluation strategy, as a smoothing kernel might also remove relevant information. Another, potentially more intuitive approach, is to compare the number of manually counted people with the number of rightly classified, and detected corners per image tile.

4 CONCLUSION AND FUTURE WORK

It has been shown that despite the low spatial resolution in aerial images it is possible to perform an automatic classification of the image in regions containing lots of interest points. The preliminary filtering with the FAST corner detector is fast and allows us to concentrate on regions of interest. These image regions are convolved with a Gabor filter bank with a variety of different scales and orientations. A support vector machine classifies



(a) Input image

(b) Misclassified image

(c) Reasonably good classified (green ≙ crowd, red ≙ non-crowd)

Figure 6: Figures b and c have a low $RMSE$ when compared to the reference image. However, while Figure b also has a low $RMSE$ it has a high D_{KL} which helps us to understand the totally misclassified image. The result in Figure c is considerably better where both $RMSE$ and D_{KL} are low.

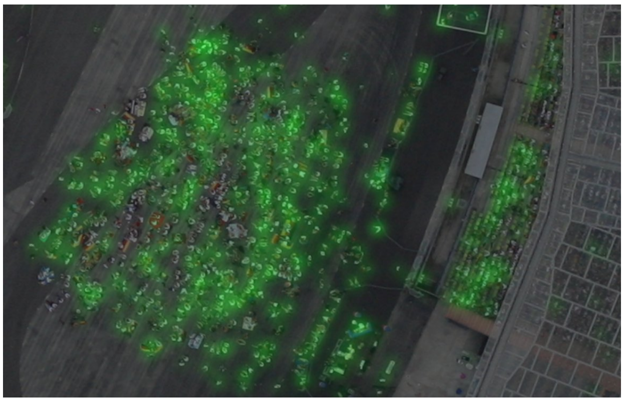


Figure 7: Estimated crowd density (High image intensities correspond to high crowd densities.)

the resulting feature space after manual training. The quality of the results clearly depends on good training samples and similar images. The Gabor filter gives some good first results, however, a good global classifier has still to be found. In our future work Gabor filters will play a role in combination with other texture features. Another major aspect is the non-discriminative nature of crowds. Crowds get more dense in a continuous way and not in discrete steps which does not fit to a binary classification. Ideally, the end user could adjust the ”level of crowd density” himself and then the algorithm shows regions with this density level in the image. We believe that this study shows the potential of modern pattern recognition methods applied on crowd density estimations in aerial images and gives some valuable hints for further investigations.

REFERENCES

Arandjelovic, O., 2008. Crowd detection from still images. In: British Machine Vision Conference, Vol. 1, Citeseer, pp. 523–532.

Aswin C, S., Robert, P., Pavan, T., Amitabh, V., Rama, C. et al., 2009. Modeling and visualization of human activities for multicamera networks. EURASIP Journal on Image and Video Processing.

Daugman, J. G., 1988. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. Acoustics, Speech and Signal Processing, IEEE Transactions on 36(7), pp. 1169–1179.

Daugman, J. G. et al., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Optical Society of America, Journal, A: Optics and Image Science 2, pp. 1160–1169.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. and Vapnik, V., 1997. Support vector regression machines. Advances in neural information processing systems pp. 155–161.

Ghidoni, S., Cielniak, G. and Menegatti, E., 2012. Texture-based crowd detection and localisation.

Han, J. and Ma, K.-K., 2007. Rotation-invariant and scale-invariant gabor features for texture image retrieval. Image and Vision Computing 25(9), pp. 1474–1481.

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. 15, pp. 50.

Hinz, S., 2009. Density and motion estimation of people in crowded environments based on aerial image sequences. In: ISPRS Hannover Workshop on High-Resolution Earth Imaging for Geospatial Information, Vol. 1.

Lee, C.-J. and Wang, S.-D., 1999. Fingerprint feature extraction using gabor filters. Electronics Letters 35(4), pp. 288–290.

Lin, S.-F., Chen, J.-Y. and Chao, H.-X., 2001. Estimation of number of people in crowded scenes using perspective transformation. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 31(6), pp. 645–654.

Manjunath, B. S. and Ma, W.-Y., 1996. Texture features for browsing and retrieval of image data. Pattern Analysis and Machine Intelligence, IEEE Transactions on 18(8), pp. 837–842.

Rosten, E. and Drummond, T., 2006. Machine learning for high-speed corner detection. Lecture Notes in Computer Science 3951, pp. 430.

Sirmacek, B. and Reinartz, P., 2011. Automatic crowd density and motion analysis in airborne image sequences based on a probabilistic framework. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, pp. 898–905.

Wang, X., Ding, X. and Liu, C., 2005. Gabor filters-based feature extraction for character recognition. Pattern recognition 38(3), pp. 369–379.