

Projekt:	Kontrola keramičnih ploščic
različica:	0.9
šifra dokumenta:	ZU-1-2005
datum:	7.1.2005

## Specifikacija zahtev uporabnika

### Splošni opis zahtev

Potrebno je izdelati aplikacijo, ki bo sposobna določiti ali keramična ploščica na sliki ustreza vnaprej določenim specifikacijam. Če ploščica na sliki temu ne ustreza, mora aplikacija uporabniku povedati zakaj ne ustreza. Sliko naj aplikacija avtomatsko naloži z vnaprej določene lokacije na disku in jo po nalaganju izbriše.

ZU001				
Aplikacija mora biti sposobna ugotoviti ali so dolžine stranic ploščice v tolerancah, ki jih določi operater				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 5	vir: vod.proizv.

ZU002				
Aplikacija mora biti sposobna ugotoviti, ali so koti med stranicami ploščice v tolerancah ki jih določi operater				
potrebnost: 5	stabilnost: 4	preverljivost: 5	prioriteta: 5	vir: vod.proizv.

ZU003				
Idealne mere ploščic in dovoljene tolerance vnese operater				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 5	vir: vod.proizv.

ZU004				
Vnos mer in toleranc naj bo enostaven				
potrebnost: 2	stabilnost: 4	preverljivost: 4	prioriteta: 2	vir: operater

ZU005				
Aplikacija mora delovati tudi, če je slika posneta pri slabši svetlobi				
potrebnost: 2	stabilnost: 4	preverljivost: 4	prioriteta: 2	vir: vod.proizv.
ZU006				
Aplikacija mora biti sposobna delovati 24 ur na dan				
potrebnost: 5	stabilnost: 5	preverljivost: 4	prioriteta: 4	vir: direktor
ZU007				
Aplikacija mora delovati na računalniku s procesorjem Pentium II, 266 Mhz in 128 MB pomnilnika				
potrebnost: 5	stabilnost: 3	preverljivost: 5	prioriteta: 5	vir: direktor
ZU008				
Aplikacija mora eno sliko obdelati najkasneje v 10 sekundah				
potrebnost: 5	stabilnost: 5	preverljivost: 4	prioriteta: 4	vir: direktor
ZU009				
Aplikacija mora biti enostavna za upravljanje				
potrebnost: 3	stabilnost: 5	preverljivost: 4	prioriteta: 3	vir: operater
ZU010				
Slike ploščic, ki jih mora aplikacija obdelati, bodo v formatu JPEG				
potrebnost: 5	stabilnost: 4	preverljivost: 5	prioriteta: 4	vir: operater
ZU011				
Aplikacija mora imeti vgrajen sistem pomoči, ki naj bo enostaven za uporabo				
potrebnost: 4	stabilnost: 5	preverljivost: 3	prioriteta: 1	vir: operater

Projekt:	Kontrola keramičnih ploščic
različica:	0.9
šifra dokumenta:	ZO-1-2005
datum:	25.1.2005

## Specifikacija zahtev s stališča strojne in programske opreme

### Splošni opis aplikacije

Aplikacija naj omogoča avtomatsko nalaganje slike z diska in njeno analizo. Aplikacija mora analizirati keramično ploščico na sliki in ugotoviti, če ustreza vnaprej določenim kriterijem. Operater (uporabnik) mora imeti možnost nastaviti te kriterije prek uporabniškega vmesnika aplikacije.

ZO001				
Nalaganje JPEG slike iz diska				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 5	vir: ZU010

ZO002				
Binarizacija slike				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 4	vir: ZU005

ZO003				
Iskanje robov ploščice na sliki				
potrebnost: 5	stabilnost: 4	preverljivost: 4	prioriteta: 5	vir: ZU001, ZU002

ZO004				
Iskanje premic, ki oklepajo ploščico na sliki				
potrebnost: 5	stabilnost: 4	preverljivost: 4	prioriteta: 5	vir: ZU001, ZU002

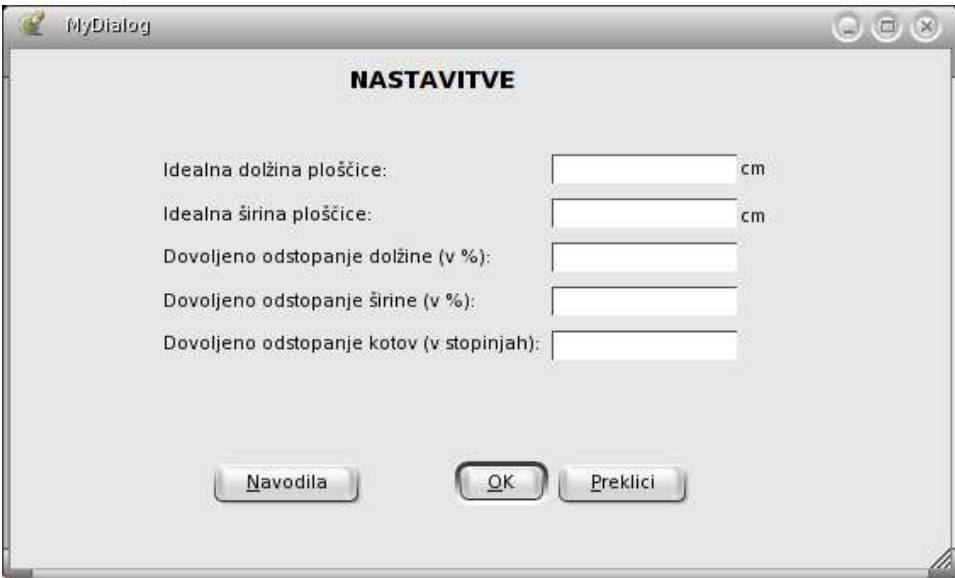
ZO005				
Iskanje vogalov ploščice				
potrebnost: 5	stabilnost: 4	preverljivost: 4	prioriteta: 5	vir: ZU001, ZU002

ZO006				
Izračun dolžine stranic ploščice				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 5	vir: ZU001, ZU002

ZO007				
Izračun kotov med ploščicami				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 5	vir: ZU001, ZU002

ZO008				
Preverjanje ali so dolžine stranic v dovoljenih mejah tolerance				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 4	vir: ZU001, ZU002

ZO009				
Preverjanje ali so koti med stranicami v dovoljenih mejah tolerance				
potrebnost: 5	stabilnost: 5	preverljivost: 5	prioriteta: 4	vir: ZU001, ZU002

ZO010				
Vnos idealnih mer in odstopanj za ploščice prek uporabniškega vmesnika				
				
potrebnost: 4	stabilnost: 5	preverljivost: 5	prioriteta: 3	vir: ZU003, ZU004, ZU009

ZO011

Glavno okno za obdelavo slik ploščic



potrebnost: 5

stabilnost: 4

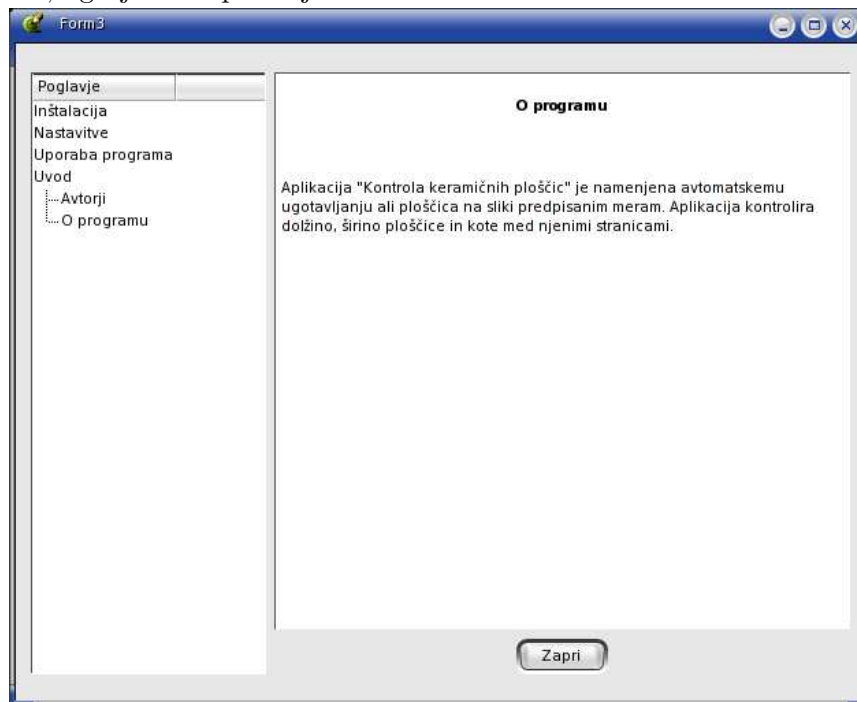
preverljivost: 5

prioriteta: 4

vir: ZU009

ZO012

Prikaz pomoči, vgrajene v aplikacijo



potrebnost: 3

stabilnost: 4

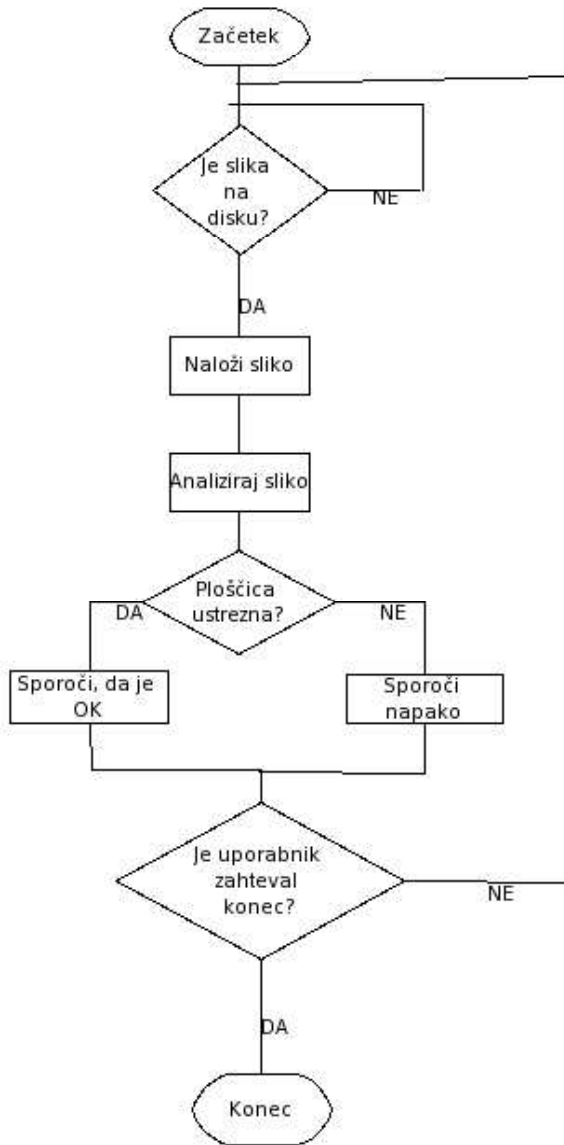
preverljivost: 5

prioriteta: 1

vir: ZU0011

ZO013				
Test aplikacije, če lahko deluje 24 ur na dan na računalniku s procesorjem Pentium II, 266 Mhz in 128 Mb pomnilnika				
potrebnost: 5	stabilnost: 4	preverljivost: 4	prioriteta: 4	vir: ZU006, ZU007

ZO014				
Test aplikacije, lahko obdela eno sliko v 10 sekundah				
potrebnost: 5	stabilnost: 4	preverljivost: 4	prioriteta: 5	vir: ZU008



ANALIZA SLIKE:



Projekt:	Kontrola keramičnih ploščic
različica:	0.9
šifra dokumenta:	PN-1-2005
datum:	17.2.2005

## Podrobni načrt

### Splošen opis sistema

#### Strojna oprema:

- Računalnik Siemens X3S s procesorjem Pentium II, 266 Mhz in 128 Mb RAMA.

#### Programska oprema:

- Operacijski sistem Mandrake Linux 10.0,
- Knjižnica Qt, verzija 3.1 ali novejša.

### Specifikacija komponent:

PN001	
Nalaganje slike iz diska	
Vhod: /	
Funkcija: S pomočjo funkcije QPixmap::load naložimo sliko z diska v objekt tipa QPixmap	
Izhod: Razred QPixmap, ki vsebuje sliko ploščice	
	vir: ZO001

PN002	
Binarizacija slike	
Vhod: Razred QPixmap, ki vsebuje sliko ploščice (izhod iz PN001)	
Funkcija: S pomočjo barv na vogalih slike določimo barvo ozadja in vse pixle, ki so take ali podobne barve na originalni sliki pobarvamo črno, vse ostale pa belo	
Izhod: Binarna slika ploščice	
	vir: ZO002

PN003
Iskanje robov
Vhod: Binarna slika (izhod iz PN002)
Funkcija: Na binarni sliki ploščice določimo, kateri pixli mejijo na pixle, ki imajo različno barvo od njih samih. Na novi binarni sliki označimo z belo barvo tiste pixle, katerih istoležni pixli predstavljajo rob na originalni sliki. Preostale pixle označimo s črno barvo.
Izhod: Binarna slika, na kateri so robovi označeni z belo barvo, ostalo pa je črno
vir: ZO003

PN004
Iskanje premic - napolnitev Houghovega akumulacijskega polja
Vhod: Binarna slika (izhod iz PN003)
Funkcija: Iz binarne slike robov ploščice s pomočjo Houghove transformacije napolnimo Houghovo akumulacijsko polje. Akumulacijsko polje je sestavljeno iz $nr \cdot nk$ celic, kjer $nr$ predstavlja število različnih oddaljenosti premice od izhodišča, $nk$ število različnih kotov, ki jih premice oklepajo z abscisno osjo. Vrednost v celici s koordinatama $(r, k)$ predstavlja število belih pixlov na vhodni binarni sliki, skozi katere gre premica, ki jo označujeta $r$ -ta oddaljenost od izhodišča in $k$ -ti kot.
Izhod: Napolnjeno Houghovo akumulacijsko polje
vir: ZO004

PN005
Iskanje premic - iskanje 4 največjih vrednosti Houghovega akumulacijskega polja
Vhod: Napolnjeno Houghovo akumulacijsko polje (izhod iz PN004)
Funkcija: V napolnjenega Houghovem akumulacijskem polju poiščemo 4 največje vrednosti, ki predstavljajo premice, ki oklepajo ploščico. Iz pozicij v polju, kjer se te vrednosti nahajajo, izračunamo parametre 4 premic.
Izhod: Parametri 4 premic
vir: ZO004

PN006
Iskanje vogalov ploščice
Vhod: Parametri 4 premic (izhod iz PN005)
Funkcija: Iz parametrov 4 premic izračunamo vogale. Najprej določimo para vzporednih premic na osnovi kota med njimi, potem pa določimo presečišča vsake premice s tistima dvema, ki ji nista vzporedni
Izhod: Koordinate 4 vogalov
vir: ZO005

PN007
Izračun dolžine stranic
Vhod: Koordinate 4 vogalov (izhod iz PN006)
Funkcija: Iz koordinat 4 vogalov izračunamo dolžine daljic med njimi, upoštevajoč razmerje med pixli in centimetri
Izhod: dolžine vseh 4 stranic ploščice
vir: ZO006

PN008
Izračun kotov med stranicami
Vhod: Parametri 4 premic (izhod iz PN005)
Funkcija: Iz parametrov 4 premic izračunamo kote med stranicami. Najprej določimo para vzporednih premic na osnovi kota med njimi, potem pa določimo kot, ki ga vsaka premica oklepa s tistima dvema, ki ji nista vzporedni
Izhod: vsi 4 koti med stranicami ploščice
vir: ZO007

PN009
Preverjanje, če so dolžine stranic v mejah tolerance
Vhod: dolžine vseh 4 stranic ploščice (izhod iz PN007)
Funkcija: preverimo, če so dolžine vseh 4 stranic ploščice v mejah tolerance, kot jih določajo parametri v nastavitvah.
Izhod: vrednost TRUE (če so dolžine v mejah tolerance) in FALSE sicer
vir: ZO008

PN010
Preverjanje, če so koti med stranicami v mejah tolerance
Vhod: vsi 4 koti med stranicami ploščice (izhod iz PN008)
Funkcija: preverimo, če so vsi 4 koti med stranicami ploščice v mejah tolerance, kot jih določajo parametri v nastavitvah.
Izhod: vrednost TRUE (če so koti v mejah tolerance) in FALSE sicer
vir: ZO009

PN011
Vnos parametrov
Vhod: /
Funkcija: odpre se okno/dialog, kjer so prikazane trenutne nastavitve parametrov za različne meje toleranc, katere je možno spremeniti. Klik na gumb OK zapre okno in nastavitve iz dialoga zapiše v za to namenjene globalne spremenljivke. Klik na gumb Prekliči pa samo zapre okno.
Izhod: /
vir: ZO010

PN012	
Glavno okno aplikacije	
Vhod: /	
<p>Funkcija: v tem oknu/dialogu se izvaja glavni del procesiranja. Aplikacija najprej pregleda dogovorjen imenik na disku, če se tam nahaja slika ploščice. Če se ne, potem za kratek čas prepusti nadzor operacijskemu sistemu (idle) in ponovi pregled imenika. Če pa se slika nahaja v imeniku, potem jo aplikacija naloži, prikaže v glavnem oknu in jo izbriše z diska. Nato začne z obdelavo slike. S pomočjo klika na gumb "Prekini delovanje" se avtomatska obdelava slike prekine. S klikom na gumb "Nastavitve" se odpre okno za vnos nastavitvev, s klikom na gumb "Izhod iz aplikacije" pa zaključimo delo z aplikacijo.</p>	
Izhod: /	
	vir: ZO011

PN013	
Okno za prikaz pomoči	
Vhod: /	
<p>Funkcija: V tem oknu se prikaže vgrajena pomoč uporabniku. Na levi strani lahko uporabnik izbere poglavje, katerega vsebina se nato prikaže na desni strani.</p>	
Izhod: /	
	vir: ZO012

# Specifikacija vmesnikov:

## Binarna slika:

- celi števili  $w$  (dolžina slike v pixlih),  $h$  (širina slike v pixlih),
- vektor  $im$  (vsebuje  $w \cdot h$  binarnih vrednosti, ki predstavljajo vsebino binarne slike).

## Houghovo akumulacijsko polje:

- celo število  $nr$  (število različnih oddaljenosti premic od izhodišča),
- celo število  $nk$  (število različnih kotov med premicami in abscisno osjo),
- vektor  $av$  (vsebuje  $nr \cdot nk$  celih števil, od katerih vsako predstavlja število belih pixlov, skozi katere gre premica).

## Parametri premice:

realna števila  $a$ ,  $b$ ,  $c$ , ki določajo premico  $ax + by + c = 0$ .

## Koordinate vogala:

celi števili  $x$ ,  $y$ , ki določata pozicijo vogala na sliki.

## Kot med stranicami:

realno število, ki nam pove kot v radianih.

## Dialog za vnos nastavitv/meja toleranc:

- vnosno polje QLineEdit  $leIDP$  - vnos idealne dolžine),
- vnosno polje QLineEdit  $leISP$  - vnos idealne širine,
- vnosno polje QLineEdit  $leDOD$  - vnos dovoljenega odstopanja dolžine,
- vnosno polje QLineEdit  $leDOS$  - vnos dovoljenega odstopanja širine,
- vnosno polje QLineEdit  $leDOK$  - vnos dovoljenega odstopanja kotov,
- gumb QPushButton  $pbPomoc$  - povzroči prikaz pomoči,
- gumb QPushButton  $pbOK$  - povzroči uveljavitev sprememb nastavitv,
- gumb QPushButton  $pbPrekli$  - povzroči, da se spremembe nastavitv ne uveljavijo.

## Dialog za prikaz pomoči:

- drevesni prikaz poglavij QListView  $lvPoglavja$  - prikaz vseh poglavij,
- prikaz HTML kode QTextBox  $tbPomoc$  - prikaz vsebine poglavja,
- gumb QPushButton  $pbZapri$  - zapre okno s pomočjo.

## Glavno okno:

- prikaz slike QLabel lbSlika - prikaže sliko ploščice v obdelavi,
- gumb QPushButton pbPrekini - prekine avtomatsko obdelavo,
- gumb QPushButton pbNastavitve - odpre dialog za vnos meja toleranc,
- gumb QPushButton pbPomoc - odpre dialog za prikaz pomoči,
- gumb QPushButton pbIzhod - zaključimo delo z aplikacijo,
- prikaz teksta QLabel lbStatus - izpisuje, kaj aplikacija trenutno počne.