MODEL-BASED CHARACTER RECOGNITION IN LOW RESOLUTION

Annika Kuhl, Tele Tan, Svetha Venkatesh

Department of Computing, Curtin University of Technology
GPO Box U1987, Perth, 6845, Western Australia
(a.kuhl, t.tan, s.venkatesh)@curtin.edu.au

ABSTRACT

We propose a combined character separation and recognition approach for low-resolution images of alphanumeric text. By synthesising the image formation process a set of low-resolution templates is created for each character. Cluster algorithms and normalised cross-correlation are then applied to match these templates and thereby allowing both character separation and recognition to be achieved at the same time. Thus characters are recognised using their low-resolution appearance only without applying image enhancement methods. Experiments showed that this approach is able to recognise low-resolution alphanumeric text of down to 5 pixels in size.

***Index Terms*—** Optical Character Recognition, Text processing, Number Plate Recognition

1. INTRODUCTION

The problem of low-resolution character recognition arises in many situations such as number plate recognition for large scale surveillance or text documents captured by low-cost surveillance cameras or mobile phones. Although small, low cost and low-resolution cameras are omnipresent, existing recognition algorithms for degraded low-resolution footage are far from being robust and deployable. Most existing optical character recognition (OCR) approaches require a resolution of at least 300dpi for an A4 page of font size 10, which equals a character height of at least 30 pixels [1]. As OCR algorithms work on binary images only, a lower resolution will result in degraded or concatenated binary characters, leading to recognition errors. This is because low-resolution characters of less than 20 pixels in height do not show clear edges, but appearing instead as amalgamates of mixed grey scale pixels. Thus, low-resolution character recognition is challenging. Existing work on low-resolution number plates uses super-resolution [2] for visual improvement. As an alternative a neural network based word recogniser for images has been proposed by [3].

By modelling the image formation process, we propose a recognition process that is driven by parameterised templates. Low-resolution templates of each character are generated in a way that reproduces the degradation caused by down-sampling and subsequent discretisation of the imaging chips. These templates are then used by a template matching based combined character segmentation and recognition system, making character separation or detection unnecessary.

The novelty here is the synthesis of a set of low-resolution templates for each character derived from the image formation process. Normalised cross-correlation is used for matching these. This method is well studied, robust to lighting changes and fast when small size images and templates are used, making it practicable. Clustering is applied to determine the best matching character for each position in the image and thus simultaneously solving for character segmentation.

2. BACKGROUND

For recognising segmented single characters several feature extraction-based methods have been proposed in the past [4]. In order to process degraded low-resolution document images there are several approaches to enhance the quality and resolution of these images, super-resolution and resolution expansion are proposed by [5] and [6] respectively. The authors of [6] use image expansion to increase the resolution of low-resolution text documents via constrained interpolation. This method is aimed to enhance the image resolution in order to improve OCR results as OCR requires character separation, which is very difficult in low-resolution images. But most enhancement methods modify the image in a way which may create artefacts that adversely affect the OCR. Our proposed approach recognises characters by their low-resolution appearance and thus avoids any intermediate image enhancement technique that could create additional errors.

Previous work on low-resolution character recognition of document images is based on dual eigenspace decomposition and the synthetic generation of data [7]. A large set of degraded characters is modelled by down-sampling a clear binary image followed by image zooming. These resulting blurred characters are then used for training. This approach still uses heuristics to segment single characters which is still a challenge in low-resolution due to merging of characters.

The method most similar to our approach uses a neural network based character recogniser together with dynamic programming to segment and recognise low-resolution words in a single step [3]. They create low-resolution feature models of each character with which a neural network is trained on. For recognition the word is broken up into slices and dynamic programming is used to determine the best combination of slices that yield the most probable word.

In dealing with number plate recognition most existing algorithms consist of the following steps: (1) number plate detection, (2) pre-processing, (3) character segmentation, (4) character recognition. While there are methods for detecting number plates in low-resolution images [8], most approaches for low-resolution number plate recognition use image enhancement techniques to prepare the number plate for OCR [2] or high-resolution plates are assumed [9]. Super-Resolution is also used as a pre-processing step to enhance low-resolution number plates in [2]. Several frames of a low-resolution video are combined to form a high-resolution image, but this only achieves a visual improvement not actual recognition.

3. METHOD

The image formation process of the optical camera system can be modelled as a linear system [10]

$$Y = AX + z \tag{1}$$

where X is the high-resolution image that is degraded by the degrading matrix A, z models the uncertainties due to noise and Y is the resulting low-resolution camera image. The degrading matrix A represents image warp, motion blur and blur due to the optical system, as well as the down-sampling process caused by the finite and discrete imaging chip.

Determining the high-resolution image X from one or more low-resolution images Y is an ill-posed problem [6]. The degrading matrix A is unknown and hard to estimate. Thus, we propose a simplified model of the image formation process by considering only the down-sampling process.

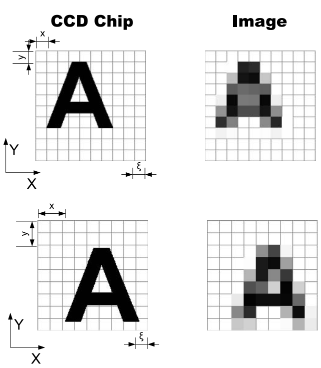


Fig. 1. High-resolution 'A' overlaid on low-resolution grid (left) and down-sampled (right).

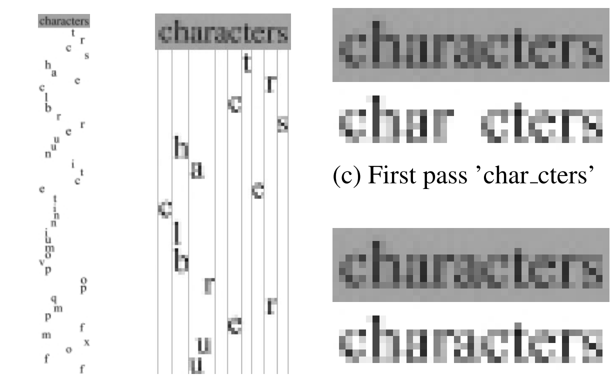
Assuming a perfect imaging chip, neglecting blur and constraining the image warp to translation in the 2D image plane, we synthesise the image formation process using the three parameters x, y and ξ, as shown in Fig. 1. As the high-resolution image is projected onto the finite chip it is down-sampled according to the size of the pixels ξ, where ξ equals the number of high-resolution pixels that fit into a single low-resolution pixel. The parameters x and y, which describe the offset, determine the grey value appearance of the low-resolution character template, as the grey value of each pixel equals the percentage of its coverage.

Thus in order for each character of the alphabet to be sufficiently modelled a high-resolution binary image of each character is required. This image is down-sampled according to the three parameters, x, y and ξ, and the grey values of each low-resolution pixel are computed. The whole process of creating several templates for each character can be done offline, incrementing x, y and ξ as described in Sec. 3.1.

It should be pointed out that this template generation process is based on the down-sampling process of the imaging chip. The proposed recognition method is therefore not suited for images that have been modified after acquisition, as most image enhancement or super-resolution technique will not preserve the original grey value appearance of the low-resolution characters, making them unsuitable as inputs for our approach.

3.1. Low-Resolution Word Recognition

Throughout this paper we use the term 'word' to describe a string of alphanumeric characters contained in an image. The problem that arises with word recognition in low-resolution is that adjacent characters can become merged, increasingly so as resolution decreases, making single character detection as well as recognition difficult.



(c) First pass 'char_cters'

(a) Initial. (b) Reduced set. (d) Final result 'characters'

Fig. 2. Partial steps of the proposed combined character detection and recognition approach for the word 'characters'.

We propose a parameterised template matching approach for low-resolution character recognition that can be used for combined character detection and recognition based on the down-sampling process described earlier. Instead of first trying to separate each character of the word and then applying character recognition techniques, we perform both processes in a single step. Therefore k templates for each character $c_i, i = \{\{A..Z\} \cup \{0..9\}\}$ are created as follows

$$c_i^k = T(c_i, x, y, \xi) \tag{2}$$

where c_i is a high-resolution binary image of size $H \times W$ and T creates a template using the parameters x, y and ξ, which are all integer multiples (e.g. Fig. 1). The created template c_i^k is of size $\frac{H}{\xi} \times \frac{W}{\xi}$ and the height of the input image I is used to estimate a rough range of the parameter ξ which determines the size of the low-resolution template - the bigger ξ the smaller the resulting template (see Fig. 1). k ranges across all combinations of $\{x, y, \xi\}$ with $x \leq \xi$ and $y \leq \xi$. Furthermore the font type is assumed to be known in order to choose matching templates. For each character template c_i^k, the position where c_i^k correlates best is given as

$$p_i^k = \underset{R}{\operatorname{argmax}}\{\operatorname{ncorr}(c_i^k, R, I)\}, \tag{3}$$

where I is the cropped image containing the word or part of the word and ncorr calculates the normalised cross-correlation between I and the template c_i^k at the 2D image position R.

After p_i^k is calculated all templates with correlation values below a certain threshold, empirically defined to be 0.75, are deleted. We further clean the remaining set of templates by removing duplicate templates of each letter retaining the best match at each position. This can leave more than one template of each letter but they will be at different positions within the image.

The result after these steps for the low-resolution image of the word 'characters' is shown in Fig. 2(a). The character templates are ordered according to their correlation value, starting with the highest at the top. The characters of this word, only five pixels high on average, are merged and partially overlap each other, making it difficult to separate them.

This initial character template selection in Fig. 2(a) is further reduced by selecting the top character matches, starting with the highest correlation value, until the last m character templates that have been added do not match new image pixels. In the example in Fig. 2(b) $m = 4$ and the position of the last four letters 'r', 'e', 'u' and 'u' is already covered by previous templates, therefore no more need to be added.

In a last step the remaining character templates are clustered to select the character that best fits each position. Hierarchical clustering is used to find disjoint columns, where each column represents one or more characters of the word. Fig. 2(b) shows the cluster bounds of each column. Within each column the set of character templates with the maximum average correlation value across all possible permutations is chosen as the best character match. A possible permutation consists of a set of character templates whose positions fill out all pixels in the column. In the example in Fig. 2(b) the third column consists of {'a', 'r', 'u', 'u'}. The set {'a', 'r'} fills the whole column and achieves the highest average correlation value. The intermediate result is shown in Fig. 2(c), with the cropped image at top and the synthesised character templates below. Unshown in the example are problems arising from e.g. 'm' and 'nn'. These can be easily confused and a dictionary lookup would be needed to disambiguate them.

Not that the second 'a' in Fig. 2(c) is not yet recognised after the first run. A post-processing step detects gaps of not yet recognised characters and the algorithm is recursively performed on each of the gaps individually using the cropped image gaps as input image I. This is uncommon but these gaps are most likely to occur in words with several identical characters where all matches are initially located at the same position in the image. Or the reduced template set (see Fig. 2(b)) cuts off too many character templates.

Even though a large number of templates is used for each word, this heuristic template matching approach for character recognition is still practicable as the size of the low-resolution input image is quite small and so are the templates used. The approach is also suitable for FPGA hardware implementation which will provide a further speed up in performance. Furthermore the simplified model of the image formation process does not require the estimation of the degrading matrix A.

4. EXPERIMENTS

4.1. Suitability and Interchangeability of Different Fonts

For each font type (Arial, Times New Roman, and the fonts used for standard Western Australian (WA) and standard German (G) number plates) we printed each of the 36 characters (26 upper case letters and 10 numbers) on a piece of cardboard using a bold font size of 130 for Arial and Times and equivalent font sizes for the WA and G number plate font. These cardboards are placed 2m in front of a camera with a resolution of 320×240 which results in an average character height of 6 to 7 pixels. To achieve variations in the grey value appearance, the position of the cardboards is translated during recording. An example of the letter 'A' of the WA font in nine different frames is shown in Fig. 3. Even though these images are affected by noise the generated templates model the original image well, achieving correlation values of 0.98.

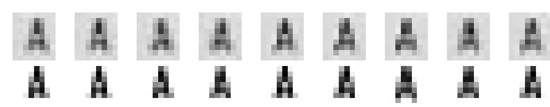


Fig. 3. Different grey value appearances of the letter 'A' (top) and the synthetically generated templates (bottom).

For comparing the suitability and interchangeability of different font types, we used 20 frames of each character of each font and recognised them using each of the four font types. The result is shown in Tab. 1, where each row represents a particular font used for recognition and each column denotes a test sequence of the true font. As can be seen the main diagonal of this correlation matrix achieves maximal recognition rates. Times New Roman appears to be the font most difficult to recognise - with only 89%. This is mainly due to its serif style which results in characters that appear less bold compared to sans-serif font types of the same size.

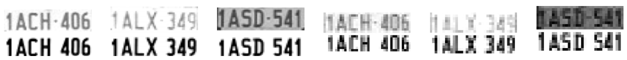
true font →	**Arial**	**WA**	**German**	**Times**
Arial	**98.05%**	38.61%	47.50%	46.81%
WA	42.36%	**92.92%**	60.28%	22.92%
German	53.05%	66.66%	**97.50%**	33.75%
Times	50.00%	19.44%	22.08%	**89.03%**

Table 1. Character Recognition Results for different fonts.

Arial and the German number plate font seem to be best suited for low-resolution character recognition. Arial is a sans-serif font whose characters are all quite distinct compared to standard WA number plate font. The German number plate font is a special falsification-hindering script which also eases recognition. Summarising, the font type of the characters should be known beforehand in order to choose matching templates as non-matching fonts decrease recognition rates immense especially in low-resolution.

4.2. Number Plate Recognition

We further tested the method on WA number plates in outdoor conditions. The distances between the camera and the plate is about 1.5m and 3m respectively. The resolution of the recorded video is 320×240 resulting in average character heights of 13 and 7 pixels respectively. The position of the camera is slightly varied while the video is recorded. We recorded 20 frames of each of the 36 different number plates.



(a) 1ACH406 (b) 1ALX349 (c) 1ASD541 (d) 1ACH406 (e) 1ALX349 (f) 1ASD541

Fig. 4. Cropped number plate (top) and synthetically modelled (bottom) of height 13 ((a)-(c)) and 7 pixels ((d)-(f))

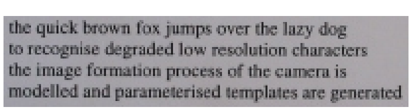
The position of the plates in each frame is determined manually and three different example plates for each distance are shown in Fig. 4, with the modelled character templates at the bottom of each image. The results are summarised in Tab. 2. Main reason for false recognitions is the increasing similarities of the characters with decreasing resolution. But such false classifications could be partially resolved by incorporating the number and letter scheme of the specific plate.

Character height	13 pixel	7 pixel
Plate accuracy	95.83%	50.69%
Character accuracy	99.41%	90.52%

Table 2. Recognition Results. Plate accuracy is the number of correct recognised plates in percentage of 720 possible frames (20 frames for each of the 36 number plates) and character accuracy is the number of correctly recognised characters, out of 5040 possible (4 numbers and 3 letters per plate and frame).

4.3. Text Recognition

To demonstrate the ability of our approach for low-resolution text recognition, we captured a text of font size 12 using a resolution of 640×480 at a distances of 30cm. The resulting average characters are 5 pixels in height as shown in Fig. 5(top).



the quick brown fox iumps over the lazy dog
to recognise degraded low resolution characters
rhe image formation process of the camera is
modelled and parameterised templates are generated

Fig. 5. Image (top) and recognised transcript (bottom).

The resulting transcript is shown in Fig. 5(bottom), 27 out of 29 words are recognised correctly. The two errors are 'iumps' instead of 'jumps', and 'rhe' instead of 'the'. Note that we do not make use of a dictionary, which is an optional step that would increase the recognition performance of our approach even further especially in low-resolution.

5. CONCLUSION

We proposed a new method for low-resolution character recognition by modelling the down-sampling process of the optical camera system. Using this model a large number of different low-resolution templates for each character of the alphabet are generated which are then used for alphanumeric text recognition. This approach does not require an initial character segmentation. Furthermore all images and videos used within the experiments section are compressed.

We avoid using image enhancement or super-resolution techniques, as these may create artefacts that decrease the recognition rates of OCR. Instead we recognise characters using their low-resolution appearance only. And even though a large number of templates is used for each word, our approach is still practicable as the size of the low-resolution input image is quite small and so are the templates used.

The proposed method is best suited for applications in which the font type is known beforehand like number plate recognition. Experiments showed that recognition rates are highest when using the correct font for recognition.

6. REFERENCES

[1] F.R. Jenkins S.V. Rice and T.A. Nartker, "The fifth annual test of ocr accuracy," Tech. Rep., UNLV Inf. Sci. Res. Inst., 1996.

[2] K.V. Suresh, G.M. Kumar, and A.N. Rajagopalan, "Superresolution of license plates in real traffic videos," *ITS*, vol. 8, no. 2, pp. 321–331, April 2007.

[3] Charles Jacobs, Patrice Y. Simard, Paul Viola, and James Rinker, "Text recognition of low-resolution document images," in *ICDAR*. 2005, pp. 695–699, IEEE Computer Society.

[4] O.D. Trier, A.K. Jain, and T. Taxt, "Feature-extraction methods for character-recognition: A survey," *PR*, vol. 29, no. 4, pp. 641–662, April 1996.

[5] G. Dalley, B. Freeman, and J. Marks, "Single-frame text superresolution: a bayesian approach," in *ICIP04*, 2004, pp. V: 3295–3298.

[6] P.D. Thouin and C.I. Chang, "A method for restoration of lowresolution document images," *IJDAR*, vol. 2, no. 4, pp. 200–210, 1999.

[7] J. Sun, Y. Hotta, Y. Katsuyama, and S. Naoi, "Camera based degraded text recognition using grayscale feature," in *ICDAR05*, 2005, pp. I: 182–186.

[8] Hsien-Huang P. Wu, Hung-Hsiang Chen, Ruei-Jan Wu, and Day-Fann Shen, "License plate extraction in low resolution video," in *ICPR*. 2006, pp. 824–827, IEEE Comp. Society.

[9] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *ITS*, vol. 7, no. 3, pp. 377–392, September 2006.

[10] M. Elad and A. Feuer, "Restoration of single super-resolution image from several blurred," *IEEE Trans. on Image Processing*, pp. 1646–1658, Dec 1997.