DACTYLOS

Technická dokumentácia

Verzia 1.0

Bc. Róbert Mokráš

25.1.2015

Obsah

[1 Program 3](#_Toc409964993)

[1.1 Požiadavky 3](#_Toc409964994)

[1.2 Spustenie programu 3](#_Toc409964995)

[2 Funkcie 3](#_Toc409964996)

[2.1 Hlavné Funkcie 3](#_Toc409964997)

[2.2 Get a Set funkcie 4](#_Toc409964998)

[2.3 Pomocné funkcie 4](#_Toc409964999)

# **1 Program**

## **1.1 Požiadavky**

Minimálne požiadavky pre správny beh programu:

* Nainštalovaný programovací jazyk C++
* Nainštalovaná knižnica OpenCV
* Nainštalovaný Operačný systém
* Dostatok voľnej pamäte RAM

## **1.2 Spustenie programu**

Program sa skladá z jedného hlavičkového súboru preprocessing.hpp a jedného zdrojového súboru preprocessing.cpp. Tieto súbory slúžia iba na vloženie do iných zdrojových súborov, nie sú spustiteľné samostatne.

# **2 Funkcie**

## **2.1 Hlavné Funkcie**

Všetky hlavné funkcie sú ako public.

* Program má 2 typy konštruktorov

Preprocessing(cv::Mat & image);  
Preprocessing(cv::Mat & image, inthorizontal\_resolution, intvertical\_resolution);

Prvý konštruktor má iba jeden argument a to vstupný obrázok, ktorý si upraví do čiernobielej podoby a dá implicitné rozlíšenie 500 Dpi. Druhý konštruktor pracuje na tom istom princípe, len má ešte dva dodatočné argumenty a to vertikálne a horizontálne rozlíšenie obrázka.

* Segmentation(cv::Mat & image);

Funkcia oddeľuje odtlačok prsta od pozadia. Vzniká v nej maska odtlačku prsta, ktorá sa uloží do premennej mask.

* OrientationMap(cv::Mat & image);

Funkcia vypočítava lokálne smery papilárnych línií a priemeruje ich pomocou Gaussového filtra. Vytvorí sa smerová mapa odtlačku prsta.

* GaborFilter(cv::Mat & image);

Funkcia upravuje odtlačok prsta. Vstupom do tejto funkcie je smerová mapa.

* Binarization(cv::Mat & image);

Funkcia pridelí jednotlivým pixelom hodnotu 0 alebo 255.

* ThinningImage(cv::Mat & image);

Funkcia zúži papilárne línie na hrúbku jeden pixel.

## **2.2 Get a Set funkcie**

Všetky get a set funkcie sú ako inlinepublic funkcie a pristupujú k súkromným premenným v triede.

* inlineintGet\_sizeofBlockGabor() – vráti veľkosť bloku pri Gáborovom filtri
* inlineintGet\_sizeofBlockOrientation() – vráti veľkosť bloku pri orientácii papilárnych línií
* inlinedoubleGet\_sigma() – vráti sigmu z Gáborovho filtra
* inlinedoubleGet\_lambda() – vráti lambdu z Gáborovho filtra
* inlinedoubleGet\_gamma() – vráti gammu z Gáborovho filtra
* inlinevoidSet\_sizeofBlockGabor(intsize) – nastaví veľkosť bloku pri Gáborovom filtri
* inlinevoidSet\_sizeofBlockOrientation(intsize) – nastaví veľkosť bloku pri orientácii papilárnych línií
* inlinevoidSet\_sigma(double sigma) – nastaví sigmu z Gáborovho filtra
* inlinevoidSet\_lambda(doublelambda) – nastaví lambdu z Gáborovho filtra
* inlinevoidSet\_gamma(gamma) – nastaví gammu z Gáborovho filtra

## **2.3Pomocné funkcie**

Všetky pomocné funkcie sú private

* intsupplement\_rows(int W, intW\_row);

funkcia dopočíta zvyšné riadky v odtlačku prsta

* intsupplement\_cols(int W, intW\_col);

funkcia dopočíta zvyšné stĺpce v odtlačku prsta

* voidwiener\_to\_mask(intblock\_size\_x, intblock\_size\_y);

funkcia aplikuje na masku odtlačku wienerov filter

* voidGuoHall(cv::Mat & image, intiteration);

funkcia, ktorá je potrebná pre výpočty zúženia papilárnych línií vo funkcii ThinningImage()