Instant Segmentation and Feature Extraction for Recognition of Simple Objects on Mobile Phones

Andreas Hartl, Clemens Arth, Dieter Schmalstieg
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{hartl,arth,schmalstieg}@icg.tugraz.at

Abstract

Object detection and recognition algorithms are an integral part of the architecture of many modern image processing systems employing Computer Vision (CV) techniques. In this paper we describe our work in the area of segmentation and recognition of simple objects in mobile phone imagery. Given an image of several objects on a structured background, we show how these objects can be segmented efficiently and how features can be extracted efficiently for further object recognition and classification. We prove the algorithms presented are useful given a set of test cases, and we show that the algorithms discussed can be used for instant object segmentation and recognition in a real-world application on ordinary off-the-shelf smartphones.

1. Introduction

Computer vision (CV) based object detection and recognition has become an integral part of many modern image processing systems. Especially in the area of CV for industrial applications, many of these systems are based on algorithms that put heavy requirements on the hardware used and the image capturing conditions. This observation is still valid, even if the objects to be detected or recognized are of relatively simple shape, *e.g.* screws on a conveyor belt, simple workparts for manufacturing, medical pills, etc.

As a result of the complexity of the task and the need for significant amounts of computational resources, the use of CV based detection and recognition systems on mobile phones has not been widely investigated yet. Nevertheless, the deployment of such algorithms in mobile scenarios is still an important task, and the steadily increasing power of mobile devices opens up a huge field of applications for these types of algorithms in mobile phone scenarios. There is a strong motivation to push forward the development of algorithms suitable for performing these tasks on common off-the-shelf mobile phones.

In this work we discuss an approach to efficiently use object detection and recognition algorithms together on mobile phones. We present a way for efficient detection and segmentation of simple objects on a structured background. Based on the result of segmentation, a set of features is estimated which can be used for object classification, either on site with an on-board database, or by using an online database on the Internet. This setup allows for the generation of powerful mobile applications.

The rest of the paper is structured as follows. In Section 2 we give a short overview of CV on mobile phones. In Section 3 we discuss our approach for object segmentation on a structured background. Section 4 contains a closer description of the feature extraction and classification approach, followed by an extensive experimental evaluation in Section 5. Final remarks and an outlook on future work is finally given in 6.

2. Related Work

Since mobile phones have only recently become computationally powerful devices, the use of CV algorithms on mobile phones does not have a long history.

There is, however, some related work from a number of different research areas. Also, since mobile phone hardware is essentially embedded hardware, many algorithms have been developed in the embedded CV community[1].

Face detection for example is a popular research topic. Some camera phones have this feature nowadays, since most ordinary cameras have face detection algorithms already built in hardware. Theocharides *et al.* [18] propose the hardware implementation of a face recognition system based on neural network algorithm originally developed by Rowley *et al.* [14]. Rahman *et al.* presented an algorithm running in software on an *TI OMAP3430* platform [13]. The algorithm performs color classification using Gaussian mix-

[1] Note that in the following we will not discuss approaches which offload the CV work to a server, but only refer to related work performing the CV task on-site right on the mobile device.

ture models and basically detects skin-like areas as faces. The authors also present a hybrid system combining their original approach with the very popular Viola-and-Jones algorithm [12]. Ng *et al*. describes a system for face verification on mobile phones based on minimum average correlation filters [10]. A study on iris detection and verification is conducted by Park *et al*. [11].

In the field of Augmented Reality (AR) on mobile phones, Wagner *et al*. present and evaluate natural features for object tracking in real-time on mobile phones [20]. Klein and Murray recently presented a version of their PTAM software on an iPhone™ for small workspace self-localization and mapping (SLAM) [8]. For the task of outdoor AR, Takacs *et al*. suggest an approach using features subdivided into cells called *Loxels* [17]. An approach for self-localization in large-scale environments was presented recently by Arth *et al*. using point cloud reconstructions of city areas [1]. A system for guiding visually impaired people in traffic situations was presented recently by Ivanchenko *et al*. [7]. The system is running on a *Nokia N95* mobile phone, detects two-striped crosswalks and guides the user across streets.

The collectivity of related work mentioned above is not exhaustive. Other approaches include OCR [9, 22] or museum guidance [5] for example. However, it is apparent that CV has started to conquer the field of mobile phone development in recent years. Although there are applications for the most obvious tasks, to the best of our knowledge there is no basic study in the area of object segmentation and recognition on mobile phones available yet, which is comparable to our work presented in this paper.

3. Object Segmentation

When using mobile phones as image capturing devices, a non-fixed acquisition setup is typically the case. Segmentation methods, thus, should be invariant to changes in object position, orientation and scale. Naturally, perspective distortion, varying lighting conditions and blurred input images pose additional problems. Thus the goal is to develop an approach which is able to work despite adverse environmental conditions, and deliver results reliably and instantly on a mobile phone.

For solving the task successfully and instantly on mobile phones, we propose a method for robust segmentation based on a marker target with a structured background. A checkerboard background is chosen to allow segmentation of arbitrarily colored objects in a robust manner (see Figure 1). The marker target itself can be detected efficiently using readily available marker tracking software, like *Studierstube* or *ARToolkitPlus*[2]. Moreover, the marker itself has a fixed known size, which enables us to accurately measure

[2]http://studierstube.icg.tu-graz.ac.at

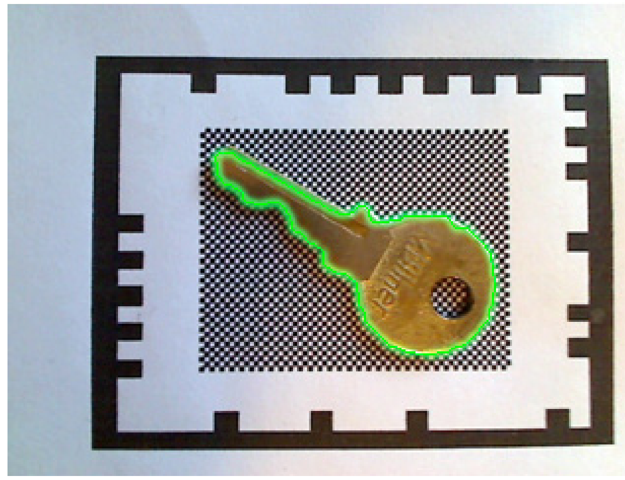

Figure 1. A frame marker with a generic object. The detected object contour is marked with green color.

the object dimensions. During the marker detection step, a homography H is calculated which essentially maps the marker in the image to the undistorted view of the marker (a plane-to-plane mapping [6]). This homography H is used to define a rectangular region of interest (ROI) within the original image to reduce the amount of data for segmentation and to facilitate parameter selection. Note that we do not rectify the image prior to segmentation. The final results are acquired by local adaptive thresholding and applying a set of morphological operations on a grayscale input image:

$$M_{seg} = (\neg(M_{Th_1} \bullet SE_1) \circ SE_2) \bullet SE_1, \qquad (1)$$

In Equation 1, M_{Th_1} denotes a mask obtained by local adaptive thresholding with a neighborhood size of $(2 \cdot w_h) + 1$, and SE_1 and SE_2 denote square-shaped and disk-shaped structuring elements of length $(2 \cdot w_h) + 1$ and $(2 \cdot w_h) + 3$, respectively. The symbol $\neg$ is used for mathematical inversion and the symbols $\circ$ and $\bullet$ denote morphological opening and closing. For local adaptive thresholding we use an efficient method originally proposed by Shafait *et al*. [15]. To extract contours, region labeling with integrated boundary computation is carried out following the linear-time method proposed by Chang *et al*. [3].

The choice of the single parameter w_h has turned out to be noncritical. It may be automatically selected by using prior knowledge about the image position of the checkerboard area and its amount of squares in each direction, if necessary. However, the length of the checkerboard pattern must be chosen with care, since it influences the quality of segmentation to a large extent. In turn, this choice is dependent on the desired image resolution, because a sufficiently sharp image of the background pattern is required[3].

[3]As a good tradeoff between different image resolutions, segmentation accuracy, runtime and usability, setting sq_w to $0.6mm$ is suggested.

Needless to say that the approach only works under the assumption that objects do not contain a similar pattern to the chosen background.

We have found the proposed method to be sufficiently invariant to perspective distortion and variable lighting. The algorithm relies on a series of relatively simple processing steps and the amount of input data can be considerably reduced. These properties make the approach particularly interesting for non-fixed setups using mobile devices with limited resources. Although the use of a marker with checkerboard background seems to be prohibitive, there are multiple justifications for this. For measuring object size, some reference measurements are necessary which can simply be provided by knowledge about the real marker dimensions. Preceding tests have shown that even popular and powerful segmentation algorithms like Graph Cuts or MSER are not able to deliver satisfying segmentation results for arbitrarily colored objects on, for example, uniform white background. Moreover, these algorithms can hardly be used as efficiently as our approach in terms of computational and memory requirements.

4. Feature Extraction and Classification

For object classification, several features and properties have to be estimated from the segmented object. Typical features include *object size*, *color* and *shape*, but also material related properties like *surface glossiness* or *object transparency*. However, the latter properties are hard to estimate accurately, so the former ones must serve sufficiently for reliable classification. For calculation of each feature, efficient solutions are needed to allow instant computation on mobile devices.

4.1. Object Size

Correct measurements of the object size can only be acquired if invariance to perspective distortion is ensured. This can be facilitated by rectification of the object (or its boundary) according to the previously calculated homography H. Still, some assumptions have to be made according to the specific object category. While it is relatively simple to calculate the length and width of circular or rectangular objects, a more sophisticated method is needed for taking such measurements on more general objects.

First, we determine the direction of maximum variance within the segmented region. Because the boundary can be distorted due to errors during segmentation, a subset of all region points is taken for increased robustness. This subset of points is selected and rectified using the homography H. Subsequently, the direction of maximum variance $\mathbf{v_1}$ $= [v_1; v_2]$ is determined by analysis of the covariance matrix $\mathbf{C}$ [21]. Generally, we define the length of an object as the extension of its boundary along the major axis of its

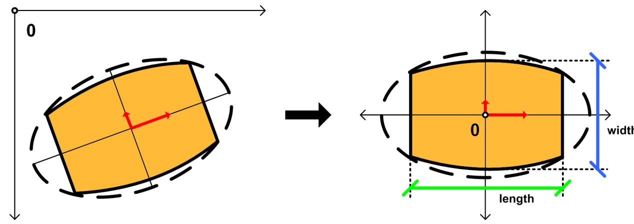

Figure 2. Variance estimation for a generic object (beige color) is shown on the left, the general definition of length and width is depicted on the right side.

projection. The extension in the perpendicular direction is then defined as width. Both measurements are acquired by recording the minimum and maximum values of the corresponding projection. As mentioned above, the definition of length and width might vary according to the given object category (see Figure 2 for an illustration of the measurement process).

The square size sq_w of the checkerboard pattern, that limits the accuracy of segmentation, also limits the accuracy of the measurements. Under the assumption that the target entirely fills the viewable area and does not exhibit other kinds of distortion, this is basically the only factor influencing the expected error. If we assume that objects may be correctly segmented under the previous conditions, the maximum error e_{max} can be estimated to be:

$$e_{max} = 2 \cdot sq_w \tag{2}$$

Using an ordinary ruler as a measurement tool, it is rather difficult to measure lengths below 1 mm, and arbitrary object shapes further aggravate measurements. For these reasons, the accuracy of the proposed method can be expected to be better than what can be obtained with a ruler.

The most expensive step in the proposed method for size estimation is the determination of the direction of the major axis. By using a suitable spacing for sampling region points, which in turn is dependent on the current region size, the required effort can be considerably reduced with little degradation in accuracy. So, the problem can still be computed in a reasonable amount of time for larger region sizes. Due to the low amount of memory resources needed and the possibility to do all calculations using fixed-point arithmetics, if necessary, the algorithm is ideally suitable for mobile devices.

4.2. Object Color

Methods for object color estimation have to be tolerant to varying lighting conditions, and the results should correspond to human perception as closely as possible. In our approach, as a first step we reduce the influence of varying lighting conditions by applying a method for local white balance. This method is based on reference measurements on the white border of the marker-based target and applying a subsequent correction to each segmented pixel accordingly. For information on the algorithm used, the interested reader is referred to the work of Süsstrunk *et al*. [16].

Depending on the given task, usually a restricted discrete subset of colors is assumed to cover the entire range of possible object colors. Facilitating this assumption and for reasons of efficiency, we perform color estimation using an sRGB lookup table containing the set of possible object colors. This lookup table is generated by defining a set of suitable colors manually, and partitioning the color space with heuristic thresholds accordingly (including a class for all colors not covered by any manually chosen one). Alternatively, suitable color classes can be generated by calculation from a set of training samples in an offline stage. For assigning a discrete color to each pixel, we evaluate the ΔE_{CIE00} color distance metric in CIE LAB space (for details refer to the work of Vik [19]). Through assigning each pixel the most likely color, we get a per-pixel classification result for a segmented object.

The classification results are aggregated in a histogram, and the final color estimation result is drawn from subsequent analysis of this histogram information. Currently, we assume objects to be of one single or two dominant colors, and do not expose color gradients. We sort the histogram h_c in descending order and apply the four measurements:

single color (i) coverage: $$c_i = \frac{h_c(0)}{\sum h_c} \quad (3)$$

two color (i, j) coverage: $$c_{i,j} = \frac{(h_c(0) + h_c(1))}{\sum h_c} \quad (4)$$

single color (i) significance: $$s_i = \frac{h_c(0) - h_c(1)}{h_c(0)} \quad (5)$$

two color (i, j) significance: $$s_{i,j} = \frac{1}{s_i} \quad (6)$$

An object is assumed to be of a specific color, if one of the coverages is above a threshold th_c. The products

$$p_i = c_i \cdot s_i \quad (7)$$

$$p_{i,j} = c_{i,j} \cdot s_{i,j} \quad (8)$$

are used to decide, whether one or two dominant colors are present. If $p_i > p_{i,j}$, the result is the corresponding label i of the maximum entry in h_c. Otherwise two colors are assumed and the labels i, j corresponding to the first two entries in h_c are reported.

The proposed method is particularly suited for non-fixed setups, because it is able to reduce the influence of ambient lighting and the results correspond to human perception of color. The approach is rather efficient and thus instantly computable on mobile phones, since costly color conversions and distance computations are avoided. The size of the lookup table (s_{LT} entries per channel) can be chosen depending on the memory resources available, thus the approach scales well for different hardware setups. Note that color classification for more than two colors can easily be performed using a more evolved histogram analysis.

4.3. Object Shape

The boundary of the segmented region can serve as a basis for the estimation of object shape. As for determining object size, shape estimation should be invariant to changes in translation, rotation, scale, and should also tolerate a certain degree of perspective distortion. In this respect, we propose a modified pairwise geometric histogram (PGH), originally proposed by Evans *et al*. [4] for this task. The PGH descriptor features good descriptive power and also allows for efficient shape matching.

In the PGH descriptor, oriented line segments are investigated. Their relative orientation and perpendicular distance is analyzed and this information is collected in a 2D histogram of size $D \cdot A$. The set of possible angles and distances is mapped onto the histogram by accumulation of occurrences. During the process, each line is used as a reference line and the angle, as well as the perpendicular distance, is computed to all the other lines. Every line is represented as a histogram and the accumulation of all histograms represents the shape of the object (see Figure 3). The boundary of a segmented object has to be approximated by a polygonal representation. An efficient implementation of *Critical Point Detection* (CPD) proposed by Zhu *et al.* [23] can be used therefor.

The PGH is not scale invariant in its original form. However, limited scale invariance may be achieved by application of a suitable and stable similarity metric [2]. For efficiency reasons we assume that the maximum distance for a PGH can be estimated from the rectified contour. Thus, the scale is estimated by searching for the maximum distance δ_{max} of the boundary from its centroid, and the maximum PGH distance is computed as:

$$\delta_{PGH} = 2 \cdot \delta_{max} \quad (9)$$

Shape matching is finally done by using the Euclidean distance as a similarity metric. Note that for simple convex objects the PGH can be constructed solely by the points of the object's convex hull (no CPD required).

The proposed method for shape estimation can be instantly computed on mobile devices, because the number of input points needed for PGH computation can be scaled on demand. For simple convex objects, information already available about the convex hull can be used to avoid sampling critical boundary points. For concave objects, the use of the CPD only introduces negligible computation overhead. Shape matching using the proposed similarity metric can be further optimized by using suitable data structures.

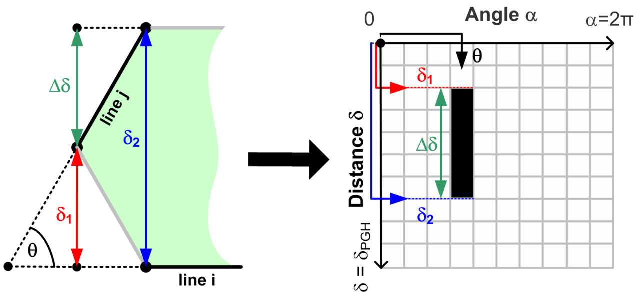

Figure 3. Line j is described in the pairwise geometric histogram with reference to line i. The collectivity of all descriptions forms the final shape descriptor of the object.

		Objects			
		Coins	Med. Pills	Screws/Nuts	Misc
Number		8	13	13	6
Convex		true	true	false	false
Image Size	**320x240**	1	13	0	0
	422x318	8	13	6	5
	506x380	8	13	10	5
	576x432	8	13	11	6
	640x480	8	13	13	6

Table 1. The five different object categories used in our experimental evaluation. The lower part of the table lists the number of successfully segmented objects for varying image resolutions. Note that we varied the image resolution linearly in the number of image pixels.

This makes the algorithm ideally suited to mobile devices due to its low computational complexity and the moderate memory requirements.

5. Experiments

In the following, we evaluate the individual parts of our approach in detail on different object categories. Finally we demonstrate the application of our approach on the task of medical pill recognition on a modern phone. All parts of our algorithm were implemented in C/C++. As our mobile platform we choose an Asus M530w smartphone[4] running Windows Mobile 6. The device features a 2 mega-pixel autofocus camera, a 416MHz fixed-point CPU and 64 MB of RAM, as well as 256 MB Flash memory. At the time of writing, this phones hardware is already a little outdated, so our approach is expected to run at a higher performance on future smartphones.

5.1. Object Segmentation

First, we evaluate our segmentation method for varying image resolutions. Table 1 lists the details about the different sets of object categories used. The categories are sorted by increasing complexity, from relatively simple objects such as coins to arbitrary objects like keys. We generated ground-truth data by segmenting the object in the test images manually.

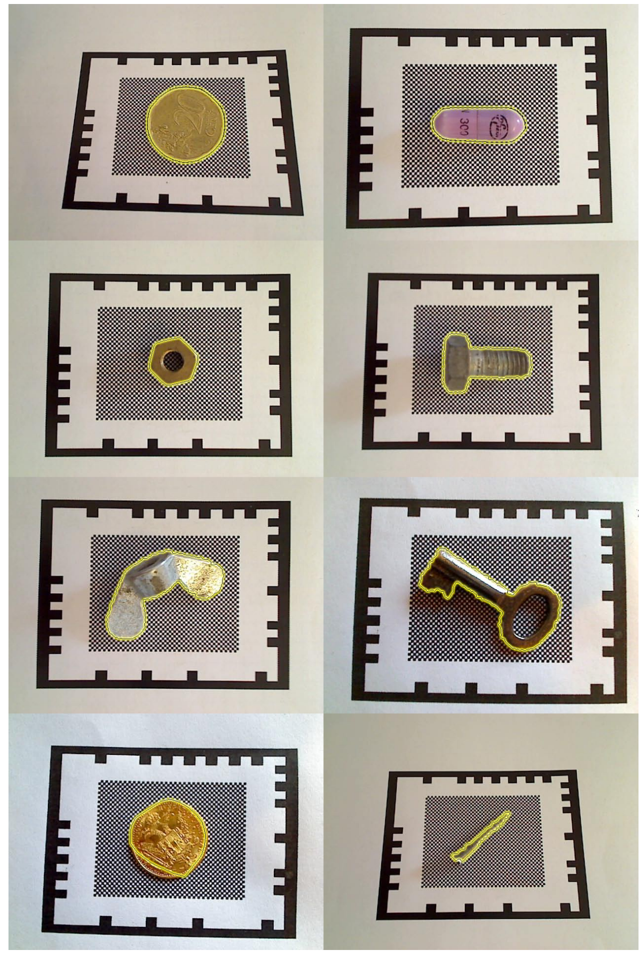
Figure 4. Sample results from the object segmentation step. The two images in the last row show erroneous and inaccurate segmentation results.

Figure 4 shows some sample segmentation results. Quite accurate segmentation results are generated in most cases, but as can be seen from the last two images, in some cases reflections or bad lighting effects can cause inaccurate or erroneous results. The lower part of Table 1 gives an overview about the number of successfully segmented objects in the individual categories for varying image resolutions. For a considerable number of objects, the algorithm is not able to generate a reasonable segmentation result at lower image resolutions. This could be improved by using different marker-based targets that match specifically to the current resolution (modified square length sq_w).

Figure 5 depicts the segmentation accuracy for varying image resolutions and the runtime performance on the mobile phone. As expected, the segmentation accuracy increases with increasing image resolution, but saturates at

[4] http://www.asus.com

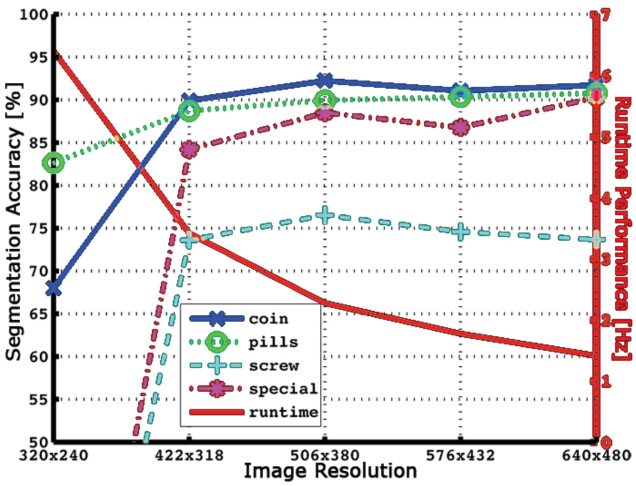

Figure 5. Segmentation accuracy and runtime for varying image resolutions on the mobile device. *Accuracy* is measured as relative difference between the segmentation result and the ground truth, compared to the total object size. The *runtime performance* is averaged over all segmentation runs for all object categories in the test set.

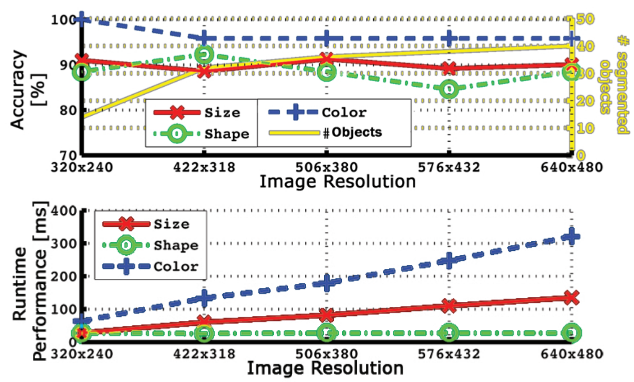

Figure 6. Accuracy and runtime performance for varying image resolutions on the mobile device. In the upper graph, the number of objects is also plotted, which is delivered to the feature extraction algorithm from the segmentation step. Accuracy is measured in terms of difference in object width/height measured, compared to the ground truth object size. For shape and color, accuracy is measured in terms of percentage of correct votes for shape and color categories, compared to the overall number of objects.

a medium resolution level. Also the time spent in the segmentation step on the mobile device increases accordingly, so the fictional segmentation frame rate drops from about 6 *fps* for the lowest resolution to about 1.4 *fps* for the highest tested resolution.

5.2. Feature Extraction

In the second experiment, we investigate the feature extraction algorithms on our mobile device for varying image resolutions. Figure 6 shows the accuracy and the runtime

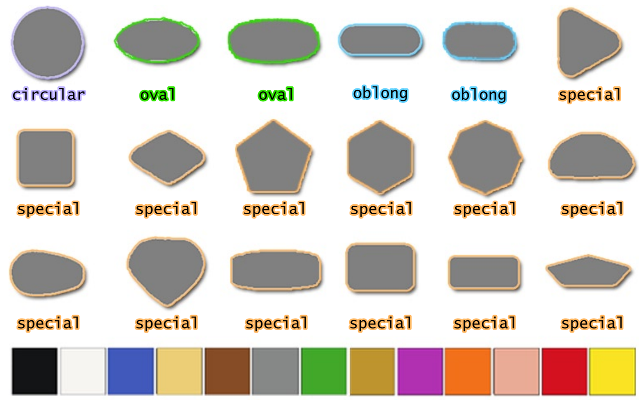

Figure 7. Four different shape classes with differently colored boundaries and 13 color classes used in our experiments.

performance plotted against varying image resolutions. Additionally, in the upper plot the number of objects delivered by the previous segmentation step is given for each individual resolution. As can be seen, the accuracy for the individual features stays approximately the same for all tested resolutions. As an important aspect, for low resolutions the number of objects passing the previous segmentation stage is significantly lower than for images with high resolution. This result directly implies that the limiting factor in our approach is basically the segmentation step. Segmentation does not work successfully on most objects for low image resolutions, although the subsequent feature extraction would lead to accurate results. Thus, an improvement to the previous segmentation algorithm is advised. In the lower plot, it's easy to see that the runtime for the *size* and *color* estimation algorithm increases linearly with increasing image resolution, while the *shape* estimation runtime stays constant. The reason for this is that we chose to linearly increase the number of sample pixels used for the estimation of the former features. In contrast, the number of investigated line segments for the *shape* feature stays almost constant.

5.3. Application to Medical Pill Recognition

Finally we propose the use of our segmentation and feature extraction approach in the task of medical pill recognition on mobile phones. Often, the process of drug identification must be carried out in a mobile scenario, where for example a person is found unconscious. In the absence of other tools, only a visual inspection is possible. In this application scenario the proposed segmentation and feature estimation methods are used to estimate the corresponding properties of a pharmaceutical pill and to query a database.

We have generated a reference set of medical pills for this experimental evaluation, where we have restricted ourselves to use four different shape classes and 13 color classes, as illustrated in Figure 7. Single pills were allowed

Shapes	*circular*	*oval*	*oblong*	*special*
Instances	41	26	33	8
Colors	*single*	*multi*		
Instances	98	10		
Sizes [mm]	*min. length*	*min width*	*max length*	*max width*
	5.68	5.68	18.07	18.07

Table 2. Reference testset of medical pills for our experimental evaluation. The testset exhibits reasonable variation in shape and pill size.

Color	*black*	*white*	*blue*	*beige*	*brown*
Instances	1	29	6	11	4
Color	*gray*	*green*	*ocher*	*pink/violet*	*orange*
Instances	1	5	6	3	6
Color	*rose/peach*	*red*	*yellow*		
Instances	11	5	10		

Table 3. Distribution of single-colored pills over the 13 color classes used.

to be of one or two dominant colors. Note that collecting a large amount of different medical pills is a tedious task, especially when the aim is to collect samples from a larger number of different color and shape classes. Because pills with rarely used shapes are especially hard to get, we collected them in a large class for special shapes. In Table 2 details about the testset are given, Table 3 describes the distribution over the 13 color classes for single-colored pills.

For the experiment we fixed the image resolution to 640x480 pixels. All samples were captured under two different lighting conditions, fluorescent light and daylight. From initial evaluation of this data, optimal parameters for shape estimation ($D = 12, A = 12$) and color estimation ($s_{LT} = 36$) were determined. In this step, the best recognition rate for shape is 89% and that for color is 84%. In size estimation an average deviation of 0.43 mm and a maximum deviation of 1.37 mm is achieved.

We retrieved pill information from the reference set described in Tables 2 and 3 by querying, either using single features or simple AND-combinations of several features. Querying for object size, results in a range of ± 0.7 mm from the measured value were allowed. From the final list of possible candidates, we counted a reported pill as successfully recognized if it is among the first N candidates.

Retrieval performance for single features gives information about the descriptive power of a feature (see Figure 8). *Size* is determined to be most powerful feature, followed by *color* and *shape*. *Size* and *color* give better results with less candidates than the feature *shape*. Performance with the feature *size* reaches a recognition rate of 84.26 % ($N = 7$ candidates). This rate can be significantly improved, if the results are initially ordered by the minimum sum of deviations in length and width (indicated as *size-sorted* result in Table 4).

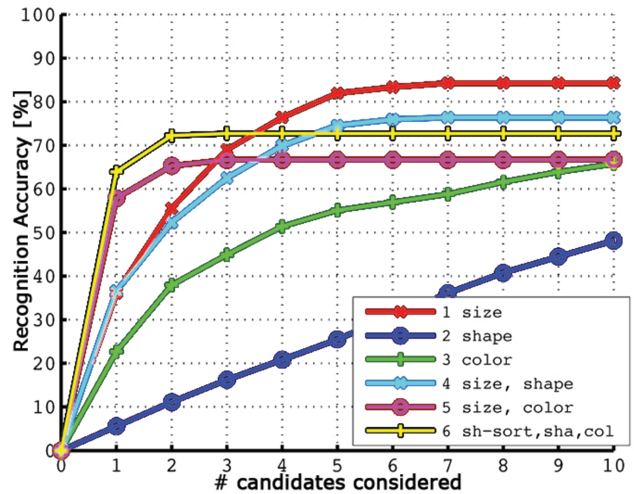

Figure 8. Pill retrieval on the reference set.

Size	Size (sorted)	Shape	Color	RR_{max}	#
X				84.26 %	7
	X			**100.00 %**	8
		X		48.15 %	10
			X	65.74 %	10
X		X		**76.39 %**	7
X			X	66.70 %	3
X		X	X	59.72 %	3
	X	X		**90.28 %**	6
	X		X	81.48 %	3
	X	X	X	72.69 %	3

Table 4. Pill retrieval on the reference set with single features and feature combinations ($N_{max} = 10$). RR_{max} denotes the highest recognition rate achieved.

Peak performance drops when using feature combinations. The reason can be seen in mutual reactions of errors from the individual feature estimators and our use of a strict AND combination. Although this way of combining features seems to be a disadvantageous choice at first, this is what online available databases are most likely to offer (see for example the Identa[5] online database). Without doubt, using a probabilistic combination of features accounting for estimation uncertainty of individual features, the recognition performance could be increased significantly. When using *size-sorted* results, considerable gains in recognition rate are possible. With subsequent querying for color matches, a recognition rate of 90.28 % ($N = 6$ candidates) is achieved. The peak performance for several feature combinations is listed in Table 4.

6. Conclusion

In this paper we presented our approach to instant object segmentation and recognition of simple objects on modern

[5] http://www.gelbe-liste.de/pharmindex/identa

mobile phones. After describing the basic principles of our algorithms in detail, the applicability and usefulness of our approach was demonstrated on the real-world application of medical pill recognition. The results of our extensive evaluation prove the performance and accuracy of our approach, and its suitability for several tasks. This was demonstrated on the application of pill recognition on mobile phones. The slim design of our algorithms was shown to be ideally suited to modern mobile phone architectures despite existing hardware limitations.

Although a lot of effort is put into generating an accurate and fast object segmentation, under several circumstances the results were still not satisfying. Difficult lighting conditions can generate artifacts or highlights on objects, which cause segmentation errors or, in the worst case, algorithm failure. Thus, an important part of future work must be dedicated to further improve the quality of the segmentation result. Another interesting topic of upcoming research will be the estimation of additional object features. For known categories, additional properties can probably be estimated, *e.g*. object transparency or surface glossiness, to further draw conclusions on the object material from. Needless to mention that further optimization of our algorithms to reach real-time performance on modern mobile phones is left as an open issue.

Acknowledgements

This research work was partially funded by the Christian Doppler Research Association.

References

[1] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide Area Localization on Mobile Phones. In *ISMAR*, pages 73–82, Oct. 2009. 2

[2] A. Ashbrook, N. Thacker, P. Rockett, and C. Brown. Robust Recognition of Scaled Shapes using Pairwise Geometric Histograms. In *BMVC*, pages 503–512, 1995. 4

[3] F. Chang, C.-J. Chen, and C.-J. Lu. A Linear-Time Component-Labeling Algorithm using Contour Tracing Technique. *CVIU*, 93(2):206–220, 2004. 2

[4] A. Evans, N. Thacker, and J. Mayhew. Pairwise Represenations of Shape. In *ICPR*, pages 133–136. Los Alamitos, Calif. u.a. IEEE Computer Soc. Pr., 1992. 4

[5] P. Föckler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber. PhoneGuide: Museum Guidance supported by On-device Object Recognition on Mobile Phones. In *Proc. Int. Conf. on Mobile and Ubiquitous Multimedia (MUM)*, pages 3–10, 2005. 2

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2008. 2

[7] V. Ivanchenko, J. Coughlan, and H. Shen. Staying in the Crosswalk: A System for Guiding Visually Impaired Pedestrians at Traffic Intersections. *Assoc. for the Advancement of Assistive Technology in Europe (AAATE)*, Sep. 2009. 2

[8] G. Klein and D. Murray. Parallel Tracking and Mapping on a Camera Phone. In *ISMAR*, pages 83–86, Oct. 2009. 2

[9] M. Koga, R. Mine, T. Kameyama, T. Takahashi, M. Yamazaki, and T. Yamaguchi. Camera-based Kanji OCR for Mobile-phones: Practical Issues. In *ICDAR*, volume 2, pages 635 – 639, Aug. 2005. 2

[10] C. Ng, M. Savvides, and P. Khosla. Real-time Face Verification System on a Cell-phone using Advanced Correlation Filters. In *IEEE Workshop on Automatic Identification Advanced Technologies*, pages 57 – 62, Oct. 2005. 2

[11] K. R. Park, H.-A. Park, B. J. Kang, E. C. Lee, and D. S. Jeong. A Study on Iris Localization and Recognition on Mobile Phones. *EURASIP Journal on Advances in Signal Processing*, 2008. 2

[12] M. Rahman, N. Kehtarnavaz, and J. Ren. A Hybrid Face Detection Approach for Real-time Deployment on Mobile Devices. In *ICIP*, pages 3233 –3236, Nov. 2009. 2

[13] M. Rahman, J. Ren, and N. Kehtarnavaz. Real-time Implementation of Robust Face Detection on Mobile Platforms. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 0:1353–1356, 2009. 1

[14] H. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-based Face Detection. In *CVPR*, pages 38–44, Jun. 1998. 1

[15] F. Shafait, D. Keysers, and T. Breuel. Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images. In *Proc. Int. Conf. on Document Recognition and Retrieval (DRR)*, volume 6815. SPIE, 1 2008. 2

[16] S. Süsstrunk, J. Holm, and G. D. Finlayson. Chromatic Adaptation Performance of Different RGB Sensors. *Proc. of SPIE, the Int. Society for Optical Engineering*, 4300:172–183, 2001. 4

[17] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod. Outdoors Augmented Reality on Mobile Phone using Loxel-based Visual Feature Organization. In *Proc. ACM Int. Conf. on Multimedia Information Retrieval (MIR)*, pages 427–434, 2008. 2

[18] T. Theocharides, G. Link, N. Vijaykrishnan, M. Irwin, and W. Wolf. Embedded Hardware Face Detection. In *Proc. Int. Conference on VLSI Design*, pages 133 – 138, 2004. 1

[19] M. Vik. Industrial Colour Difference Evaluation: LCAM Textile Data. In *Proceedings of the AIC Color and Paints (Interim Meeting of the International Color Association)*, pages 138–142, 2004. 4

[20] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose Tracking from Natural Features on Mobile Phones. In *ISMAR*, pages 125–134, Sep. 2008. 2

[21] S. Wijewickrema and A. P. Paplinski. Principal Component Analysis for the Approximation of a Fruit as an Ellipse, 2004. 3

[22] J. Zhang, X. Chen, J. Yang, and A. Waibel. A PDA-based Sign Translator. In *Proc. IEEE Int. Conf. on Multimodal Interfaces (ICMI)*, pages 217 –222, 2002. 2

[23] P. Zhu and P. M. Chirlian. On Critical Point Detection of Digital Shapes. *PAMI*, 17(8):737–748, 1995. 4