Bag of visual words — A computer vision method applied to bulk material sorting

Matthias Richter[1,2], Kai-Uwe Vieth[2], Thomas Längle[2], Jürgen Beyerer[2,1]

[1]Karlsruher Institute of Technology (KIT), Adenauerring 4, 76131 Karlsruhe

[2]Fraunhofer Institut of Optronics, System Technologies and Image Exploitation IOSB, Fraunhoferstraße 1, 76131 Karlsruhe

Corresponding author: matthias.richter@kit.edu

Abstract

Every bulk material sorting machine uses classification: objects are either discarded or accepted into one or more bins. In a classical system, both the features and the classifier are constructed by a vision engineer. While successful in the past, this approach is reaching its limits. More challenging tasks and more complex objects call for automated methods to learn both the classification rules and the features. Pattern recognition methods to do the former are well established, but the latter is still open to debate. In a previous work, we have presented an approach to the sorting of wine berries, which is based on bag of visual words [1]. In this paper, we show that the approach is not limited to this product only, but can be extended to other fields as well. In particular, we consider the task of sorting visually similar pebble stones. The method shows nearly perfect classification out of the box. Both the object descriptors and classification rules are learned from an annotated sample. User interaction was only required to obtain the annotations.

Introduction

Classification is a core task when sorting bulk materials: The material is sorted into two or more classes, for example corresponding to acceptable and defective material. The classification rules operate on descriptors that explain each object in terms of certain predefined features, e.g., color, area, length etc. To find good features, a vision engineer has to analyze the product to be sorted and then decide which features and which range of values characterize the material and the classes the best. Doing this, the engineer has to either manually select features and verify the selection on a validation sample or he has to apply some kind of automatic feature selection. Either way, classical feature descriptors are used to describe the objects with the aim of obtaining a good sorting result.

With less clearly defined classes (e.g., aesthetic qualities or degrees of ripeness of a fruit) and a growing variability in the appearance of the object, however, the manual approach becomes less and less feasible. Furthermore, rule based systems, that are often used because of their speed, straightforward implemlentation and easy interpretation, quickly become unmanageable when the material requires more complex decisions. Automatic deri-

vation of object descriptors—not necessarily based on classical features—and classification rules would clearly be an improvement.

With this goal in mind, we applied the bag of visual words approach in the context of grape sorting [1], an application that has become more and more important during the last years [2]. In this paper, we show that the approach is not limited to this specific product, but can be applied in different scenarios, more specifically in the context of mineral sorting.

Related Work

There have been several approaches to learn discriminative features from a sample of the material to inspect. As color is the most informative feature in many cases, numerous methods focus primarily on this aspect. Barni et al., for example, detect several defects on chicken meat using color alone [3]. To this end, they model the typical color distribution of each defect using a multivariate normal distribution. Each distribution effectively encodes the probability that a given color characterizes the corresponding defect. Pixels are marked as defective if the defect probability exceeds a threshold and the whole piece of chicken meat is discarded if there are too many “defective” pixels in the image. Duffy et al. use a similar method to detect burn marks on air filters[4]. Instead of using a parametric distribution, however, they derive defect probabilities from characteristic color histograms of both defective and intact surfaces. They furthermore automatically determine an appropriate threshold from training samles. In a follow-up publication, Bergasa, Duffy et al. extend the method and model the color distribution of defective pixels by a mixture of Gaussians, which they learn from training images using vector quantization [5]. In [6], Zhang et al. pursue a similar approach to grade the quality of dates. They build a training set by sorting 40 date samples into one of four classes that represent different grades of ripeness and collect a joint histogram of the red and green color channels for each of the classes. The histograms are then fused into a back projection table, where missing entries are filled in by linear interpolation using the neighboring values. Finally, the ripeness of a fruit is estimated according to color statistics of the back projected query image. Like-minded, Li et al. propose an elaborate method to assess the ripeness of tomato fruits [7]. They (manually) quantize the HSV color space into 144 bands that roughly correspond to human color perception of tomato fruits. The luminance component is discarded and the hue and saturation channel are mapped to a one dimensional subspace. Here, dominant colors are found using cluster analysis. Finally, they derive template histograms of dominant colors for five levels of ripeness from an annotated sample. An unknown fruit is classified by matching its histogram of dominant colors against each of the templates. A similar idea is also described by Richter et al., who use a back projection table of “color classes” to classify wine berries[8]. First, joint RGB-histograms of all the materials that are expected to be encountered during sorting are collected from a sample. The histograms are post-processed to suppress outliers and then fused into color classes, which are then dilated to allow generalization to unseen colors. Finally, a mapping from color value to color class is

established using maximum-a-posteriori classification of each possible RGB tuple. Objects are classified according to the frequency of occurrence of the color classes.

All these approaches show good results in their application domains. However, as they focus exclusively on color features, broad application in different industrial settings is questionable. Especially in applications where texture, instead of color, is the most informative feature, the methods may fail. The approaches by Zhang et al. and Li et al. are further strictly tailored to their respective products, which may render them unusable with other products such as minerals.

Methods

Our approach is based on the bag of visual words (BOW) framework. BOW was motivated by the classification of text documents. The enabling insight is that the fundamental, meaningful elements of a document (such as this one) are words. Some of the words in a document carry more information than others. One can even get an idea about the content of the document when just given a list of words and how often they appear in the document. Csurka et al. proposed a similar approach to categorize images [9]. The key idea is to consider an image to be composed of visual words, where the vocabulary of visual words is determined from a set of basic features. Then, images can be classified by determining which of the visual words occur in the image and how often they do.

The BOW framework can be formalized as follows: Given a set of N training images $\mathcal{D} = \{I_n \mid n = 1..N\}$, T_n low level local descriptors $\mathbf{x}_{tn}$ are extracted from key points in each image. Key points are usually automatically determined using a key-point detector that is sensitive to edges and corners in the image. Therefore, the number T_n of low level descriptors can be different from image to image. The $\mathbf{x}_{tn}$ are then clustered using K-Means to obtain K cluster centers $\mathbf{c}_k$. These cluster centers form the visual vocabulary: each $\mathbf{c}_k$, or rather the vornoi region around it, corresponds to a visual word.

An object descriptor is built from an unseen image I by again extracting T low level descriptors $\mathbf{x}_t$ and building a count statistic over the closest visual words. More specifically, the descriptor $\mathbf{m} = (m_1, \ldots, m_K)^\top$ is built by hard assignment to the nearest cluster center,

$$m_k = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\left[\mathbf{c}_k = \operatorname*{argmin}_{\mathbf{c}} \parallel \mathbf{x}_t - \mathbf{c} \parallel\right].$$

Figure 1 visualizes this approach.

Application to bulk material sorting

Bulk material sorting places some restrictions on the BOW framework. First, the objects under inspection are typically very small and contain very few key points where local

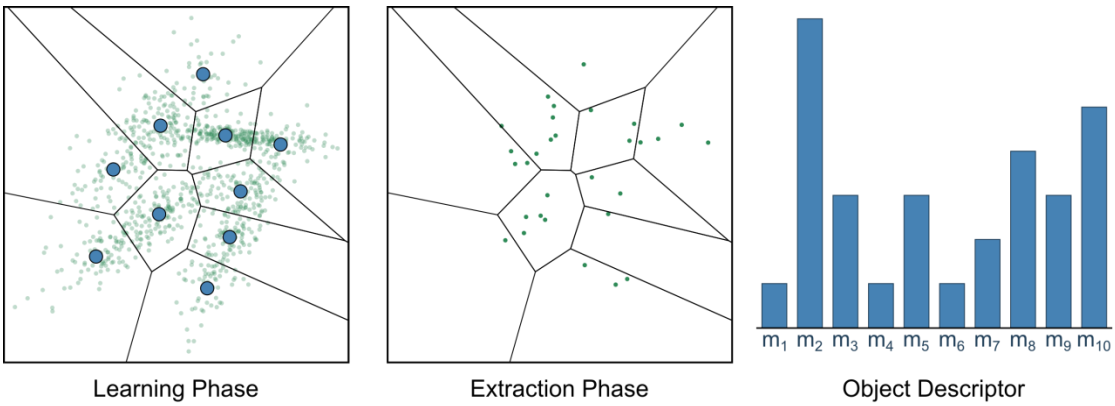


Figure 1: Overview of the BOW approach. Left: The vocabulary of visual words is determined by clustering a large number of low level descriptors. Center: For a given image, a descriptor is derived by observig the voronoi region in which the low level features fall into. Right: The resulting object descriptor is a count statistic over the visual words.

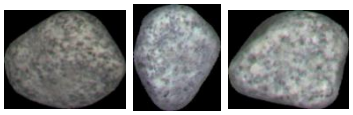
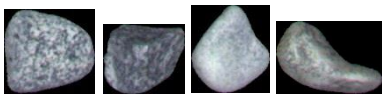
descriptors can be extracted. Second, bulk material sorters are real-time systems, which means the available time to process an object is strictly limted. On the other hand, the system setup is under full control, which means that the low level descriptors do not have to be robust against nuisances like lighting conditions and cluttered backgrounds.

Given these circumstances, we adapt the BOW framework to use *dense sampling* and *primitive descriptors*. With *dense sampling*, we consider each foreground pixel as a key point to extract the low level local descriptors. Since segmentation of fore- and background is a necessary step in any bulk material sorter, segmentation does not introduce additional computation time. However, dense sampling is only feasible, if the low level descriptors are inexpensive to compute. This gives rise to *primitive descriptors*: local feature descriptors that require little or no computation to obtain. In particular, we consider the following primitive features: color channels, gray image, gradient magnitude at different scales, rotation invariant uniform local binary pattern [10] and a distance transform image. Note, again, that these primitive features are extracted *from every object pixel*. In case of color features, this means that the features are “computed” by a simple image lookup. Also note that dense sampling and primitive descriptors are both required to make BOW usable in the context of bulk material sorting. Without the former, the descriptors do not provide enough discriminative information. Without the latter, the computational load would be too high to keep the real-time promises.

Experiments

To validate the suitability of our approach in the context of mineral sorting, we used a bulk material sorter to record images of two different kinds of pebble stones. The stones were chosen to be visually similar, so as to simulate a challenging sorting problem. In fact, it is not immediately obvious, which features should be used to classify the stones.

Table 1: Overview of the dataset used in our experiments.

Class	Description	# Samples	Example images
A	Large gray pebble stones with dark spots	1212	
B	Gray pebble stones with light spots	4671	

The imaging part of the sorting system consisted of a RGB line camera producing a resolution of 170 µm × 170 µm per pixel. The overall inspection width was 700 mm. Spatial binning with a factor of 2 was used to reduce the size of the images, giving an effective resolution of 340 µm × 340 µm per pixel. Halogen lamps were used as illumination.

We recorded the two different types of stones in two separate rounds. Objects were detected as if in regular operation and an image that contained only the object was written to disk for each object. This way, we obtained a total of 5883 images. Table 1 shows more details on the dataset. Note that it was not possible to fully clear the machine of leftover material from previous runs. The result is that class B contains a small fraction of objects from class A.

Implementation Details

We ran the experiments offline on a 16-core Intel Xeon CPU with 2,96GHz and 32GB of RAM. The method was implemented using the interpreted, JIT-compiled, garbage collected Julia language [11]. The learning phase utilized data-parallelism, but the evaluation phase was implemented as a single thread to allow comparable timing measurements.

We used three color channels, the gray image, gradient magnitude at four different scales, i.e. gradient images filtered with a Gaussian filter (σ=1,1.5,2,2.5), rotation invariant local binary patterns [10] and the distance transform as primitive features (see Figure 2). Thus, we obtain a 10-dimensional low level local feature descriptor. Because the color space has significant impact on the learned vocabulary (clustering implies distance measurement), we experimented with five different color spaces (RGB, HSL, Lab, Luv and XYZ).

Metric

As is common in automated visual inspection tasks, the dataset is noticeably imbalanced. As this imbalance affects often-used metric like the Accuracy and the F_1-score, we instead chose the symmetric Matthews Correlation Coefficient (MCC) [12] to measure classification performance. With n_{tp}, n_{tn}, n_{fp} and n_{fn} denoting the number of true positive, true negative, false positive and false negative classifications, MCC can be computed as

$$\mathrm{MCC} = \frac{n_{tp}n_{tn} - n_{fp}n_{fn}}{\sqrt{(n_{tp} + n_{fp})(n_{tp} + n_{fn})(n_{tn} + n_{fp})(n_{tn} + n_{fn})}}.$$

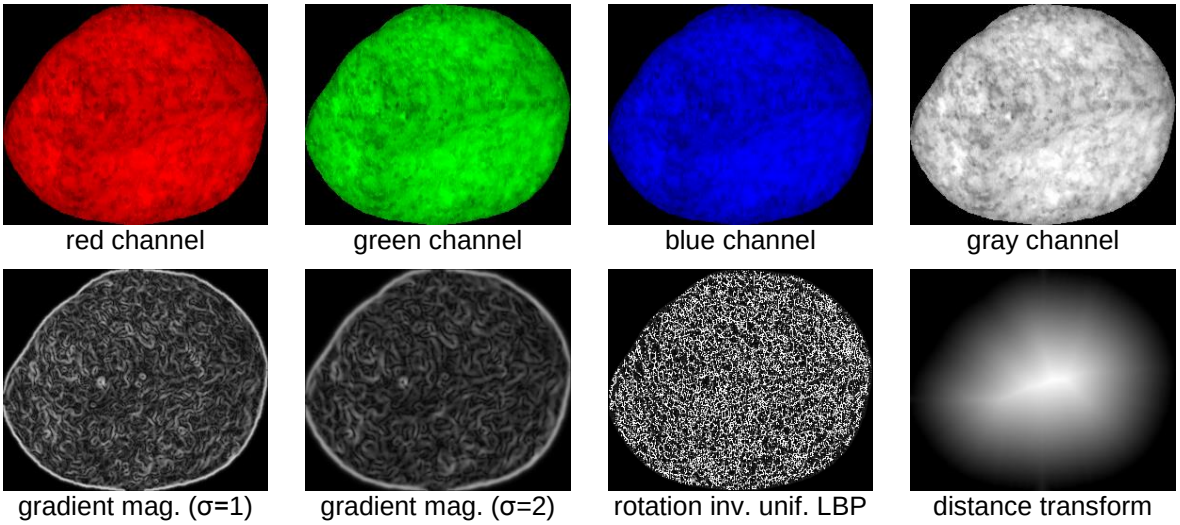


Figure 2: Examples of the primitive feature channels used in our experiments. Gradient magnitude of the scales σ=1.5 and σ=2.5 are not shown. Although the full image is shows, the primitive features are extracted from every object pixel. Note: the brightness and contrast of the images was increased for this paper.

MCC can be interpreted as the correlation between the classifier's prediction and the ground truth: a MCC near ± 1 indicates good classification, whereas $MCC = 0$ is equivalent to a random choice.

Results

Figure 3 shows the classification performance obtained in our experiments. With all configurations, nearly perfect classification can be reached. With the linear SVM, the impact of the color space is negligible, although the results are most stable when color is encoded in the Lab color space. With the decision tree, one can obtain a slightly better classification result than with the linear SVM. Here, visual words building on the Luv and Lab color spaces seem to outperform other color spaces, although only by a small margin.

With all configurations, features could be extracted in under 90ms per sample once the vocabulary was determined. The bulk of the processing time was spent in computing the distance transform image. Clearly, this is too slow for a real-time application. The available processing time in a bulk material sorter depends on the speed of the objects as well as the distance between inspection line and ejection speed, but in a typical system, the available time slot is in the order of 15 ms for *all* objects between inspection line and ejection stage. Note, however, that our implementation is prototypical and not optimized for speed. In particular, we implemented the method using the Julia language, which is interpreted and uses a garbage collector. We expect that significantly speedups can be achieved by implementing the method in a compiled language and offloading parts of the computation to specialized hardware, e.g. a framegrabber or a GPU. As the BOW descriptor is built by quantizing the feature space, lookup tables might prove especially useful.

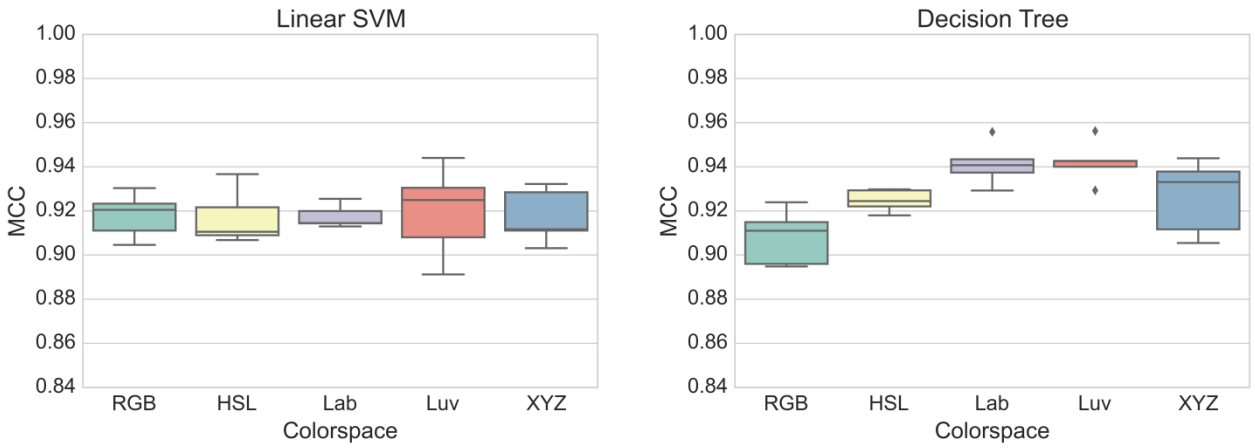


Figure 3: Classification performance with different classifiers and feature configurations.

Conclusion

In this paper, we have shown the application of the bag of visual words framework in the context of bulk mineral sorting. The approach encodes the content of images using a vocabulary of visual words that was learned in a previous step. In the context of visual inspection, the BOW framework is applied by introducing *dense sampling* and *primitive descriptors*. The method is fully automatic and can derive both object descriptors and classification rules from an annotated sample of the material to be sorted. This reduces human interaction to a minimum.

In this paper, we used color, texture (gradient magnitude, local binary patterns) and shape (distance transform) features to classify two kinds of visually similar pebble stones. With both a linear SVM and a decision tree classifier, we achieved nearly perfect classification rates. The method is very fast, but not yet fast enough for application in real-time systems. For the future, we plan to investigate methods to offload certain computations to specialized hardware such as a framegrabber or a GPU.

References

[1] Richter, M.; Längle, T.; Beyerer, J.: *Visual words for automated visual inspection of bulk materials*. In: *Proceedings of International Conference on Machine Vision Applications*. Tokyo, Japan, 2015, pp. 210–213, ISBN 978-4-901122-15-3.

[2] Schenker, T.; Negara, C.; Vieth, K.-U.; Gelo, S.: *Quality Improvement of Wine due to Hyperspectral Analysis*. In: *Proceedings of Sensor Based Sorting.* Aachen, Germany, 2014, pp. 117–126.

[3] Barni, M.; Cappellini, V.; Mecocci, A.: *Colour-based detection of defects on chicken meat*. In: *Image and Vision Computing.* 15, 7 (1997), pp.549–556.

[4] Duffy, N.; Crowley, J.; Lacey, G.: *Object detection using colour*. In: *15th International Conference on Pattern Recognition.* 2000, pp. 700–703.

[5] Bergasa, L.; Duffy, N.; Lacey, G.; Mazo, M.: *Industrial inspection using Gaussian functions in a colour space.* In: *Image and Vision Computing.* 18, 12 (2000), pp. 951–957.

[6] Zhang, D.; Lee, D.-J.; Tippetts, B. J.; Lillywhite, K. D.: *Date maturity and quality evaluation using color distribution analysis and back projection*. In: *Journal of Food Engineering.* 131 (2014), pp. 161–169.

[7] Li, C.; Cao, Q.; & Guo, F.: *A method for color classification of fruits based on machine vision*. In: *WSEAS Transactions on Systems*, 8, 2 (2009), pp. 312–321.

[8] Richter, M.; Längle, T.; Beyerer, J.: *An approach to color-based sorting of bulk materials with automated estimation of system parameters*. In: *tm-Technisches Messen,* 82, 3 (2015), pp. 135–144.

[9] Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C.: *Visual categorization with bags of keypoints*. In *International Workshop on Statistical Learning in Computer Vision (ECCV).* 2004, pp. 1–22.

[10] Ojala, T.; Pietikainen, M.; Maenpaa, T.: *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 24, 7 (2002), pp. 971–987.

[11] Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B.:*Julia: A fresh approach to numerical computin*g. In: *arXiv preprint arXiv:1411.1607* (2014).

[12] Matthews, B. W.: *Comparison of the predicted and observed secondary structure of T4 phage lysozyme*. In: *Biochimica et Biophysica Acta*. 405, 2 (1975), pp. 442–451.