卒 業 研 究 報 告 書

題　　　目

Reduction of Computational Time in Scene Character Recognition with Reference Point

研 究 グ ル ー プ　　第3グループ

指 導 教 員　　岩村　雅一 准教授

平 成 24 年（　2012 年）度 卒 業

（No.　1090107041　）　　松田 崇宏

大 阪 府 立 大 学 工 学 部 知 能 情 報 工 学 科

リファレンスポイントを用いた情景内文字認識の高速化

第3グループ　松田崇宏

1. はじめに

近年カメラ付き携帯端末が発達していることから，情景内の文字を撮影することで，文字の翻訳結果や関連した情報をユーザーに提示するサービスが考えられる．情景内文字認識技術の一つとして，Iwamuraらの手法[1]がある．Iwamuraらの手法では局所特徴が使われており，異なる角度から撮影された文字や，複雑な背景上にある文字でも認識が可能であることが利点である．しかし処理速度は速いとは言えず，計算機上でも1〜2fps程度でしか動作しない．そこで本論文では，Iwamuraらの手法において，認識処理に使われる特徴点数を削減し，処理時間を削減する手法を提案する．

2. Iwamuraらの手法

Iwamuraらの手法は事例ベース認識であり，学習と認識の大きく2つのステップに分けることができる．そして更に認識処理は，特徴抽出，特徴のマッチング，文字認識と文字領域の推定の3つのステップに分けることができる．

学習時の処理について説明する．まず学習用の文字画像の局所領域から局所特徴を抽出する．そして，それらの局所特徴にその文字の文字IDを持たせて，データベースに登録する．

認識時の処理について説明する．認識対象である質問画像が与えられた時，まず学習時と同様に，質問画像の局所領域から局所特徴を抽出する．次に，質問画像から抽出された局所特徴と，データベースに登録された局所特徴とのマッチングを行う．マッチングには高速化のために，佐藤らの近似最近傍探索手法[2]を使用する．そして，質問画像から抽出された局所特徴に一番類似度が高い局所特徴の文字IDを持たせる．最後に，文字IDごとに特徴点の配置を調べる．その配置とデータベースの特徴点の配置がほぼ一致した場合，その文字を認識結果とする．この時，特徴点の配置から文字領域の推定も同時に行う．

3. 提案手法

Iwamuraらの手法において，抽出された局所特徴の中には文字から抽出された局所特徴だけでなく，背景や誤検出された局所特徴が含まれる．このような認識に寄与しない特徴点は，認識時の最終ステップである文字の認識と文字領域の推定処理において冗長な処理を引き起こし，処理時間を増大させる原因となる．そこで特徴点のマッチングを行った後，認識に寄与しない特徴点を削減する処理を追加し，処理速度を向上させる手法を提案する．

学習時にあらかじめ局所特徴を抽出した後，局所特徴だけでなく各特徴点から文字の中心，すなわち画像の中心までの位置もデータベースに登録する．そして認識時に特徴のマッチングの後，各特徴点からあらかじめ登録してある位置（リファレンスポイント，以下RP）を求める．もし特徴点が同一の文字領域から正しく得られた場合，それらのRPは一箇所に密集する．逆に，背景や誤検出によって得られた特徴点のRPは一箇所に密集しない．そこで，それぞれの特徴点のRPに対して，その近傍に一定数以上のRPがあった場合，それらの特徴点を全て残す．逆に，その近傍に一定数以上のRPがなかった場合，その特徴点は捨てる．この処理を全ての特徴点に対して行うことで，最終的に認識に寄与しない特徴点だけを削減することができる．

表1: (1)Iwamuraらの手法と(2)Iwamuraらの手法を改良した提案手法の比較

	(1)	(2)
認識率	55%	55%
特徴点数	14370	557
RPの計算時間	-	3ms
認識時の最終ステップでの計算時間	30ms	7ms
処理全体の計算時間	414ms	391ms

4. 実験

Iwamuraらの手法と，Iwamuraらの手法を改良した提案手法の比較実験を行った．学習画像には，MSゴシックのひらがな，カタカナ，英数字，常用漢字の合計2150文字を使用した．質問画像には，A4サイズの紙にMSゴシックのひらがな，カタカナ，アルファベットを20文字ずつ印刷したものを6枚用意した．更にそれらを異なる角度から10枚ずつ撮影した計60枚を使用した．使用した計算機のCPUはcore i5 2.3GHz，メモリは6GBである．

結果を表1に示す．認識時の最終ステップとは，特徴点の配置を用いて文字の認識と文字領域の推定を行う処理のことである．結果より，認識率を維持しつつ特徴点を大幅に削減することができていることがわかる．これにより，計算コストが削減でき，処理速度を向上することができたと言える．

5. まとめと今後の課題

本稿では，Iwamuraらの手法に，リファレンスポイントを用いて認識に寄与しない特徴点を削減し，処理速度を向上させる手法を提案した．今回はRPの近傍点とみなす距離と個数の閾値を固定値としたが，今後の課題としては，これらの閾値を動的に変化させることが考えられる．

参考文献

[1] M. Iwamura, T. Kobayashi and K. Kise: "Recognition of multiple characters in a scene image using arrangement of local features", Proc. of 11th International Conference on Document Analysis and Recognition (ICDAR 2011), pp. 1409–1413 (2011).

[2] 佐藤, 岩村, 黄瀬: "空間インデクシングに基づく距離推定を用いた高速かつ省メモリな近似最近傍探索手法", 信学技報，PRMU2012-142, **112**, 441, pp. 73–78 (2013).

Contents

List of Figures

List of Tables

1 Introduction

These days more and more people have begun to use a smart phone, and there are a lot of applications for smart phones. "Utsushite Honyaku" is one of them and is used for scene character recognition [1]. This application can recognize words and characters in a captured image and translate them automatically. By using this application, the users can avoid typing the text, and they can know the meanings of the words and characters. In such an application, the accuracy and processing time are very important. Therefore we have to deal with several requirements. First, it has to be robust to perspective distortions because the users may take character images from various directions. Second, it also has to be robust to text layout changes. Sometimes texts are not on a straight line but on a circle. In addition, some texts in scene images are on a colorful background. Recognizing characters on a complex background is a very difficult problem and such a problem often causes bad recognition results. At last, the system must run in real time not to make the users wait so long.

Iwamura et al. proposed a method which solves the problems above [2]. The method consists of 3 steps. First, local features are extracted from local parts of a character image. Second, the extracted features are matched with ones stored in a database. Finally, characters are detected and recognized by using arrangements of the matched local features. The local feature descriptor is robust to perspective distortions and it can recognize characters on a complex background. Therefore the method obtains a high recognition rate for scene character recognition. However the processing time is a big problem. Although their method runs at around 1 fps in high spec machines, it is very difficult to run in smart phones. Therefore, we need to reduce the computational time.

In this paper, we consider to reduce the computational time. After matching features, matched features include a lot of redundant features. In order to reduce them, we employ *reference point* (in short *RP*) [3]. By using RP, we can know how correct the features are. Therefore we discard features with lower reliability. Then, because of the decrease of features, we can reduce the computational time in character recognition process except feature extraction and matching.

2 Character Recognition Method Using Local Features

In this section, we introduce a method of Iwamura et al [2]. Figure 2.1 shows an overview of the method. As already described in the previous section, the recognition process is mainly divided into 3 steps. We describe the detail of each step below.

2.1 Feature Extraction

In the feature extraction step, we detect feature points and describe feature vectors from a query image by using SIFT. To detect each feature point, we compare a pixel value with the neighbor pixels. If the pixel value is extremal, the pixel is regarded as a feature point. After detecting feature points, we describe feature vectors from each feature point. 128-dimensional feature vectors are described by considering the pixel values around each feature point.

2.2 Matching Features

Then, each feature vector is matched with feature vectors stored in a database. Euclidean distance is used to compute the distance between two vectors and the pair with the shortest distance is the matching result. After the process, each feature point has the label that the feature point is matched to which feature point of which reference character. The position of the matched feature point is used to recognize characters in the following process. In the matching process, we used an approximate nearest neighbor search method proposed by Sato et al. [4].

2.3 Character Recognition

The final step is the character recognition. Figure 2.2 shows an overview of the process. We use an assumption that a character region in the query image has the large number of feature points of the corresponding character. Therefore, we search for such a region by looking into a close region around each feature point. However, this process shows only the vague position of the characters. Thus we use the information of matched feature points to detect the exact region of each character. Precisely, we find three pairs of feature points from the vaguely revealed character region. The feature points must have the same character ID as the character of the region. Then we calculate an affine transformation matrix from the positions of the feature points. By projecting the reference character region to the query image with the matrix, we can detect the exact region of the character. Besides, we improve the detection accuracy by applying RANSAC to the feature points in the character region [5]. By applying RANSAC the probability of mis-detection decreases so much, thus we can regard that the character recognition process finishes at the same time. According to the result of RANSAC, each recognized character has a score based on the confidence ratio to the recognition result.

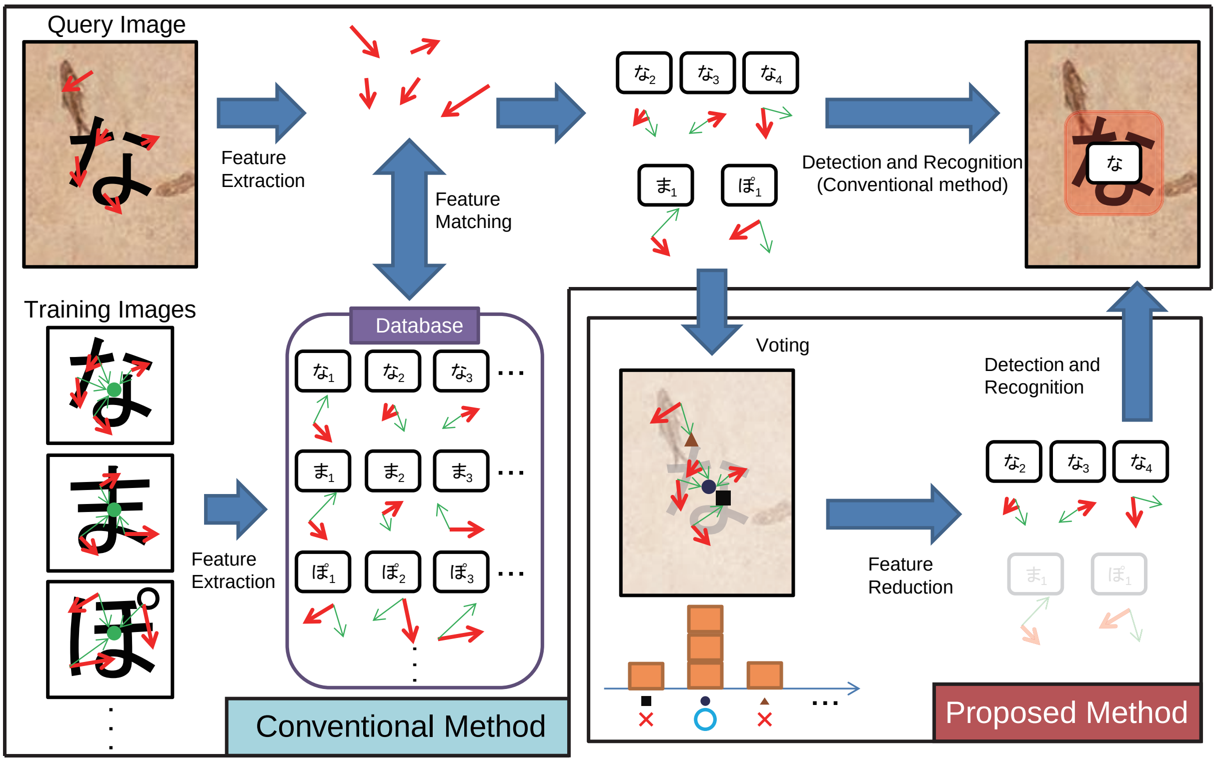

Figure 2.1 An overview of the conventional method proposed by Iwamura et al. and the proposed method.

In the conventional method, in learning process, local features are extracted from learning images and stored in the database. In recognition process, local features are extracted from a query image. Next, each feature is matched with one in the database. After that, by using the matched features, the characters are recognized and the character regions are segmented at the same time. In the proposed method, in learning process, also an orientation and distance from each feature point to the center of the character (a reference point) is stored. In the learning process, after matching process, reference point of each feature is estimated by the orientation and distance, and we cast a vote for the reference point. Then features whose reference point has a low score are discarded.

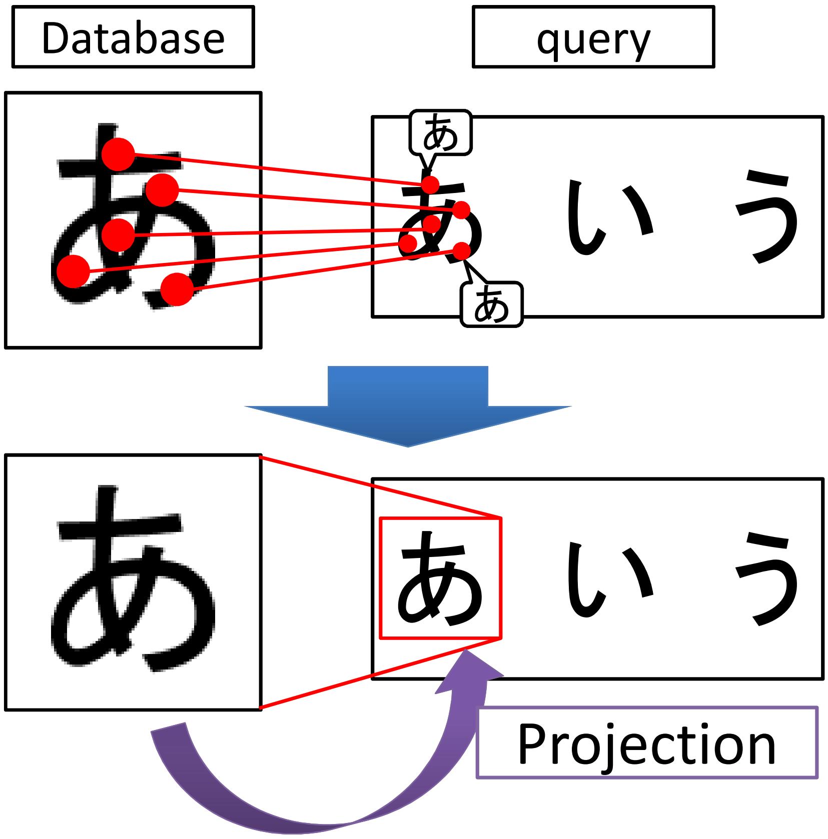

Figure 2.2 A detail of the character recognition process.
First, local features extracted from a query image are matched with those of reference images. Then the character region is projected to the query image by using mapping matrix calculated from the arrangements of matched local features.

3 Proposed Method

Extracted and matched features include a lot of redundant features in the recognition process. The redundant features are extracted from the background and parts of characters that are similar to other parts of characters. Although we can avoid mis-detection caused by them with RANSAC, we cannot avoid the increase of the computational time. Therefore we consider that if we can discard the redundant features, we can reduce the computational time. In this section, we propose a method to reduce the redundant features with an idea of *reference point* (in short *RP*) [3].

Figure 2.1 shows an overview of the proposed method with RP. As shown in the figure, in the learning process we extracted not only feature vectors but also orientations and distances from each feature point to the center of the reference character. We define a point which is located in the orientation and distance from the feature point as an RP. If features are correctly extracted and matched, the RP is expected to correspond to the center of each reference character. However, in many cases, there are falsely extracted and matched features. Now, votes are casted for RPs features.The number of votes of an RP is equivalent to the number of RPs in the close place. If a feature is correctly extracted and matched, the number of votes of an RP becomes high. If not, it becomes low. Then, if the number of votes of an RP is fewer than a threshold t, we discard only the feature.

However, in a real condition, even if features are correctly extracted and matched, they are not often completely concentrated to one point but scattered around there. Therefore we regard RPs within radius r as the same RPs.

4 Experiment

In order to evaluate the effectiveness of the proposed method, we conduct an experiment with different value of t and r. In this section, we show three different types of results from running the system. One is that how much features can be decreased with RP. The second result shows the computational time. Finally we show the recognition rate.

4.1 Experimental Condition

As the reference images, we employed 71 categories of Hiragana and Katakana respectively and 1,945 categories of Kanji (Chinese character) in MS Gothic font with the same condition as [2]. Some pairs of alphabet characters are treated as the same class (for example I and l are the same class) since they are in the relationship of similar transformation. In the experiment, we categorized the alphabets with the same manner as [6]. We used a computer whose CPU was core i5 2.3GHz and the memory was 6GB. Figure 4.1 shows the 6 query images. The resolution of the images was 640×480. We changed t from 2 to 10 and r from 0 to 99.

4.2 Results

Figure 4.2 shows the numbers of features. The figure shows that a large number of features were discarded by the proposed method. However, the discarded features may include not only redundant features but also valid features.

Figure 4.3 shows the computational time in character recognition process except for feature extraction and matching. It shows that the smaller t became and the larger r became, the higher the computational time was. Therefore the computational time with RP decreased in any case.

Figure 4.4 and 4.5 show the recognition rates. As shown in Figure 4.4, we know that all recognition rates were the same with r larger than 15 and t was in the range of 2 to 5. As shown in Figure 4.5, when t varied from 5 to 10, we know that the larger t was, the lower the recognition rate was. Therefore we know that the best t was 5.

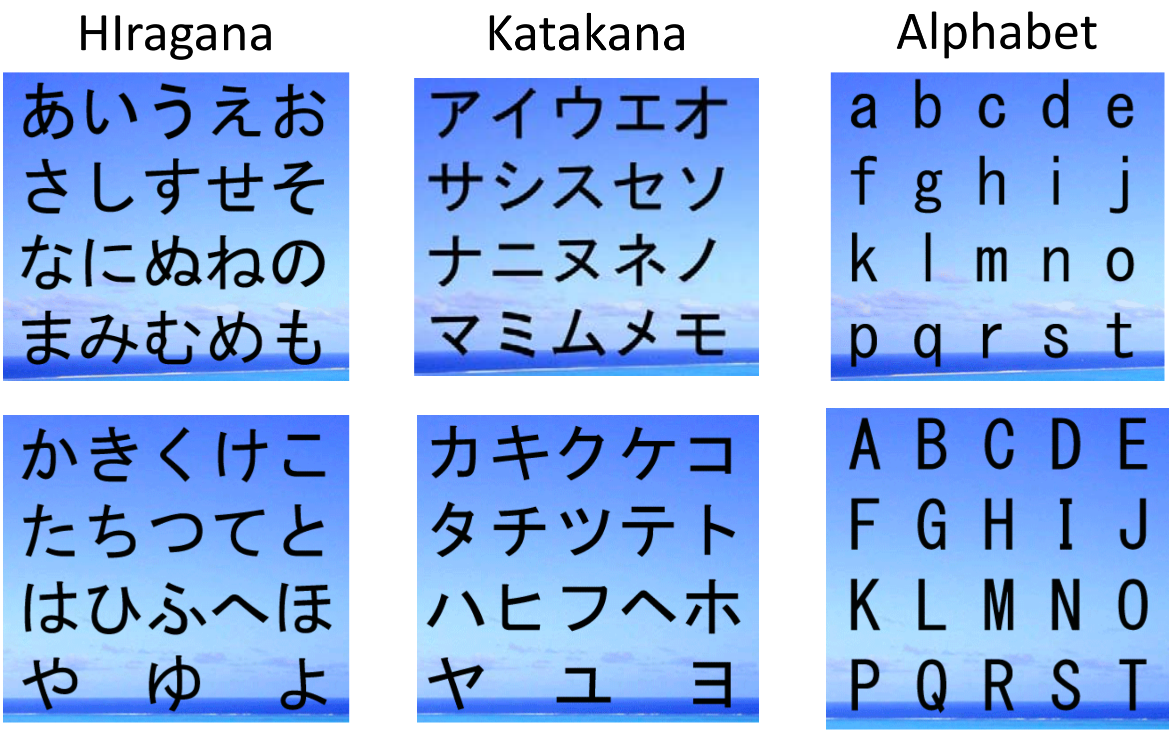

Figure 4.1 A part of query images.

From these experiments, we found that when t was equivalent to 5 and r was equivalent to 16, the performance of the proposed method was the best. Now we also show the results of conventional method and proposed method with the best parameters in Table 4.2. This table shows that with the same recognition rates the number of features was decreased by about 1/30. And decrease of the number of features reduced the computational time by about 1/4. This result shows that a lot of features and the computational time in the recognition process except feature extraction and matching with keeping recognition rate could be reduced. But we cannot say that we reduced the total computational time substantially.

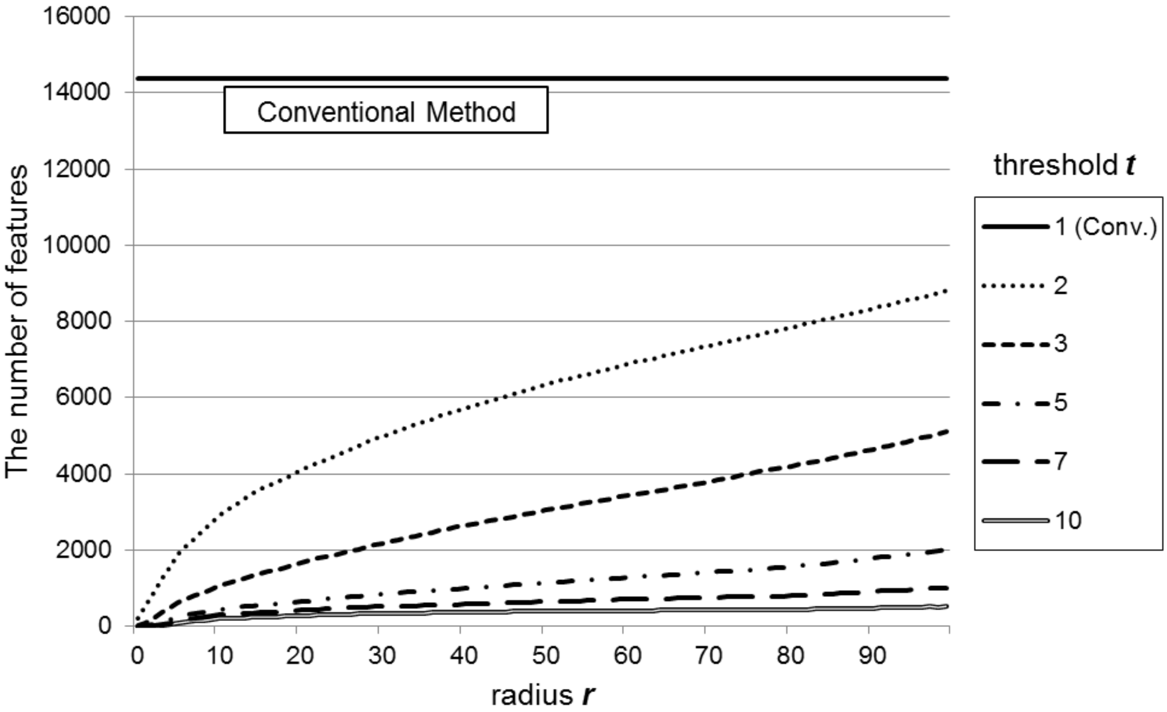

Figure 4.2 The number of features.

Table 4.1 The results of the conventional method and the proposed method with the best parameters (t=5 and r=16).
Computational time shows the time in the recognition process except feature extraction and matching and includes time of RP in Prop.
Total computational time shows the time in the recognition process.

	Conv.	Prop.
Recognition Rate	55%	55%
Number of Features	14370	557
Computational Time	30ms	10ms
Total Computational Time	414ms	391ms

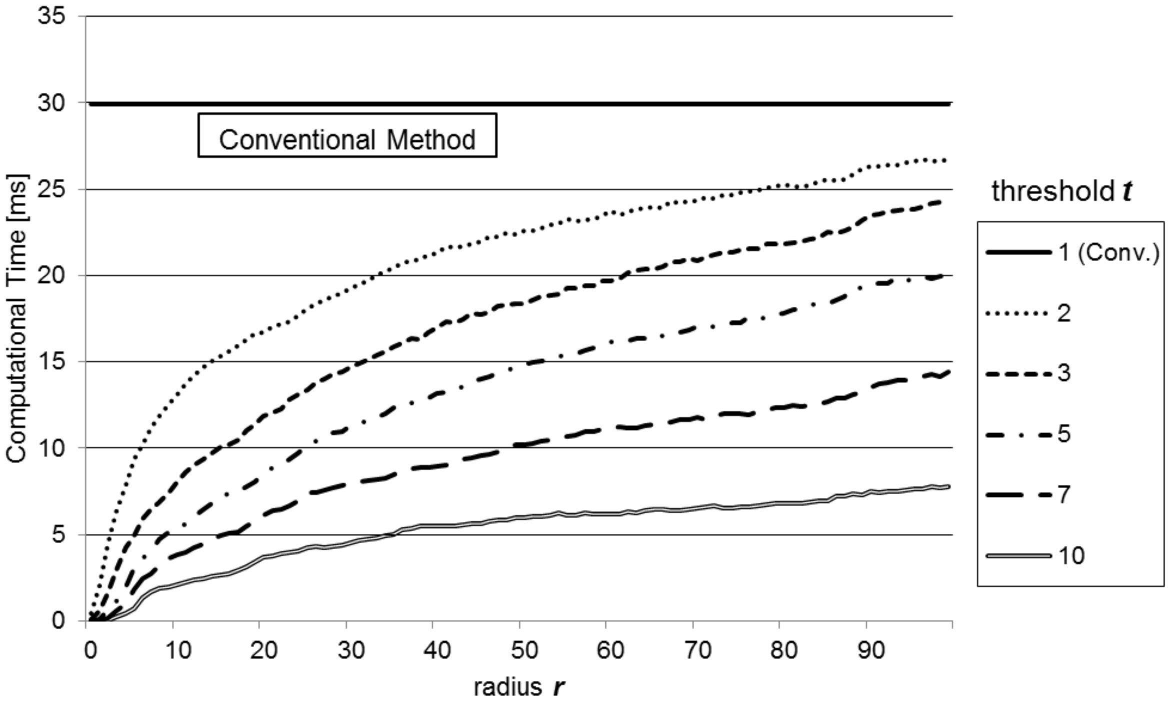

Figure 4.3 The computational time in character recognition process except feature extraction and matching.

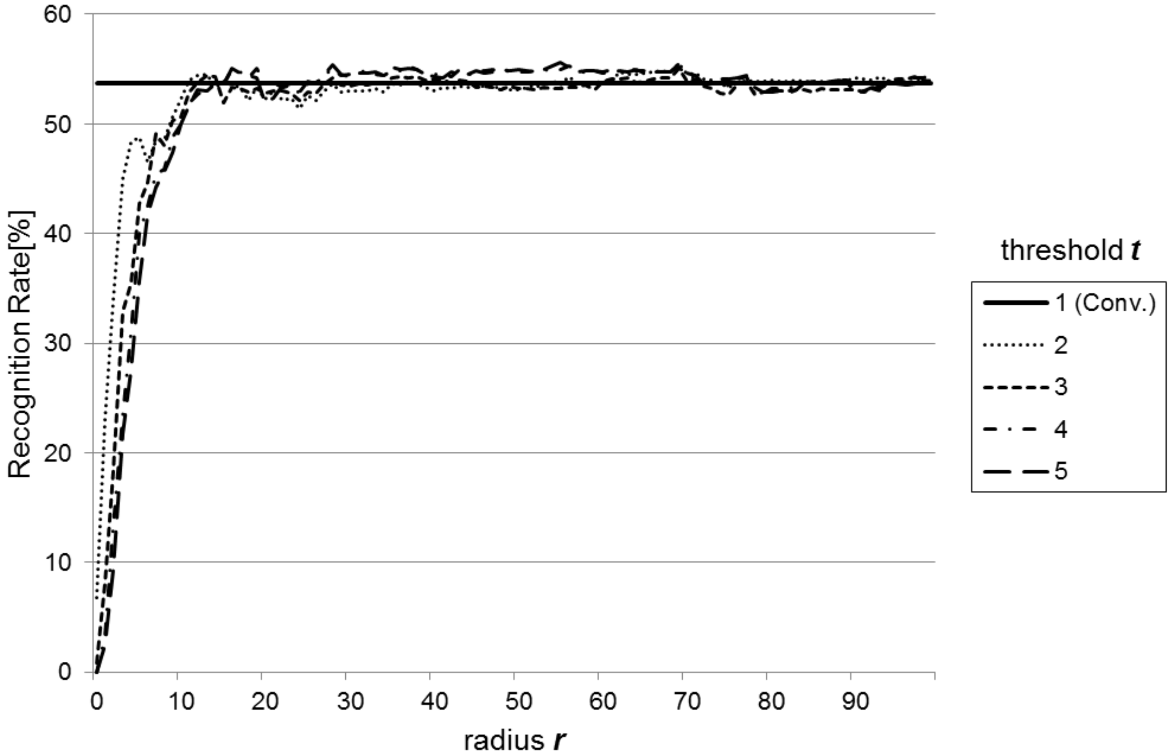

Figure 4.4 The recognition rates (t = 2 to 5).

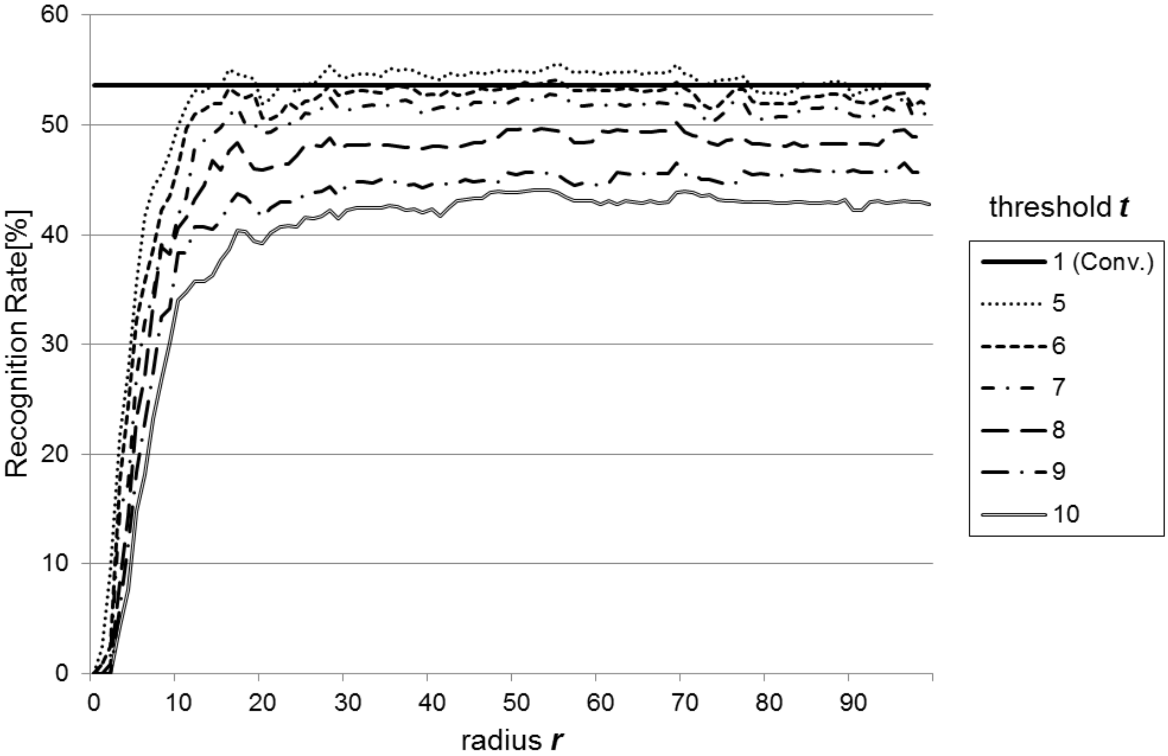

Figure 4.5 The recognition rates ($t = 5$ to 10).

5 Conclusion

In this paper, we proposed the method to reduce a computational time in a method proposed by Iwamura et al. By using the idea of reference point, we reduced the redundant features. As a result, the computational time with RP decrease by about 1/4 in comparison with the one without RP. Our future works are to vary t and r dynamically and experiment the proposed method to various query images. And we also have to improve other processes to reduce the total processing time.

謝辞

本研究の進行にあたり，研究内容について直接のご指導を頂いたほか，論文作成やプレゼンテーションなど，様々な面で御助力を頂きました岩村雅一准教授，小林拓也氏に深い謝意を表します．また，日頃より多くの御指摘，御助言をして下さった黄瀬浩一教授，内海ゆづ子助教，岩田基助教に深く感謝致します．また，様々な面において御支援，御助力を下さった知能メディア処理研究室の皆様に感謝致します．

2013 年 3 月 8 日

参考文献

[1] http://www.nttdocomo.co.jp/service/information/utsushite_honyaku/.

[2] M. Iwamura, T. Kobayashi, and K. Kise, “Recognition of multiple characters in a scene image using arrangement of local features,” Proc. of 11th International Conference on Document Analysis and Recognition (ICDAR 2011), pp.1409–1413, Sep. 2011.

[3] K. Martin and K. Koichi, “Using a reference point for local configuration of sift-like features for object recognition with serious background clutter,” IPSJ Transactions on Computer Vision and Applications（CVA）, vol.3, pp.110–121, Dec. 2011.

[4] T. Sato， M. Iwamura, and K. Kise, “Fast and memory efficient approximate nearest neighbor search with distance estimation based on space indexing,” IEICE， PRMU2012-142, vol.112, no.441, pp.73–78, Feb. 2013.

[5] 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, vol.24, no.6, pp.381–395, Jun. 1981.

[6] M. Iwamura, T. Tsuji, and K. Kise, “Memory-based recognition of camera-captured characters,” Proc. of the 9th IAPR International Workshop on Document Analysis Systems (DAS2010), pp.89–96, Jun. 2010.