DOCUMENTATIE PROIECT PROGRAMARE PROCEDURALA

IMPORTANT: CITESTE README.TXT PENTRU A OBSERVA METODA SI ORDINEA DE COMILARE A MODULELOR (sau ruleaza in terminalul linux gcc message.c algorithms.c geometrics.c bitmap.c vector.c ocrdetect.c main.c -o main -lm)

PENTRU VERIFICAREA PROIECTULUI ESTE INDICAT(ABSOLUT INDICAT) SA SE FOLOSEASCA PATH-URI CE NU CONTIN SPATII LIBERE

SUNT 3 FISIERE DE COMPLETAT encrypt.txt , decrypt.txt , templates.txt. MAIN.C contine instructiuni referitoare la continutul acestora.

Salt la implementarile cerute de cerinta in EnuntProiect.pdf

Cerinta 1: algorithms.c dword xorshift32(dword seed) linia 21
genereaza numere pseudorandom in functie de seed

Cerinta 2: bitmap.c ntstatus readBitmapLinearized(const char * path, bitmap *bmp) linia 144
citeste imaginea din path si o stocheaza in structura bmp

Cerinta 3: bitmap.c ntstatus writeBitmapLinearized(const char *path, const bitmap *bmp) linia 223
scrie continutul structurii bmp pe disk la locatie

Cerinta 4: bitmap.c ntstatus encryptBitmap(const char *path, const char *dest_path, const char *key_path) linia 382
cripteaza imaginea de la locatie si salveaza continutul criptat in dest_path folosind cheia de la locatia key_path

Cerinta 5: bitmap.c ntstatus decryptBitmap(const char *path, const char *dest_path, const char *key_path) linia 464
decripteaza imaginea de la locatie si salveaza continutul decriptat in dest_path folosind cheia de la locatia key_path

Cerinta 6: bitmap.c ntstatus chiSquareTest(const char *path) linia 552
afiseaza testele chisquare pentru imaginea de la path

Cerinta 7: ocrdetect.c ntstatus matchTemplate(bitmap *image, bitmap *temp,templatedata *tdata, pvector vectorOfDetections, threshold thrsh, dword precision) linia 341
salveaza in vectorul de detectii vectorOfDetections toate pozitiile pentru care procesul de corelare indica o corelatie mai mare decat thrsh,

o detectie valida indica prezenta imaginii temp in imaginea image la o pozitie x,y.

Tdata e o structura optionala care contine informatii referitoare la sablonul temp

Cerinta 8: bitmap.c ntvoid drawBitmapRect(bitmap *bmp, rect rc, pixel pxl) linia 91
deseneaza un contur al dreptunghiului rc de culoarea pxl
se foloseste algoritmul lui Bressenham pentru o desenare dinamica

Cerinta 9: ocrdetect.c ntvoid sortTemplateDetections(pvector vec) linia 200
sorteaza elementele vectorului in functie de indicele de corelatie

Cerinta 10: ocrdetect.c ntvoid supressNonMaximums(pvector detectionVector, threshold thrsh) linia 204
elimina suprapunerile inavalide din vector

Cerinta 11: rezolvata in main.c si ocrdetect.c

DOCUMENTATIE PENTRU algorithms.c

*ubound inseamna upperbound si reprezinta numarul de elemente

ntstatus generateRandomSequence(dword seed, qword ubound, dword **random_sequence)
genereaza o secventa de numere pseudorandom

ntstatus inversePermutation(qword ubound, dword **permutation_sequence)
modifica vectorul permutation_sequence astfel incat acesta sa contina inversa permutarii

ntstatus generatePermutation(qword ubound,const dword *random_sequence, dword **permutation_sequence)
genereaza o permutare aleatoare folosind sirul de numere pseudorandom

ntstatus generatePermuttedSequence(dword *permutation_sequence,qword ubound, void *src_sequence,void **dest_sequence, dword szElement)
aplica o permutare asupra unui vector de marime ubound si salveaza noul sir obtinut in dest_sequence.

DOCUMENTATIE PENTRU bitmap.c

pixel rgb(color r, color g, color b)
returneaza o structura pixel ce indica o culoare rgb

ntvoid setBitmapPixelAt(bitmap *bmp, point coord, pixel pxl)
seteaza un pixel la o pozitie (I,j) in imaginea bmp

pixel getBitmapPixelAt(bitmap *bmp, point coord)
returneaza un pixel de la o pozitie (I,j)

ntvoid drawBitmapLine(bitmap *bmp, point startp, point endp, pixel pxl)
deseneaza o linie intre 2 perechi ordonate (I,j) , (k,m)

dword computeBitmapPadding(dword width)
returneaza numarul necesar de bytes pentru a completa o linie multipla de 4

ntstatus readBitmap(const char * path, bitmap *bmp)
citeste o imagine de la path in structura bmp

ntstatus freeBitmap(bitmap *bmp)
dealoca resursele alocate de imgine

pixel xorPixelWithPixel(pixel pixel1, pixel pixel2)
returneaza un pixel rezultat ca xor intre alti 2 pixeli

pixel xorPixelWithUint32(pixel pixel1, dword _data_2)
returneaza un pixel rezultat ca xor intre un pixel si un numar natural pozitiv

ntstatus copyBitmapHeader(bitmap *bmp1, bitmap *bmp2)
copiaza header-ul imaginii bmp1 in header-ul imaginii bmp2

ntstatus readBitmapEncryptionKey(const char *keypath, encryptionKey *key)
citeste cheile de decriptare de la path-ul keypath in structura key

ntstatus cvrtBitmapGrayscale(bitmap *bmp)
converteste o imagine in greyscale

DOCUMENTATIE PENTRU geometrics.c

ntvoid setPointCoordinates(point *coord, dword x, dword y)
seteaza coordonatele structurii coord cu perechea (x,y)

ntvoid setRectCoordinates(rect *rc, dword x0, dword y0, dword x1, dword y1)
seteaza pozitia unui dreptunghi dupa coltul din stanga sus si coltul din drepata jos

dword getRectArea(rect rc)
returneaza aria unui dreptughi

bool rectsAreOverlapping(rect r1, rect r2)
spune daca 2 dreptunghiuri se intersecteaza sau nu

rect getRectOverlapping(rect r1, rect r2)
returneaza un dreppthunghi ca rezultanta a intersectiei a alte 2 dreptughiuri

dword getRectOverlappingArea(rect r1, rect r2)
returneaza aria intersectiei dintre 2 dreptughiuri

double getRectOverlappingScore(rect r1, rect r2)

indica cat de mult se intersecteaza 2 dreptughiuri

DOCUMENTATIE PENTRU ocrdetect.c

ntstatus initOcrTransaction(ocrdata *ocrd)
initializeaza structura pentru o viitoare folosire

ntstatus addOcrInput(ocrdata *ocrd, char *image_path)
indica imaginea pe care se vor realiza corelaratiile.

ntstatus addOcrOutput(ocrdata *ocrd, char *image_path)
indica locul unde va fi salvata imaginea modificata(care contine detectiile colorate incadrate in drepthunghiuri)

ntstatus addOcrTrainingData(ocrdata *ocrd, char *image_path, char *semnification, pixel frameColor)
adauga un sablon impeuna cu detalile acestuia precum semnificatie (ce inseamna acel sablon in literal) + culoarea in care se va incadra detectia

ntstatus showOcrSemnificationsInConsole(ocrdata *ocrd, bool flag)
ii spune librariei daca vrei sa arati pe display semnificatiile sabloanelor gasite in imagine (in ordinea gasirii lor)

ntstatus saveOcrDetectionsWithImage(ocrdata *ocrd, bool flag)
ii spune librariei daca vrei /sau nu sa modifici poza incadrand detectia intr-un chenar

ntstatus setOcrDetectionThreshold(ocrdata *ocrd ,double thrsh)
modifica sensibilitatea detetctiei (intre -1 si 1)

ntstatus setOcrSupressionThreshold(ocrdata *ocrd, double thrsh)
indica sensibilitatea gasirii suprapunerilor (default 0.2f)

ntstatus setOcrPrecision(ocrdata *ocrd, dword precision)
seteaza rapiditatea si eficienta algoritmului.(intre 1 si 10, valori naturale)

ntstatus setOcrSaveDetectionsWithImage(ocrdata *ocrd, bool flag)
ii spune librariei daca vrei sa salvezi /sau nu poza modificata(care contine incadrarile sabloanelor gasite) pe disk

ntstatus beginOcrTransaction(ocrdata *ocrd)
incepe procesul de template matching

ntstatus disposeOcrTransaction(ocrdata *ocrd)
dealoca resursele alocate pentru template matching

ntstatus normxcorr(ocrdata *ocrd)
se ocupa global de procesul de template matching pentru fiecare sablon pe care il contine structura ocrdata.

DOCUMENTATIE PENTRU vector.c

bool vector_realloc(vector* vector, size_t new_count)
realoca dinamic un spatiu nou de memorie

pvector vector_create(size_t count_elements, size_t size_of_element)
returneaza un pointer catre un vector alocat dinamic

void vector_release(vector* vector)
dealoca resursele alocate pentru vector

pvoid vector_at(vector* vector, size_t index)
returneaza un element gasit la pozitia index in vector

pvoid vector_begin(vector* vector)
returneaza un pointer catre inceputul vectorului

pvoid vector_end(vector* vector)
returneaza un pointer catre sfarsitul vectorului

pvoid vector_next(vector* vector, pvoid i)

returneaza un pointer catre urmatorul element gasit dupa adresa i

size_t vector_count(const vector* vector)
returneaza numarul de elemente din vector

bool vector_erase(vector* vector, size_t index)
sterge un element gasit in vector la pozitia index
bool vector_append(pvector vector, const pvoid values, size_t count)
adauga un element in vector de dimensiune count

Proiect realizat de Reznicencu Bogdan

reznicencu.bogdan99@gmail.com
bogdan.reznicencu@my.fmi.unibuc.ro
https://stackoverflow.com/users/5056622/reznicencu-bogdan
https://github.com/ReznicencuBogdan