

SISTEM PENGENALAN KARAKTER BRAILLE DENGAN METODE IMAGE CORRELATION

I Made Wirawan Dwi Putra¹, Retno Novi Dayawati², Tjokorda Agung Budi Wirayuda³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Abstrak Braille adalah sistem penulisan yang memungkinkan penyandang tuna netra dan orang yang tidak dapat melihat secara normal untuk membaca dan menulis dengan cara meraba tulisan tersebut. Ditemukan oleh Louis Braille (1809- 1852), seorang guru berkebangsaan Prancis yang mengajar penyandang tuna netra. Braille terdiri dari pola enam titik yang tersusun pada blok dengan konfigurasi 3 x 2. Setiap blok mewakili sebuah huruf, angka atau tanda baca. Pada tugas akhir kali ini dibangun aplikasi yang bisa mengenali teks karakter Braille dengan menggunakan algoritma Template Matching dengan metode Image Correlation. Metode ini melakukan pengenalan karakter dengan cara menggerakkan template ke setiap posisi dalam sebuah image dan mengevaluasi tingkat kesamaan dari setiap posisi yang telah dilalui. Pembangunan sistem ini dilakukan dengan menggunakan Microsoft Visual Studio 2005. Dari hasil pengujian didapat pola semakin kecil nilai SAD maka jumlah karakter Braille yang ditemukan akan semakin mendekati jumlah karakter Braille sebenarnya.

Kata Kunci : Braille, Template Matching, Image Correlation.

Abstract

Abstract Braille is writing system which enables blind and partially sighted people to read and write through touch. It was invented by Louis Braille (1809-1852), a French teacher of the blind. It consists of patterns of raised dots arranged in cells of up to six dots in a 3 x 2 configuration. Each cell represents a letter, numeral or punctuation mark. In this final assignment is developed application to recognize Braille character text with Template Matching algorithm using Image Correlation method. Image Correlation conduct character recognition by comparing image patterns with image template. This method conduct character recognition by moving template to every position in a image and evaluate equality level from each position which have been passed by. Character recognition accuration using this method depending on image condition and template size. System development built using Microsoft Visual Studio 2005. From the test we get the pattern if smaller SAD value resulting sum of Braille character found more closed with sum of real character.

Keywords : Braille, Template Matching, Image Correlation.

1. Pendahuluan

1.1 Latar belakang

Huruf Braille belakangan ini semakin populer dipergunakan oleh penyandang tuna-netra. Hal ini disebabkan semakin tingginya kesadaran penyandang tuna netra untuk mengenyam pendidikan formal. Namun sering kita jumpai orang yang awam huruf Braille terutama orang yang bukan penyandang tuna-netra ingin mengetahui atau mempelajari teks-teks Braille. Salah satu caranya adalah dengan menerjemahkan teks-teks Braille kedalam bentuk teks yang umum dimengerti yakni teks latin.

Permasalahan yang muncul adalah bagaimana cara yang mudah dan efektif untuk menerjemahkan tulisan-tulisan braille sehingga tulisan-tulisan braille dapat dengan mudah diterjemahkan. Sejauh ini tulisan Braille diterjemahkan dengan cara mengetik ulang tulisan-tulisan braille tersebut kedalam teks latin. Hal ini kurang efektif karena membutuhkan waktu yang lama untuk menerjemahkan tulisan braille tersebut, selain itu juga terjadi kesulitan bagi orang yang tidak mengerti tulisan braille untuk menerjemahkan teks braille. Untuk itu, maka diperlukan suatu aplikasi yang dapat menerjemahkan tulisan braille ke dalam teks latin dengan cara yang lebih efektif dan efisien.

Metode yang digunakan dalam penyelesaian masalah pengenalan karakter Braille tersebut adalah *Image Correlation*, Teknik untuk menemukan pola tertentu dari *image* dengan cara menggeser *template*. Pemilihan metode *Image Correlation* dikarenakan metode ini relatif sederhana karena cuma membandingkan nilai kesamaan pada tiap-tiap titik pada citra *template* dan citra teks Braille.

1.2 Perumusan masalah

Permasalahan yang akan dibahas dalam Tugas Akhir ini adalah

1. Bagaimana membuat perangkat lunak yang bisa digunakan untuk mengkonversi karakter huruf braille menjadi karakter huruf latin.
2. Bagaimana mengimplementasikan algoritma *Template Matching* dengan metode *Image Correlation* pada perangkat lunak untuk mengkonversi karakter huruf braille menjadi karater huruf latin.
3. Bagaimana akurasi pengenalan karakter huruf braille pada *file image* menggunakan algoritma *Template Matching* dengan metode *Image Correlation*.

Berikut merupakan batasan dari perumusan masalah diatas :

1. *Input* dari perangkat lunak adalah *file image* (*.bmp) hasil *scanning* text karakter braille.
2. Naskah braille hanya ditulis dengan menggunakan *reglet*.
3. Naskah braille yang di gunakan adalah *single-sided* (satu sisi).
4. Ukuran kertas braille yang di gunakan adalah standar A4 atau *letter*.
5. Tulisan braille terdiri dari kombinasi 6 (enam) titik.

1.3 Tujuan

Maksud dan tujuan pembuatan Tugas Akhir ini adalah :

1. Menyediakan sarana pengkonversi *huruf braille* menjadi huruf latin.
2. Memanfaatkan algoritma *Template Matching* dengan metode *Image Correlation* pada perangkat lunak pengkonversi karakter *huruf braille*.
3. Menguji dan menganalisa unjuk kerja algoritma *Template Matching* dengan metode *Image Correlation* dari segi akurasi / tingkat kesalahan.

1.4 Metodologi Penyelesaian Masalah

1. Studi pustaka
Mencari referensi yang berhubungan dengan algoritma *Template Matching* dengan metode *Image Correlation* dan hal-hal lain yang berkaitan dengan judul pada Tugas Akhir ini.
2. Analisa dan Perancangan Perangkat Lunak
Pada tahap ini dilakukan Analisa dan Perancangan Perangkat Lunak seperti menentukan kebutuhan perangkat lunak serta membuat diagram analisa dan perancangan perangkat lunak.
3. Implementasi dan Pengujian
Implementasi berdasarkan desain yang telah dibuat menggunakan Microsoft Visual Studio 2005.
4. Analisa Fungsi Hasil Implementasi
Aplikasi yang telah selesai diimplementasikan akan dievaluasi. Akan dilakukan pengujian dari sistem yang telah dibangun pada tahap implementasi kemudian menganalisa tingkat kesalahan dari proses pengenalan karakter huruf Braille pada *file image*, serta akan dilakukan koreksi dan penyempurnaan program apabila diperlukan.
5. Pengambilan kesimpulan dan penyusunan Laporan Tugas Akhir.

1.5 Sistematika Penulisan Laporan

Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

- 1 **Pendahuluan**
Berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metode penyelesaian masalah dan sistematika penulisan laporan.
- 2 **Landasan Teori**
Berisi teori yang mendasari penyusunan dan pembuatan Tugas Akhir ini.
- 3 **Analisis dan Perancangan**
Berisi tentang analisis terhadap aplikasi yang akan dibangun, serta model perancangan yang dibutuhkan untuk mengimplementasikan aplikasi yang dibuat.
- 4 **Implementasi dan Pengujian**
Berisi implementasi dari analisa dan perancangan aplikasi ke dalam bentuk perangkat lunak serta hasil pengujian terhadap aplikasi yang dibuat.
- 5 **Penutup**
Berisi simpulan yang diperoleh dari implementasi dan pengujian aplikasi serta saran terhadap pengembangan aplikasi lebih lanjut.

5. Penutup

5.1 Kesimpulan

Dari hasil analisa Tugas Akhir ini, penulis dapat mengambil kesimpulan bahwa :

- a. Pengenalan karakter dengan kriteria SAD secara umum akan menghasilkan pola yakni semakin kecil nilai SAD maka jumlah karakter Braille yang ditemukan akan semakin mendekati jumlah karakter Braille sebenarnya. Ini dikarenakan semakin kecil nilai SAD maka semakin mirip citra yang dibandingkan sehingga keakuratan juga semakin baik.
- b. Semakin besar nilai threshold (SAD) maka semakin banyak karakter yang terdeteksi. Dikarenakan tingkat kemiripan semakin mengecil sehingga bagian citra teks Braille yang dianggap sama dengan template juga semakin banyak

5.2 Implementasi

Saran dari penulis untuk memperbaiki implementasi dari aplikasi ini :

Agar lebih akurat dalam penerapan *Template Matching* dengan metode *image correlations* pada aplikasi ini dibutuhkan :

- a. Proses pendeteksian kemiringan dan perotasian citra sebesar kemiringan yang dideteksi.
- b. Pemotongan (crop gambar) yang akurat sebelum dilakukan proses *image correlation*, dimana setiap huruf menempati satu blok yang diperlukan.
- c. Proses pergeseran template dilakukan "per blok" bukan per pixel sehingga dapat menanggulangi kesalahan pada text braille yang tidak rapat atau banyak ruang kosong. Blok disini artinya ukuran template yang sesuai dengan standar huruf Braille atau kelipatannya.
- d. Pergeseran "per blok" juga dapat menyederhanakan proses pergeseran template pada gambar target (proses looping semakin sedikit) sehingga waktu proses pengenalan dapat dipersingkat.

Telkom
University

Daftar Pustaka

- [1] Dharma, Eddy Muntina. *Diktat matakuliah Grafika Citra*, Bandung:STTTelkom, 2004.
- [2] Ahmad, Usman. 2005. "Pengolahan Citra Digital dan Teknik Pemrograman". Graha Ilmu. Yogyakarta.
- [3] Munir, Rinaldi. 2004 "*Pengolahan Citra Digital Dengan Pendekatan Algoritmik*". Informatika Bandung.
- [4] Chi Mai, Luong. "*Introduction to Computer Vision and Image Processing*";
<http://www.netnam.vn/unescocourse/computervision/computer.htm>.
- [5] Kosecka, Jana, "*More on Template Matching*";
<http://cat.xula.edu/thinker/perception/recognition/templates>.
- [6] <http://wikipedia.braille/> "*The Braille alphabet*"
- [7] <http://6dotbraille.com/> "*braille-punctuation, numbers, capital-letter , and alphabet*"
- [8] Blenkhorn, Paul & Evans, Gareth *Automated Braille Production from Word-Processed Documents, IEEE Transactions on Rehabilitation Engineering* Vol 9, No 1, pp 81 - 85 March 2001.
- [9] Safriady. Tugas Akhir: *Sistem Pengenalan Huruf Braille dengan Metode Determinasi Mesh*. Jurusan Teknik Informatika STT Telkom Bandung, 2006.
- [10] Sahru, Maskur. Tugas Akhir: *Perangkat Lunak Pengkonversi Accord Gitar Menjadi Gambar Posisi Tangan Pada Gitar Dengan Metode Image Correlation*. Jurusan Teknik Informatika STT Telkom Bandung, 2007.
- [11] http://en.wikipedia.org/wiki/Template_matching
- [12] Geri Achmadi, Editor Buku '*Sejarah Braille*', Alumni FMIPA Unpad