Matlab Implementation of Coin Segmentation Algorithm

Christian Kotz

April 25, 2008

1 Description

The algorithm implemented is based on [3], although it has been altered to improve runtime performance:

[3] uses a Generalized Hough Transform with a three dimensional voting space to search for the two coordinates of the coin center and its radius r. Let us call the gray-valued image function $I : \mathbb{R}^2 \mapsto \mathbb{R}$ and the voting function $P : \mathbb{R}^3 \mapsto \mathbb{R}$. The idea is to search for the maximum of the integral:

$$P(\mathbf{x}) = \int_{\mathbb{R}^{\neq}} (\delta(\mathbf{x} + rN(\mathbf{y})) + \delta(\mathbf{x} - rN(\mathbf{y})))) W(\mathbf{y})) d\mathbf{y} \tag{1}$$

Here $N : \mathbb{R}^2 \mapsto \mathbb{R} = \nabla I / \|\nabla I\|$ is the normalized gradient of the image. The integral weights the accumulations with the magnitude of the gradient $W : \mathbb{R}^2 \mapsto \mathbb{R} = \|\nabla I\|$. δ is the indicator function — an infinite impulse response lowpass filter in Reisert's implementation.

For runtime efficiency Reisert searches for the maximum by hierarchical voting where the radius r is varied. The interval of possible radius length is divided into four intervals and the voting function P is calculated at the midpoint of each interval. The maximum is searched over all of these calculations and the process is then recursively repeated for that interval of the four, which contains the radius parameter to the maximum found. The process stops after a predefined recursion level has been reached.

The proposed algorithm differs from Reisert in three aspects:

1. The size of the accumulator is varied as hierarchical voting is performed. This has two advantages.

   (a) While the radius is only a poor approximation, it is easier to find the maximum in a small accumulator and because by means of the downsampling implicit smoothing is done to the accumulator.

   (b) At the same time processing time for the search is reduced.

   As the approximations to the solution improve in lower levels of the hierarchy, larger accumulators are used to get the same precision as the original algorithm, up to an accumulator size of the full size of the image. It is likely that the smoothing of the accumulator can entirely be avoided even in these levels, because the

Hough centers get more and more focused the further down we proceed in the hierarchy.

2. A further improvement has been introduced that effects the runtime of the algorithm to a high degree. Through a thresholded weight function the integral in the voting function need not be calculated over all pixel in the image, but only over pixel positions where the magnitude of the gradient is above a certain threshold θ. This decision can be justified by the fact that low weights do not contribute much to the integral and are also not likely to belong to the edge of the coin we are looking for. Due to this, we may ignore large parts of the image.

3. The third aspect where the proposed implementation varies from Reisert's is concerns an implementation detail. Reisert uses an infinite impulse response filter for smoothing — both in preprocessing and in smoothing the accumulator. This fast filter was not available in the toolkit that has been used. Instead a separated Gauss finite impulse response filter has been used during the experiment. The given implementation of the Hough segmentation still leaves the possibility for external preprocessing open, so that in a production environment an IIR preprocessor could lead to an overall speed increase by a small fraction.

Formally the voting function $\acute{P}$ used by the modified approach may be written in the following form:

Let θ be a positive real threshold value and $W_\theta : \mathbb{R}^2 \mapsto \mathbb{R}$ the modified weight function so that:

$$W_\theta(\mathbf{y}) = \begin{cases} 0 & \text{if } W(\mathbf{y}) < \theta \\ W(\mathbf{y}) & \text{otherwise} \end{cases} \tag{2}$$

The voting function $\acute{P} : \mathbb{R}^3 \mapsto \mathbb{R}$ is then written as:

$$\acute{P}(\mathbf{x}) = \int_{\mathbb{R}^{\not\approx}} \Big(\acute{\delta}_\lambda(\mathbf{x} + rN(\mathbf{y})) + \acute{\delta}_\lambda(\mathbf{x} - rN(\mathbf{y}))) \Big) W_\theta(\mathbf{y})) d\mathbf{y} \tag{3}$$

$\acute{\delta}_\lambda$, the indicator function used is dependent on the current recursion level λ in hierarchical voting. It is the result of the implicit smoothing due to the reduced size of the accumulator and has the effect of a convolution with an averaging filter (Spalttiefpass) which has a square kernel that's size is determined by the ratio of image size to accumulator size in pixel. At the lowest level in recursion where image size and accumulator size match, no implicit smoothing is done anymore. The indicator function at that time is an averaging kernel of exactly one pixel in width.

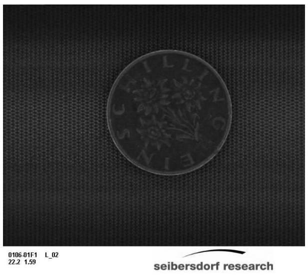
The implementation was tested on the small training MUSCLE CIS coin competition 2006 database [1]. (The images in MUSCLE CIS were recorded under controlled conditions: each image shows exactly one coin on a conveyor belt. The images are all 640 by 575 pixel in size and 8 bit one channel grayscale in depth. Each image is divided into two areas. The upper 640 by 512 pixel contain the coin. The lower area contains a logo which has to be removed for coin detection. Part of the uppermost row of pixel in the coin photography is substituted by an ASCII string holding meta information about the image. One has to take care not to interpret these pixel as image information e. g. in edge detection. Exposure and/or aperture has varied among different images, as can be seen in the variance in appearance of the conveyor belt (see Fig. 1).) An example of an accurate MUSCLE CIS coin Hough segmentation is shown in Fig. 2(a). The majority of the MUSCLE CIS coins were segmented with less accuracy. Fig. 2(b) shows an example of inaccurate Hough segmentation. Note the slight extension to the lower right. Improvements in the maximum search strategy in the accumulator should be investigated to increase the accuracy of the implemented method.

References

[1] MUSCLE CIS benchmark site: http://muscle.prip.tuwien.ac.at, last visited 05/22/2007.

[2] Christian Kotz. "Practical Work: Automatic Coin Recognition. LVA-Nr. 183.176". Pattern Recognition and Image Procesing Group. Institute of Computer Aided Automation. Faculty of Informatics. Vienna Univerity of Technology. unpublished. Oct. 2007.

[3] Marco Reisert, Olaf Ronneberger, and Hans Burkhardt. "An Efficient Gradient Based Registration Technique for Coin Recognition". September 11, 2006. In: *Proceedings of the MUSCLE CIS Coin Competition Workshop*. September 11, 2006. Berlin, Germany. Ed. by Michael Nölle and Michael Rubik. 2006. Pp. 19–31. URL: `http://muscle.prip.tuwien.ac.at/coin_workshop2006_proceedings/reisert.pdf` (visited on 01/2008/23).



(a) aperture wide?

(b) aperture narrow?

(c) dirt on conv. belt

Figure 1: MUSCLE CIS coin images

(a) acurrate Hough segmenatation.

(b) inacurrate Hough segmenatation.

Figure 2: Hough segmentations of MUSCLE CIS coins.