社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Position Detection For a Camera Pen Using LLAH and Dot Patterns

Matthias SPERBER†, Martin KLINKIGT†, Koichi KISE†, Masakazu IWAMURA†, Benjamin ADRIAN††, and Andreas DENGEL††

† Department of Computer Science and Intelligent Systems, Osaka Prefecture University
†† German Research Center for Artificial Intelligence
E-mail: †{matthias,klinkigt}@m.cs.osakafu-u.ac.jp, ††{kise,masa}@cs.osakafu-u.ac.jp,
†††{Benjamin.Adrian,Andreas.Dengel}@dfki.de

Abstract In this report, we present a method of position detection for a low-cost camera-pen. Our method allows reconstruction of handwriting, as well as retrieval of the particular document that is being written onto. For this matter, we use a point pattern printed onto the document's background, which is indexed using Locally Likely Arrangement Hashing. For position detection, the dot pattern is extracted from the camera image and then matched to the hash table. Experimental results show high retrieval accuracy (80.4%∼100.0%) when the pattern is disturbed by the document's foreground only to a certain extent.

Keywords Pen Computing, Locally Likely Arrangement Hashing, Document Recognition

1. Introduction

This report proposes a new method of developing a camera pen which enables reconstruction of handwriting in digital form. Many cases exist in which the use of paper documents is preferred to digital documents, despite all technological advances. This is true both printed media and handwritten notes. There are several important reasons for using printed text, including the higher readability as compared to a computer screen. Also on paper documents, contents can be added directly using handwriting, which is often more convenient and flexible.

On the other hand, there are also many cases in which digital technology is superior. Digital files are very flexible for storing and organizing, and can easily be shared and distributed. Their contents can be changed, deleted or reformatted very conveniently. Moreover, digital documents allow automatic further processing.

Digital pens attempt to bridge the gap by combining the advantages of both. They make it possible to use handwriting, and at the same time automatically transfer it into digital form. To realize such a digital pen, technologies must be provided to recognize the pen tip position. Both the local position on the document and the document itself should be recognized fast and accurately, so as to obtain a smooth image of handwriting. This technology should ideally not interfere with the conveniences the non-digital tools offer, in addition to being affordable and easy to handle.

In this report, we propose a camera based approach. The current position is detected by analyzing the image provided by a camera mounted on the pen. First, a randomized pattern of tiny dots is printed on paper, along with possibly a document in the foreground. Dot arrangements are stored in a database using Locally Likely Arrangement Hashing (LLAH) [1]. Our system can extract the dot pattern from a camera image, and retrieve the corresponding document from the database. Using this information, we can extract the camera angle and finally determine the pen position.

Our experimental results show that for the case that only a moderate amount of pattern dots are concealed by the document foreground, our method achieves a high accuracy between 80.4% and 100.0%. This is sufficient for accurate handwriting recognition. We achieved good results for a database size of 100 and 1,000 indexed documents, though in the latter case accuracy dropped noticeably. Our method is reasonably fast, but must be further optimized to allow real-time handwriting reconstruction.

2. Related Work

Various technologies for realizing digital pen systems have been developed and put into products. The following gives a brief overview of existing systems, as well as their advantages

and disadvantages.

2.1 Pen Tablets

A pen tablet is a flat electronic device that can be written on with a stylus. Writing is captured using a weak magnetic field. Similar to the other related work, it is also possible to use this technology with a sheet of paper, by placing it on top of the tablet. However, the position on the paper has to be calibrated. Also, no document context can be retrieved automatically, i.e., it is not known which particular document is currently used. This technology is much less portable than some of the other available systems. Existing products include [2].

2.2 Clip-On Solutions

Here, the writing surface is attached to a special clip-on device. The pen tip location relative to this device is measured constantly using triangulation of ultrasonic waves. This technology works on any writing surface, for instance sheets of paper [3] or blackboards [4]. Another advantage is the low price. On the other hand, for each document or other writing surface, the device first has to be clipped on and calibrated, and no document context is available.

2.3 Anoto

Anoto technology [5] is perhaps the most advanced available technology. Position information is encoded directly on the paper, in form of a grid of fine black dots. Each dot is displaced from its original position on the grid in one of four directions. The pattern is printed using carbon-based ink, and captured by an infrared camera close to the pen tip. While being very portable, easy to handle, and able to distinguish a very large number of documents, it is also rather pricey, concerning both the required hardware and the licensed dot paper. Also, the black dots are rather apparently visible.

2.4 Using Paper Structure

In [6], a camera pen has been proposed which uses the microscopic fiber structure of paper and video-mosaicing to reconstruct handwriting. This method is highly portable and inexpensive. However, calibration is not feasible and hence only relative movement can be captured. Document context is also unavailable.

2.5 Using LLAH On Document Content

In [7], a camera pen has been proposed which uses LLAH and centroids of connected components as feature points to recognize the position on the pen. It is again highly portable and inexpensive, and moreover able to recognize the local position on the document, as well as the document itself. One disadvantage is that this technology does not permit writing on empty sheets of paper, or blank spots of documents.

3. LLAH

The core technology the proposed method relies on is LLAH, which shall be introduced in this section. LLAH is a method that indexes local combinations of feature points. In our case, feature points are the individual dots of the dot pattern. These combinations are stored in a hash table and can be retrieved using a voting process. This allows for arbitrary arrangements of captured feature points, in our case the yellow dots extracted from the camera image, to be matched to their corresponding point arrangements in the database and thus retrieve their location. LLAH is document based. Hence, the *location* in this context comprises both (1) the document ID, and (2) the local position on the document.

The following subsections describe the LLAH process as found most appropriate for the camera pen, for the most part corresponding to the improvements found in [8] and [9].

3.1 Calculation of Features

Features calculated from feature points must satisfy two properties:

Stability. For one feature point, the same feature should be calculated, even under perspective distortions, noise or occlusion of parts of the document. This is achieved by using geometric invariants. These are values calculated from a number of points, which remain constant under geometric transformation. Though perspective invariants would be the mathematically appropriate tool, for performance reasons we approximate them using affine invariants. These are called *area ratios* and calculated from four points A, B, C, D as $a = P(A, C, D)/P(A, B, C)$, where P denotes the triangle area function. To obtain four points A, B, C, D, the closest surrounding feature points are determined. However, after perspective distortion, different feature points can move closest. Thus, an assumption is made which holds true for certain extents of distortion, stating that, out of the surrounding n closest feature points, m remain constant under distortion. The feature is then calculated using every possible subset of cardinality m out of the n nearest points. This also compensates for missing or false individual features due to noise and extraction errors. In addition, the use of only local arrangements of feature points enables LLAH to retrieve the correct document even if only partly visible.

Discrimination Power. Features calculated from different feature points should have different values, so as to be able to distinguish between them. The simplest case is to choose $m = 4$ and set the feature equal to the area ratio of these four surrounding points. However, it is often the case that different sets of four points are arranged similarly and hence yield similar invariants. A better solution is to choose $m > 4$ points and calculate the feature from all possible subse-

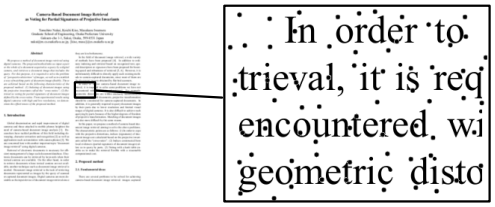


Figure 1 Randomized dot pattern (in black for demonstration).

quences containing four of these points instead. The feature then consists of the sequence of discretized $\binom{m}{4}$ invariants.

3.2 Storage

For later retrieval, all document must first be indexed in a hash table. For each feature point, features are calculated as above. The result is a number of sequences of discretized invariants of the form $F = (r_{(0)}, r_{(1)}, \ldots, r_{(\binom{m}{4}-1)})$, representing one feature point. For each sequence, a hash index is calculated using the formula:

$$H_{\text{index}} = \left(\sum_{i=0}^{\binom{m}{4}-1} r_{(i)} k^i \right) \bmod H_{\text{size}} \qquad (1)$$

where k denotes the level of quantization, H_{size} is the number of bins in the hash table. The tuple (document ID, point ID) is stored in the hash table at the respective indices. When a collision occurs, the hash entry is marked invalid [9].

3.3 Retrieval

For retrieval, hash indices are calculated following the same procedure as for storage. A vote is cast for the document IDs found at the corresponding hash entries. The document with the largest number of votes will be considered the correct document.

4. Proposed Method

4.1 Generating Dot Patterns & Creating Indexes

To enable handwriting reconstruction, a randomized pattern of tiny dots is printed on each document, such as shown in Fig. 1. An unobtrusive color such as yellow is used. To generate dot patterns, our method initially produces a regular grid of dots, with a fixed distance in between. These are then displaced both horizontally and vertically by a random offset according to a Gaussian distribution. Offsets are required to lie within the bounding square of the point, to preserve a certain level of regularity. In other words, no "holes" are allowed. This is important since for all possible positions of the pen on the paper, enough dots must be visible for position detection.

This method of generating the pattern was chosen as a tradeoff between two factors: The dot pattern should appear fairly regular to the eye, in a way that readability of document contents is not disturbed. On the other hand, each pattern must be distinctive so as to enable our method to retrieve the correct document.

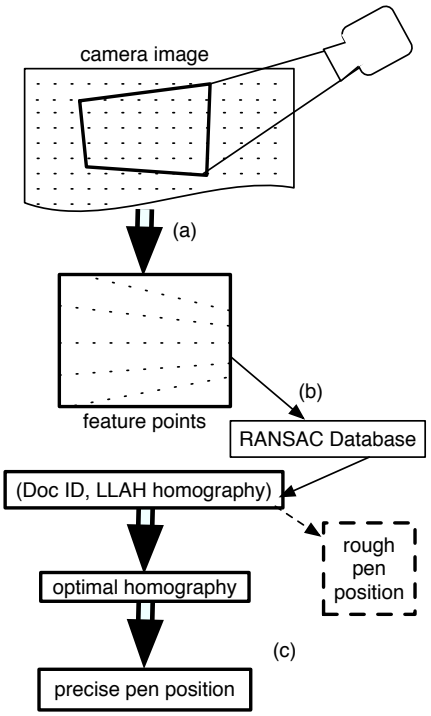


Figure 2 *Retrieval steps.* (a) Extraction of feature points from camera image, (b) LLAH query (c) determination of pen tip position using either the estimated *RANSAC homography* or the *optimal homography*.

Simultaneously, LLAH is used to index and store dot arrangements in a hash table. One of the key properties here is the use of *local* arrangements of feature points. This enables correct retrieval even for the highly limited viewing area of the camera pen, under the condition that enough feature points can be extracted.

4.2 Retrieving the Pen Position

The pen tip position is determined from the provided camera image for each frame. Figure 2 shows the steps of retrieval. First, the yellow dots are extracted from the image (Fig. 2(a)) as follows: A "distance image" d to the target color yellow is created using RGB representation and applying the following formula to each pixel (x, y):

$$d(x,y) = \sum_{c \in \{r,g,b\}} (w_c \cdot |t_c - p_c(x,y)|) \qquad (2)$$

where c runs through the three color channels, t_c denotes the channels' values of the target color and p_c those of the provided camera image. The three color channels are weighted as $w_r = \frac{2}{9}, w_g = \frac{3}{9}, w_r = \frac{4}{9}$, based on experimental results. Next, *adaptive thresholding* is performed on the result, and noise is removed using *dilation*. Finally, feature points are determined as centroids of connected components.

In the following step, these feature points are used to query

the LLAH database and retrieve the document (Fig. 2(b)). Feature points are matched and used to determine the perspective transformation (or *homography*) which transforms the original points into the arrangement captured by the camera. This is done using RANSAC [10] and ultimately enables determining the pen tip position. By ignoring outliers, RANSAC is able to estimate a homography that is very close to the actual one. However, the RANSAC approximation of the current position tends to be unstable when used at a very precise level. In other words, reconstructed handwriting is not smooth.

We thus suggest an additional step in which the *RANSAC homography* from the previous step is applied on all points in the database that belong to the retrieved document. The transformed dots are then matched to the nearest extracted dots from the camera image. The number of matched points is then much higher than the number of matches retrieved from LLAH. Also, this time it is simple to ignore outliers by imposing a distance threshold when performing matchings. From these matchings, a second homography is calculated which is optimal in terms of the least-squares error. Because of the high number of matches and the unlikeliness of outliers, results can be highly improved using this *optimal homography*, rather than the RANSAC homography (Fig. 2(c)) to calculate the pen tip position. This technique creates much smoother handwriting, at the expense of longer processing time.

Finally, some basic error detection is applied, using the observation that often, within a streak of correctly recognized documents, a few individual erroneous ones can be found. Algorithmically, we consider windows of successively retrieved documents. For any window of size l, $n > \frac{l}{2}$ documents are required to be equal in order to be marked correct. All dissenting results are marked incorrect and ignored for handwriting reconstruction. The use of windows is necessary because the user may write on several documents during one session, so simply considering the most frequent document as the correct one will not do the job.

5. Experimental Results

To investigate the usefulness of our method, we evaluated performance as well as each of the retrieval steps as shown in Fig. 2.

For the experiments, LLAH parameters were set to $n = 7$, $m = 6$, $k = 15$. Affine invariants were used. The size of the hash table was $1.34 \cdot 10^8$ bins, and for collisions, the corresponding hash entry was marked invalid. For the dot pattern, initial distance between dots was set to 2.7mm, which is equivalent to 7918 dots per document. Dot diameter was set to 0.2mm. The pattern was printed using a laser printer.

Figure 3 *The camera pen.* It contains an ordinary ballpoint pen and a tiny USB camera.

Table 1 *Performance.* Runtime needed for *complete* retrieval step with respect to database size and number of dots used for query.

DB size	101 points	80 points	50 points
1 document	23.8ms	20.9ms	18.5ms
100 documents	30.6ms	24.4ms	22.0ms
1,000 documents	53.5ms	51.9ms	49.4ms

The computer hardware featured an Intel Core CPU clocked at 2.13GHz, and 3GB RAM. For the pen, we used a low-end USB camera with a resolution of 720×576, providing 30 frames per second. The construction of the pen can be seen in Fig. 3. When facing straight down, the camera's distance to the document was about 3.3cm, providing a captured area of about 2.5×2.0 cm^2, or equivalently 68.8 dots. The actual number of dots was often higher because of a steeper camera angle when writing.

5.1 Performance

We measured performance of the retrieval step. Extraction of dot coordinates from the camera image took up a fixed amount of time, about 13.7ms. The remainder of runtime was primarily needed by the LLAH query and depended strongly on the number of documents in the database and the number of dots used for the query, as can be seen in Table 1. For the more desirable database size of 1,000 documents, about 50ms of CPU time were needed. This would allow 20 frames per second, which is insufficient for realtime capture of appropriately fast writing. For smaller databases, performance can be strongly improved by artificially decreasing the number of points. This, however, makes the calculated position less precise and thus should be considered carefully. Note that for the case of only one document in the database, runtime is much faster. It might thus be possible to achieve realtime processing speed by using a hash table that only contains the correct document, once the current document is known.

The numbers shown in Table 1 only include calculating the RANSAC homography. Required time for calculating the optimal homography depends strongly on the quality of the LLAH result, but roughly multiplies retrieval time by a factor up to two.

Table 2 *Extraction accuracy.* For the three examples shown in Fig. 4, the number of correct points and falsely extracted points, and the mean square error are denoted, each averaged over all frames marked as correct.

	visible document content		
property	none	little	much
number of correct points (avg.)	77.7	72.6	65.9
number of false positives (avg.)	3.1	6.5	10.9
mean square error	1.9 px	2.5px	3.7px

Table 3 *LLAH accuracy.* Examined for the three examples shown in Fig. 4, each for the small and large databases, respectively.

	visible document content		
DB size	none	little	much
100 documents	100.0%	94.0%	74.4%
1,000 documents	96.2%	80.4%	59.3%

5. 2 Extraction of the Dot Pattern

To evaluate our method of extracting the dot pattern (Fig. 2(a)) from the camera image, we prepared video frames, capturing the trajectory of writing the word "hello" three times: (a) No document content was visible, (b) a small amount, (c) a larger amount of content (see Fig. 4). Care was taken to impose minimal changes to lighting conditions and camera angle between the three videos.

Our goal was to measure how well points extracted from the image match to points in the LLAH database. For this purpose, we applied the *optimal homography* on database points, and matched these to the extracted points using nearest neighbors with a distance threshold. The number of documents in the LLAH database was 100. Table 2 shows that for more document content, the number of correct matches drops, due to occlusion. Also, the number of falsely extracted points increases, partly because additional content distorts the adaptive thresholding.

Finally, we also measured the mean square error of detected positions, by using the matches to calculate a homography from the correct points to the extracted points which is optimal in terms of the least-squares error. Table 2 shows that extraction with little document content is also more accurate in terms of this value. This is (1) because of effects of the two previously discussed observations, and (2) because with more content, yellow dots are more likely to be partly hidden, moving the extracted centroid of the dot away from their actual center.

5. 3 LLAH Accuracy

For evaluating the accuracy of LLAH (Fig. 2(b)), we used the same video frames used in Sect. **5. 2**. For each of the three examples, we determined the number of frames that met two conditions: (1) The correct document was recognized by LLAH, (2) the determined position was within a bounding rectangle of roughly $2.9 \times 1.5\text{cm}^2$, or 680×360 pixels, drawn around the actual handwritten word. This experiment was performed using two databases, containing 100 and 1,000 documents respectively.

Table 3 shows almost perfect result for the case of no disturbance by document contents, and good results for moderate disturbance. However, for the third example, accuracy dropped heavily, especially for the large database.

5. 4 Handwriting Reconstruction

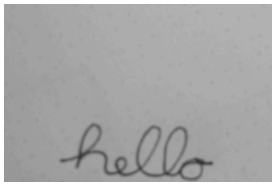
After showing potential and limitations of the proposed method on a more theoretical level, in this section we demonstrate the actual quality of reconstructed handwriting (Fig. 2(c)). Once again, we used the videos from the previous sections, this time only the one with "little" document content. For visualization, a naïve approach of drawing straight lines between consecutively determined pen coordinates was employed. Figure 5 shows the reconstructed handwriting using the small and the large database, and for each the *RANSAC homography* and *optimal homography* approaches. It can be seen that the first approach is still readable, at least for the small database, though rather unsatisfying (Fig. 5(a) and 5(c)). The second one, on the other hand, looks very smooth (Fig. 5(b)). For the large database, some stray errors occurred, but these are few and could be easily detected using some smoothing technique (Fig. 5(d)). They might also be completely avoided by using a hash table containing only the correct document, as suggested previously.

As a result, it can be seen that a smooth image of handwriting can be reconstructed up to an achieved LLAH accuracy of about 80%. If the accuracy drops farther below that, as was the case with the third video example (Fig. 4(c)), the extracted image is no longer nicely readable.

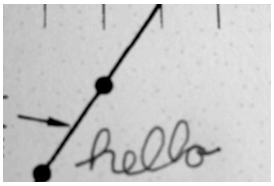
6. Discussion & Outlook

The experiments show reasonable results for the case in which the dot pattern is occluded only to a certain extent. Problems arise when too much document content interferes with the extraction of the pattern. One method of avoiding this problem is to generate the dot patterns in a way such that no collisions with document background and foreground will occur. That is, each time a dot would collide with content, a new random offset is drawn instead. Another way is to use a combination of feature points extracted from both the background, as described in this report, and the foreground, as described in [7].

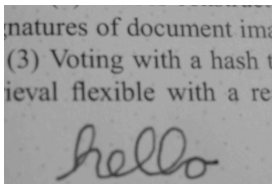
A different technique of avoiding this problem is to avoid collisions on a technical level, much similar to Anoto's solution to this problem. Here, for the dot pattern, carbon-based ink is used, while the document content is applied



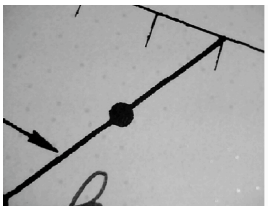
(a) No disturbance by document foreground.



(b) Little disturbance by document foreground.



(c) Much disturbance by document foreground.

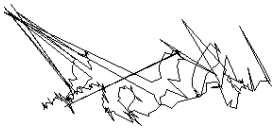


(d) Example of actual camera view for Fig. 4(b).

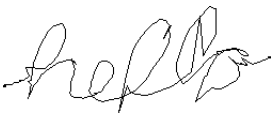
Figure 4 Handwriting to be recognized with three levels of difficulty. Note that the foreground *above* the handwriting is that of interest, since this is what the camera captures, as can be seen in Fig. 4(d).

(a) RANSAC homographies, 100 documents in DB

(b) Optimal homographies, 100 documents in DB



(c) RANSAC homographies, 1,000 documents in DB



(d) Optimal homographies, 1,000 documents in DB

Figure 5 Images of handwriting reconstructed for the example shown in Fig. 4(b).

using carbon-free ink. The dot pattern then remains visible to the infrared camera, even if concealed by content. This solution, however, comes at the price of higher cost for the proposed technology.

The experiments also show decreasing accuracy for a database containing 1,000 documents, as compared to 100 documents. This is due to the large number of features that have to be distinguished. For 1,000 documents, their number is 7,918,000. To further increase the database, one possibility would be increasing the level of quantization for LLAH. However, this might lead to problems with robustness. Other possibilities include investigating ways to make the dot arrangements more discriminative.

We mentioned earlier that we built the camera pen by mounting a USB camera on a ballpoint pen. In this case, a wired connection to the processing computer is required, which might not be available in some situations, or inconvenient in others. For these cases, the USB connection should be replaced either by a wireless connection, or a memory stick that can later be connected to the computer. The latter case furthermore requires a processing unit in the pen to store extracted dots on memory. Also, for averagely fast writing, a high-speed camera is needed. For cameras with lower speed, too few positions a captured when writing fast. Finally, also a mechanical unit in the pen should be employed to detect at what times the pen is touching the paper.

7. Conclusion

We presented a new method of developing a camera pen. The main advantage is its low cost. We used a cheap camera for the pen, and printed the dot pattern using an ordinary laser printer.

Our method allows reliable reconstruction of handwriting up to a certain level of disturbance by visible document content. For sufficient accuracy for the more difficult cases, additional methods must be investigated. Processing speed is reasonably fast, although not usable for real time handwriting reconstruction.

Future work for the proposed method includes the limited amount of supported documents, and further investigating methods to enable real-time processing.

Acknowledgements. This work was in part supported by the Grant-in-Aid for Scientific Research (B) (19300062) and the Grant-in-Aid for Challenging Exploratory Research (21650026) from Japan Society for the Promotion of Science (JSPS).

References

[1] T. Nakai, K. Kise and M. Iwamura: "Hashing with local combinations of feature points and its application to camera-based document image retrieval", Proc. CBDAR2005, pp. 87–94.

[2] Wacom Co., Ltd., http://www.wacom.com/.

[3] PC Notes Taker, Pegasus Technologies, Ltd., http://www.pc-notetaker.com/.

[4] mimio Xi, Virtual Ink Corp., http://www.mimio.com/.

[5] Anoto AB, http://www.anoto.com/.

[6] S. Uchida, K. Itou, M. Iwamura, S. Omachi and K. Kise: "On a possibility of pen-tip camera for the reconstruction of handwritings", Proc. CBDAR2009, pp. 119–126.

[7] K. Iwata, K. Kise, T. Nakai, M. Iwamura, S. Uchida and S. Omachi: "Capturing digital ink as retrieving fragments of document images", Proc. (ICDAR2009), pp. 1236–1240.

[8] T. Nakai, K. Kise and M. Iwamura: "Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval", Lecture Notes in Computer Science (7th International Workshop DAS2006), pp. 541–552 (2006).

[9] T. Nakai, K. Kise and M. Iwamura: "Camera based document image retrieval with more time and memory efficient llah", Proc. CBDAR2007, pp. 21–28.

[10] M. A. Fischler and R. C. Bolles: "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Commun. ACM, 6, pp. 381–395 (1981).