SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 314

RAZVOJ SUSTAVA ZA LABORATORIJSKO MJERENJE RAČUNALNIM VIDOM

Dražen Dostal

Zagreb, lipanj 2011.

ZAHVALA

Zahvaljujem svima koji su svojim savjetima i podrškom pridonijeli izradi ovoga diplomskog rada, posebice mentoru doc. dr. sc. Siniši Šegviću te vodstvu i osoblju Mjeriteljskog laboratorija instituta IGH d.d.

Sadržaj:

1. Uvod

S napretkom kvalitete raznih proizvoda zahtijevaju se i sve manja odstupanja karakteristika proizvoda od onih propisanih proizvodnim procesom. Među te karakteristike se između ostalog ubrajaju dimenzije proizvoda, svojstva otpornosti na deformacije ili pak krajnje tolerancije materijala od kojeg je izrađen. Za nadzor i provjeru ovih karakteristika se koriste metode iz nekoliko znanstvenih grana, kao što su mjeriteljstvo ili fizika materijala.

Do kraja prošlog stoljeća za ove zadatke su se u velikoj mjeri koristili fizikalni mjerni instrumenti (instrumenti koji koriste fizički kontakt sa ispitnim predmetom kako bi izmjerili njegova svojstva). Takvi uređaji su spori za korištenje na proizvodnoj traci, skupi su radi složenog postupka izrade, a isto tako se sve više bliže fizičkoj granici preciznosti. Alternativa ovakvim uređajima proizlazi iz područja računalnog vida.

Ideja je da se spori i skupi uređaj zamijeni sa relativno jeftinijim optičkim senzorom (CCD ili CMOS senzor, kamera) uz koji dolazi i mjerni softver. Ovakav sustav ne treba fizički kontakt sa ispitnim materijalom i kao takav je mnogo otporniji na fizička oštećenja kada dođe do pucanja ispitnog predmeta pod prevelikim opterećenjem. Dodatna prednost ovog sustava je svakako brzina, pošto je dovoljno da se na ispitni predmet nalijepe markeri (naspram postavljanja i preciznog pozicioniranja cijelog mjernog uređaja) i mjerenje se već može izvršiti. Što se tiče preciznosti, s napretkom izrade optičkih senzora, kao i povećanjem računalne moći računala te razvojem sve boljih algoritama ona postaje čak i bolja nego kod fizikalnih mjernih instrumenata.

Cilj ovog rada je razvoj spomenutog sustava za laboratorijsko mjerenje računalnim vidom, s naglaskom na metodi za mjerenje. Ideja je da se iz video izvora na obradu šalje niz slika predmeta pod presom. Predmet je označen markerima i tlači se pri čemu mu se mijenja elongacija. Na slikama se zatim raznim algoritmima detektira položaj markera, prati se njihovi pomaci te konačno određuje promjena elongacije predmeta.

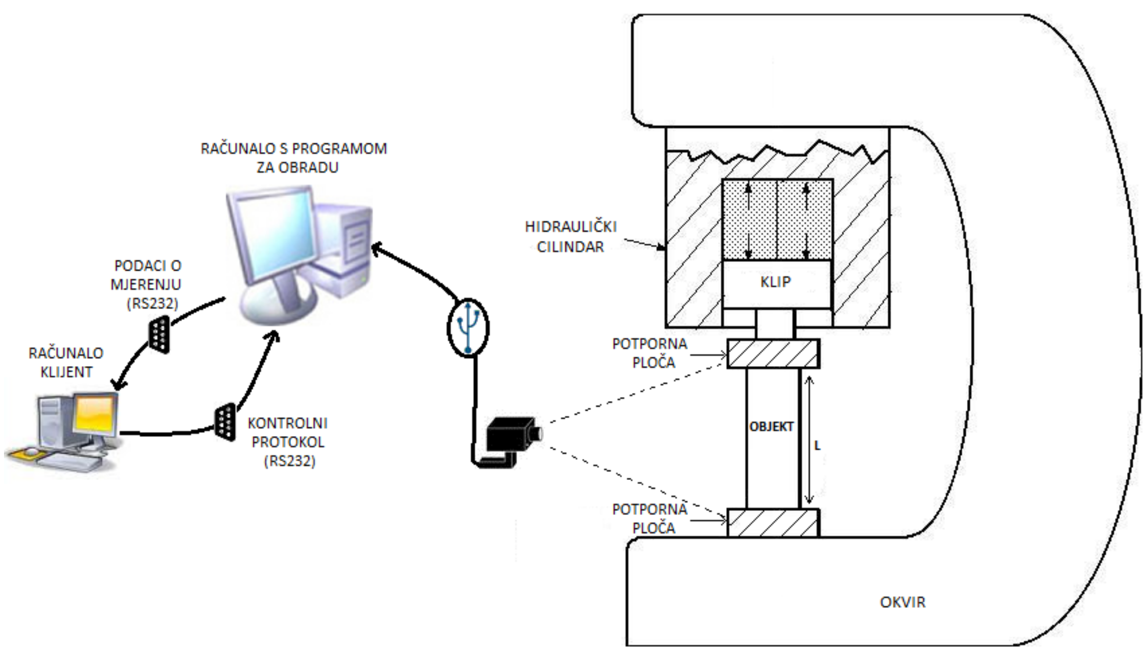
Rad je strukturiran na sljedeći način. U poglavlju 2 je opisan laboratorijski sustav i isprobane metode i postupci. U poglavlju 3 opisana je razvijena metoda te je predstavljena analiza utjecaja raznih parametara na mjerenje. U poglavlju 4 opisana je programska izvedba algoritma. Eksperimentalni rezultati prikazani su u poglavlju 5. U poglavlju 6 dan je završni komentar u kojem se ocjenjuje preciznost algoritma te ističu njegove prednosti i nedostaci. Pregled literature dan je u poglavlju 7.

2. Laboratorijski sustav i isprobane metode

2.1. Shema laboratorijskog sustava

Laboratorijski se sustav za mjerenje računalnim vidom u ovom radu sastoji od više elemenata. Primarno su to kamera i objekt (predmet) koji se mjeri. Kamera je u ovom slučaju kompaktni fotoaparat Samsung PL60 (tehničke specifikacije su navedene u Dodatku A) koji je spojen s računalom preko USB 2.0 sučelja. Objekt je označen markerima i položen između klipova prese. Ovi elementi čine osnovni sustav koji je implementiran u ovom radu.

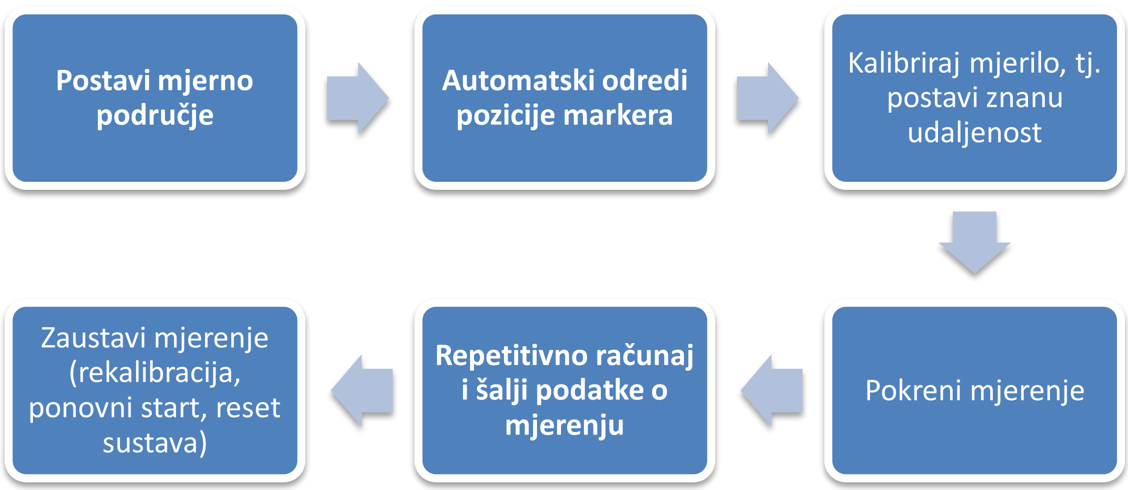
Potpuni laboratorijski sustav sadrži dodatno i klijentsko računalo koje preko serijskog sučelja (RS 232) upravlja programom za mjerenje te preko istog sučelja dobiva i rezultat mjerenja. Protokol ove komunikacije je osmišljen u okviru ovog rada i naveden je u Dodatku B. Shema potpunog sustava se vidi na slici 2.1.



Slika 2.1 – Shema potpunog laboratorijskog sustava za mjerenje računalnim vidom

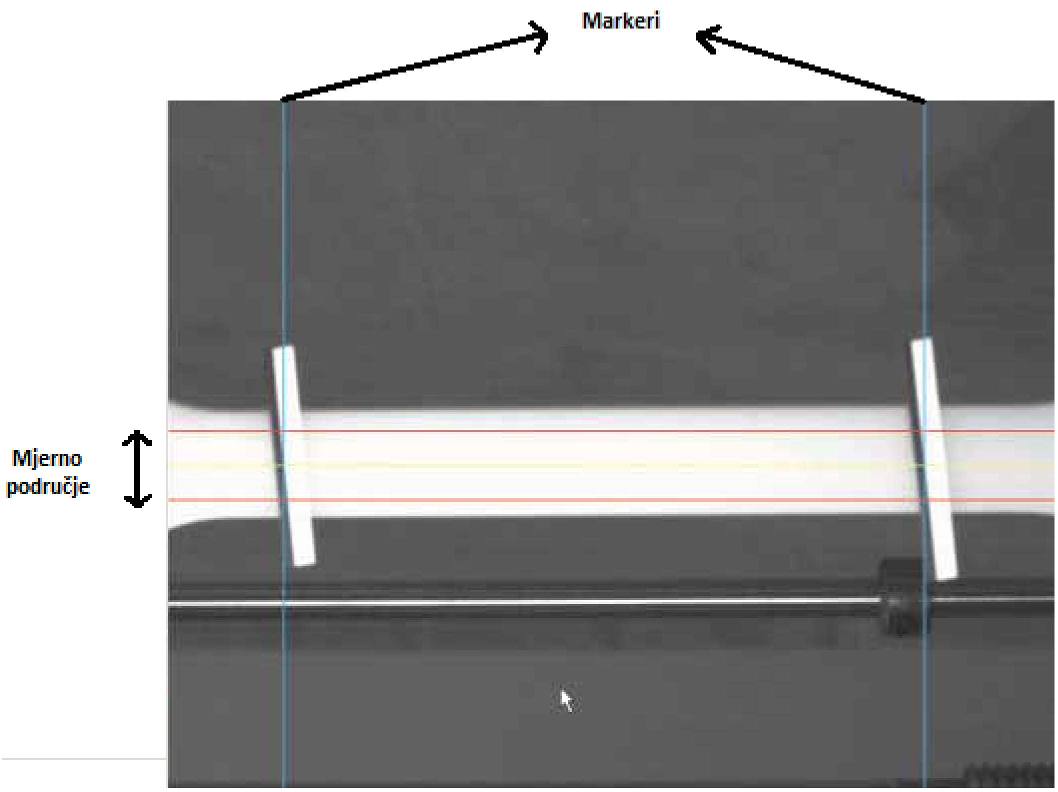
2.2. Osnovni primjer korištenja

Osnovni use-case ovog sustava se sastoji od nekoliko koraka (slika 2.2). Prvo se postavlja mjerno područje, tj. područje od interesa na slici (*ROI*) kako bi se uklonio mogući utjecaj pozadinskih predmeta i dobilo pouzdanije mjerenje. Zatim se automatski određuje položaj markera. Nakon toga slijedi kalibracija udaljenosti između markera (korisnik deklarira početnu udaljenost između markera kojoj se vjeruje). Sljedeći korak je pokretanje mjerenja u kojem se repetitivno računaju i šalju podaci o mjerenju. Konačni korak je zaustavljanje mjerenja nakon čega se sustav može ponovno kalibrirati i pokrenuti.

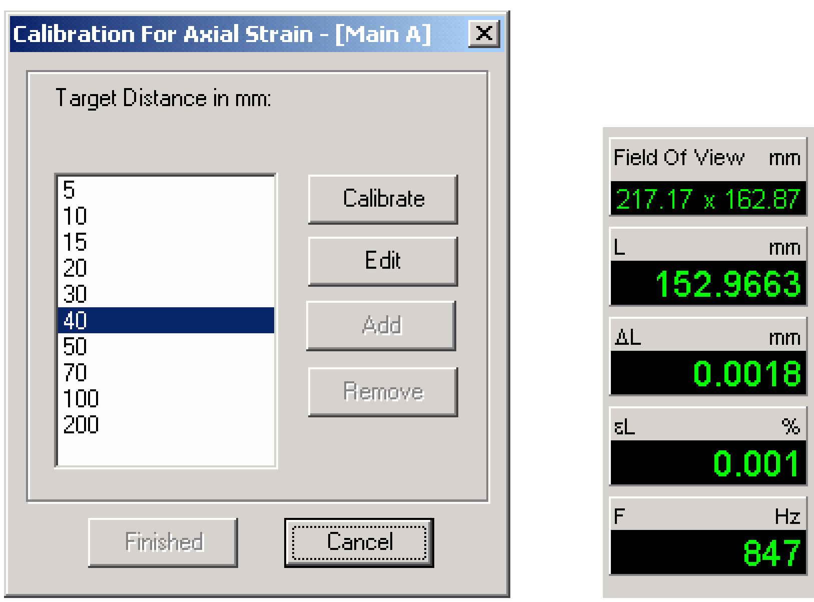


Slika 2.2 – Osnovni use-case ciljanog sustava

Na sljedećim slikama se vide spomenuti pojmovi na stvarnom primjeru već postojećeg sličnog sustava [8]. Slika 2.3 prikazuje mjerno područje i automatski detektirane markere, dok se na slici 2.4. vide prikaz kalibracije i primjer ostvarenog mjerenja.



Slika 2.3 – Prikaz mjernog područja i markera



Slika 2.4 – Prikaz kalibracije (lijevo) i primjer mjerenja (desno)

Tablica 2.1 prikazuje preciznosti (minimalnu rezoluciju) koje garantira spomenuti postojeći sustav. Ove preciznosti ovise o više parametara, kao što su veličina predmeta, rezolucija kamere ili osvjetljenje objekta, stoga služe samo kao orijentacija, ali ne i konačna ocjena uspješnosti odabrane metode u ovom radu.

Duljina predmeta (duž os pružanja)	Minimalna rezolucija
50 mm	<0.3 µm
250 mm	<1.5 µm
500 mm	<3 µm
1000 mm	<6 µm

Tablica 2.1 – Preciznost postojećeg rješenja

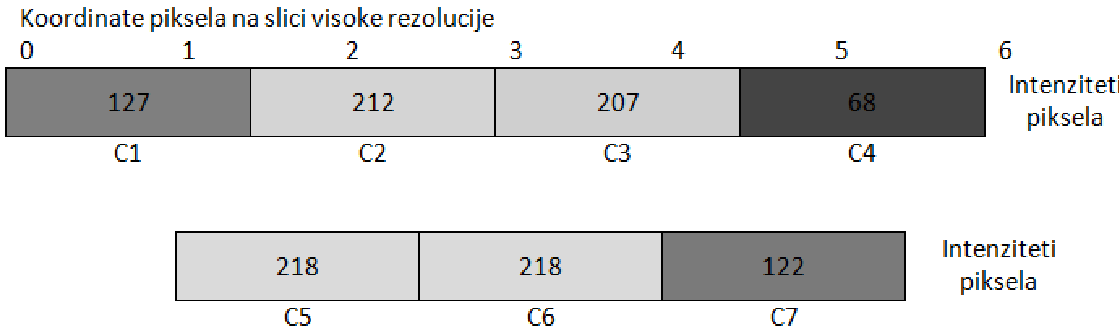
2.3. Isprobane metode i postupci

Kako bi se dobila što preciznija mjerenja, položaj markera i udaljenost između markera je potrebno odrediti s podpikselskom preciznošću. Pošto se u ovom radu koriste kontrastni markeri, glavna operacija je pronalaženje rubova markera. Postoji više načina da se ostvari podpikselska preciznost lokalizacije ruba, a svaki od njih se temelji na pokušaju pribavljanja dodatnih informacija koje bi pomogle pri ostvarivanju veće rezolucije slike. U okviru ovog rada je bilo nekoliko pokušaja implementacija tehnika temeljenih na super-rezoluciji slike:

1) Metoda super-rezolucije na temelju slijeda slika
2) Metoda super-rezolucije temeljena na slijedu slika bez eksplicitnog određivanja pomaka
3) Metoda super-rezolucije bez pomaka između slika

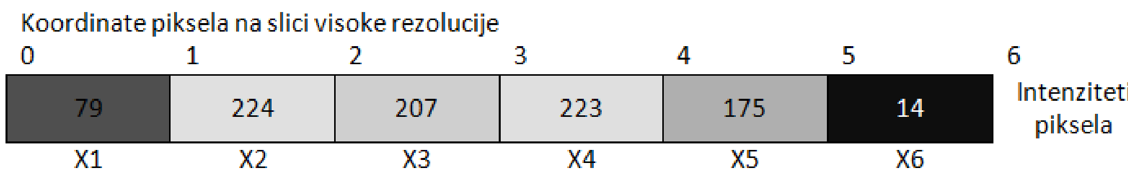
2.3.1. Metoda super-rezolucije na temelju slijeda slika

Metoda opisana u [1] se temelji na slijedu slika niske rezolucije iz kojih se rješavanjem sustava jednadžbi dolazi do iznosa intenziteta piksela slike veće rezolucije. Kao primjer postupka u 1D prostoru mogu se uzeti dvije slike (tj. jednodimenzionalni nizovi piksela) niske rezolucije na temelju kojih se računaju pikseli odgovarajuće slike visoke rezolucije:



Slika 2.5 – Dvije slike niske rezolucije (druga je pomaknuta za 0.66 piksela)

Na temelju intenziteta piksela na ovim slikama se računaju intenziteti piksela slike više rezolucije, tj. slike za koju se vjeruje da je vjernija pravoj slici:



Slika 2.6 – "Izračunata" slika visoke rezolucije

Glavni parametar ove metode je koeficijent uvećanja $\boldsymbol{\rho}$ koji predstavlja omjer broja piksela svih slika niske rezolucije i broja piksela željene slike visoke rezolucije, tj. predstavlja koeficijent povećanja rezolucije. Eksperimenti provedeni u [1] pokazuju da je maksimalno ostvarivo povećanje rezolucije ovom metodom ono s faktorom $\boldsymbol{\rho} = \mathbf{1.9}$. Nepoznati intenziteti piksela se za spomenuti primjer (slike 2.5 i 2.6) računaju na način da se svaki piksel niske rezolucije izrazi kao kombinacija piksela visoke rezolucije pomnoženih faktorom uvećanja:

$$C1 = \left(X1 + \frac{1}{2}X2\right)\rho^{-1} \quad [1]$$

$$C2 = \left(\frac{1}{2}X2 + X3\right)\rho^{-1} \quad [2]$$

$$C3 = \left(X4 + \frac{1}{2}X5\right)\rho^{-1} \quad [3]$$

$$C4 = \left(\frac{1}{2}X5 + X6\right)\rho^{-1} \tag{4}$$

$$C5 = \left(X2 + \frac{1}{2}X3\right)\rho^{-1} \tag{5}$$

$$C6 = \left(\frac{1}{2}X3 + X4\right)\rho^{-1} \tag{6}$$

$$C7 = \left(X5 + \frac{1}{2}X6\right)\rho^{-1} \tag{7}$$

Ovako predstavljen sustav jednadžbi se može prikazati u obliku sustava matrica:

$$C = A * X \tag{8}$$

$$A = \rho^{-1}\begin{bmatrix} 1 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1 \\ 0 & 1 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1/2 \end{bmatrix} \tag{9}$$

$$C = \begin{bmatrix} 127 \\ 212 \\ 207 \\ 68 \\ 218 \\ 218 \\ 122 \end{bmatrix} \tag{10}$$

Rješenje ovog sustava predstavlja iznose intenziteta piksela slike visoke rezolucije:

$$X = [A^T A]^{-1} * A^T C = [79 \quad 224 \quad 207 \quad 223 \quad 175 \quad 14] \tag{11}$$

Glavni dio ovog rješenja je matrica sustava $\boldsymbol{A}$ te njezin inverz. Unosi u ovoj matrici se temelje na pomacima između slika niske rezolucije i prostorne rešetke ciljne slike visoke rezolucije. Pošto su snimke u ovom radu dobivene s jedne nepomične kamere, spomenuti pomak je uvijek isti. To znači da su pojedini redci matrice sustava identični, odnosno matrica sustava je singularna, tj. ne može joj se izračunati inverz.

Kako bi se u potpunosti otklonila mogućnost uporabe ove metode, napravljen je postupak registracije niza ispitnih slika. Ovaj postupak je pokazao da je pomak između slika vidljiv tek kao tisućinka piksela, što nije dovoljno za ispravan rad metode (ovako mali pomak se može pripisati pogrešci registracije uslijed šuma na slikama) te je ona odbačena iz daljnjeg razmatranja.

2.3.2. Metoda super-rezolucije bez eksplicitnog određivanja pomaka

Metoda super-rezolicije opisana u [2] se temelji na slijedu slika niske rezolucije iz kojih se postupkom bez eksplicitnog određivanja pomaka određuje slika visoke rezolucije. Osnovna ideja postupka je da je niz slika niske rezolucije (*LR, Low Resolution*) dobiven iz iste slike visoke rezolucije (*HR, High Resolution*) koja je podložena degradaciji. Spomenuta degradacija uključuje zaglađivanje slike, prostorno iskrivljenje slike (pomaci) te smanjenje frekvencije uzorkovanja (*downsampling*):

$$u_i = S * W_i * B * f + n_i \quad [12]$$

$$u_i \rightarrow LR\ rešetka => \ \Omega = [1, \dots, M] \times [1, \dots, N] \quad [13]$$

$$f \rightarrow HR\ rešetka => \ \Psi = [1, \dots, Mz] \times [1, \dots, Nz] \quad [14]$$

pri čemu $\boldsymbol{u_i}$ odgovara slici niske rezolucije, $\boldsymbol{f}$ slici visoke rezolucije (faktor uvećanja $\boldsymbol{z}$), $\boldsymbol{B}$ je operator zaglađivanja, $\boldsymbol{W_i}$ predstavlja mapiranje piksela rešetke visoke rezolucije na rešetku niske rezolucije (za *i*-tu sliku niske rezolucije), $\boldsymbol{S}$ je operator smanjenja frekvencije uzorkovanja, a $\boldsymbol{n_i}$ predstavlja aditivni bijeli Gaussov šum. Postupak se pojednostavljuje računanjem zaglađene slike visoke rezolucije koja se kasnije može obraditi nekom od tehnika izoštravanja slike (*deblurring*) kako bi se dobila izvorna slika visoke rezolucije.

$$v = B * f \quad [15]$$

$$v_i = W_i * v \quad [16]$$

$$u_i = S * v_i + n_i \qquad [17]$$

Uvrštavanjem [15] i [16] u [12] dolazi se do konačnog postupka opisanog jednadžbom [17]. Rješenje koje se traži je $\boldsymbol{v_i}$, a ono predstavlja zaglađenu sliku visoke rezolucije. Sljedećim postupkom se zapravo traži najbolja procjena traženog rješenja $\boldsymbol{SR}(\boldsymbol{v_i})$:

$$E\left[SR(v_i)(x)\middle|u_j\right] = \frac{1}{W(x,i,j)} \sum_{y \in \Psi} w(x,y,i,j)\hat{u}_j(y) \qquad [18]$$

$$w(x,y,i,j) = e^{-\frac{\left\|\hat{u}_i(\mathcal{N}^d\{x\}) - \hat{u}_j(\mathcal{N}^d\{y\})\right\|_2^2}{h^2}} \qquad [19]$$

$$W(x,i,j) = \sum_{y \in \Psi} w(x,y,i,j) \qquad [20]$$

pri čemu $\hat{\boldsymbol{u}}$ predstavlja sliku visoke rezolucije dobivenu nekim od postupaka interpolacije (u ovom radu je korištena bilinearna interpolacija), $\boldsymbol{w}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{i},\boldsymbol{j})$ je mjera pouzdanosti, $\boldsymbol{\mathcal{N}^d}\{\boldsymbol{x}\}$ označava $(\mathbf{2}\boldsymbol{d}+\mathbf{1}) \times (\mathbf{2}\boldsymbol{d}+\mathbf{1})$ susjedstvo oko $\boldsymbol{x}$, s parametrom $\boldsymbol{h}$ se kontrolira zaglađenost rezultantne slike, a $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{i},\boldsymbol{j})$ je normalizator. Ovako prikazan sustav odgovara procjeni rješenja na temelju jedne slike niske rezolucije, a kada se u obzir uzme $\boldsymbol{k}$ takvih slika dolazi se do sljedećeg rješenja:

$$E\left[SR(v_i)(x)\middle|\{u_j\}_{1 \le j \le k}\right] = \frac{1}{k} \sum_{1 \le j \le k} E\left[SR(v_i)(x)\middle|u_j\right] \qquad [21]$$

U ovako opisanom sustavu se zamjećuje dosta velika računska složenost. Za svaki piksel izlazne slike prolazi se svim interpoliranim slikama niske rezolucije, za svaki piksel interpolirane slike niske rezolucije se prelaze pikseli u $(\mathbf{2}\boldsymbol{d}+\mathbf{1}) \times (\mathbf{2}\boldsymbol{d}+\mathbf{1})$ susjedstvu promatranog piksela (kako bi se izračunala $\boldsymbol{L_2^2}$ norma) što dovodi do peterostruke petlje sa složenosti $\boldsymbol{O}(\boldsymbol{k} * \boldsymbol{N}^\mathbf{2}\boldsymbol{M}^\mathbf{2}\boldsymbol{z}^\mathbf{4}(\mathbf{2}\boldsymbol{d}+\mathbf{1})^\mathbf{2})$.

Implementacija postupka je uspješno provedena u programskom jeziku matlab. Kako bi se eksperiment dodatno ubrzao, uveden je dodatni parametar $\boldsymbol{r}$ kojim se ograničava veličina susjedstva na slici niske rezolucije kojim se prelazi za svaki piksel slike visoke rezolucije (inače postupak nalaže ispitivanje cijele slike niske rezolucije). Eksperiment je proveden za sljedeće parametre:

$$k = 20, N = M = 32, z = 3, r = 13, d = 4 \qquad [22]$$

Izvršavanje eksperimenta je trajalo $\approx \boldsymbol{25min}$ čime se utvrdilo da je opisani postupak računski prezahtjevan za obradu slika u stvarnom vremenu (uračunata je i činjenica da se dosta veliko ubrzanje može dobiti implementiranjem postupka u c++ programskom jeziku).

2.3.2. Metoda super-rezolucija bez pomaka između slika

Zadnja metoda super-rezolucije iskušana u ovom radu je metoda super-rezolucije bez pomaka između slika [3]. Dodatna informacija potrebna za super-rezoluciju se kod ove metode ostvaruje na pretpostavci da su sve slike niske rezolucije nastale od slike visoke rezolucije degradirane različitim maskama zaglađivanja te pretpostavke da je prilikom smanjenja frekvencije uzorkovanja došlo do pojave aliasinga.

Postupak se provodi u frekvencijskoj domeni, pri čemu se slike niske rezolucije i maske zaglađivanja podvrgavaju diskretnoj Fourierovoj transformaciji. Spomenuti sustav degradacije se tada može prikazati kao:

$$x[m,n] \xrightarrow{DFT} X(k1,k2) \qquad [23]$$

$$y_i[m,n] \xrightarrow{DFT} Y_i(k1,k2), i = 0,\dots,M^2-1 \qquad [24]$$

$$h_i[m,n] \xrightarrow{DFT} H_i(k1,k2), i = 0,\dots,M^2-1 \qquad [25]$$

$$Y_i = H_i * X, i = 0,\dots,M^2-1 \qquad [26]$$

$$Y = H * X \qquad [27]$$

pri čemu $\boldsymbol{x}$ predstavlja sliku visoke rezolucije, $\boldsymbol{y_i}$ jednu od obzervacija (slika niske rezolucije), $\boldsymbol{M^2}$ je broj slika niske rezolucije, a $\boldsymbol{h_i}$ jednu od maski zaglađivanja. U matričnom obliku se vertikalnim slaganjem svih obzervacija i maski dolazi do konačnog sustava [27]. Elementi spomenutih matrica su navedeni u jednadžbama [28] i [29]. Za izračun slike visoke rezolucije potrebno je prvo izračunati inverz matrice $\boldsymbol{H}$. Kako bi to uopće bilo moguće, matrica ne smije biti singularna, tj. ne smije sadržavati dva ista stupca ili retka što znači da se mora temeljiti na različitim maskama zaglađivanja. Konačni izračun slike visoke rezolucije u frekvencijskoj domeni je prikazan na jednadžbama [30] i [31].

$$Y_i = \begin{bmatrix} Y_i(0,0) \\ Y_i(0,1) \\ \vdots \\ Y_i(0, \frac{N}{M} - 1) \\ \vdots \\ Y_i(\frac{N}{M} - 1, \frac{N}{M} - 1) \end{bmatrix}, X = \begin{bmatrix} X(0,0) \\ X(0,1) \\ \vdots \\ X(0, \frac{N}{M} - 1) \\ \vdots \\ X(\frac{N}{M} - 1, \frac{N}{M} - 1) \end{bmatrix} \qquad [28]$$

$$H_i = \begin{bmatrix} H_i(0,0) & 0 & \dots & 0 & H_i(\frac{N}{M}, \frac{N}{M}) & \dots & 0 \\ 0 & H_i(0,1) & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & H_i(\frac{N}{M} - 1, \frac{N}{M} - 1) & 0 & \dots & H_i(N-1, N-1) \end{bmatrix} \qquad [29]$$

$$X = H^{-1} Y \qquad [30]$$

$$X(k1, k2) \xrightarrow{IDFT} x(m, n) \qquad [31]$$

Implementacija opisanog sustava je izvedena u matlabu. Glavni problem ove metode je u tome što je potrebno izračunati inverz matrice $\boldsymbol{H}$ koja ima mnogo više elemenata od elemenata slike visoke rezolucije. Dimenzije matrice $\boldsymbol{H}$ su određene veličinom slike visoke rezolucije ($\boldsymbol{N} \times \boldsymbol{N}$) i faktorom uvećanja ($\boldsymbol{M}$). Za $\mathbf{4}$ slike niske rezolucije s $\mathbf{32} \times \mathbf{32}$ elemenata i faktor uvećanja $\mathbf{2}$, konačna slika visoke rezolucije bi trebala imati $\mathbf{64} \times \mathbf{64}$ elemenata, što znači da matrica $\boldsymbol{H}$ ima $\mathbf{2048} \times \mathbf{2048}$ ($\boldsymbol{N^2}/\boldsymbol{M} \times \boldsymbol{N^2}/\boldsymbol{M}$) elemenata. Ovo je memorijski vrlo zahtjevna operacija i za zadani primjer 1GB memorije nije bio dovoljan. Ovaj postupak je odbačen iz daljnjeg razmatranja zbog memorijske zahtjevnosti, pošto za već ovako male rezolucije postupak zahtijeva mnogo memorije (iako bi implementacija u c++ ponovno potencijalno manje memorijski zahtjevna, ali procijenjeno nedovoljno). Dodatna informacija je s daljnjim postupcima ostvarena na drugi način.

3. Razvijena metoda

Problem s isprobanim metodama super-rezolucije je taj što se većina njih zasniva na pomaku između ulaznih slika kao izvoru dodatnih informacija. Taj pomak se ostvaruje kao promjena kuta gledanja kamere ili translacija položaja kamere. Pošto se sustav u ovom radu temelji na fiksnom položaju jedne kamere i fiksnom položaju testnog objekta, super-rezolucijski postupci temeljeni na pomaku nisu primjenjivi u ovom slučaju.

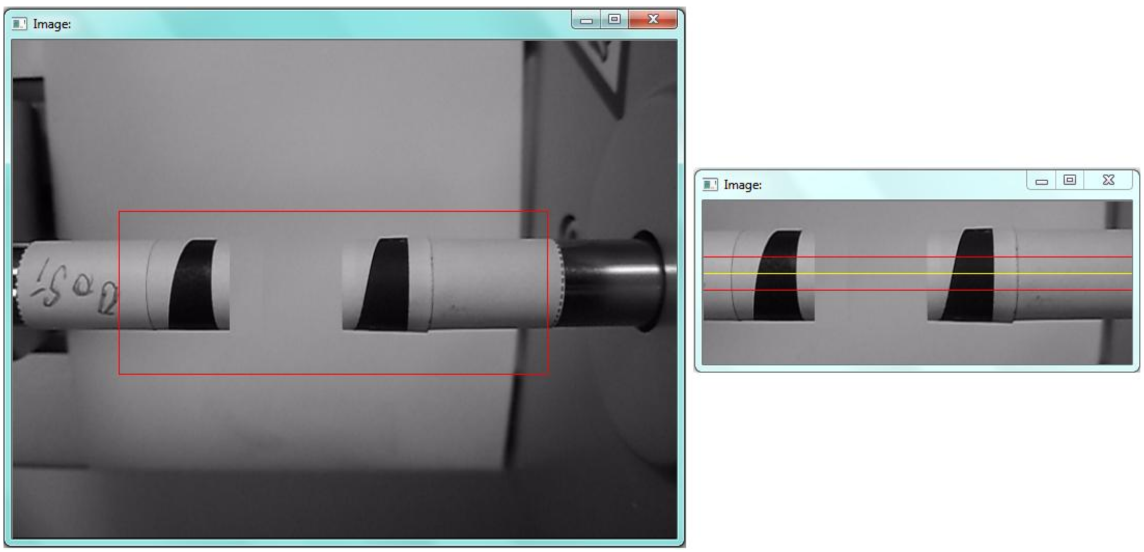
Pokazalo se da i postupci koji slove kao postupci super-rezolucije bez eksplicitnog određivanja pomaka, spomenutu dodatnu informaciju ipak dobivaju iz pomaka objekta na slijedu slika, stoga isto tako nisu primjenjivi u ovom radu. Dodatni problemi ovih postupaka su i vrijeme izvođenja i potrošnja memorije, a to dovodi u pitanje njihovu implementaciju i korisnost u primjenama koje zahtijevaju izvršavanje u stvarnom vremenu.

Radi spomenutih poteškoća, dodatna informacija je ostvarena na način da se razmak između detektiranih markera prati na više linija i kroz više slika. Takvim postupkom se dobiva veći stupanj vjerovanja da izračunata duljina predmeta odgovara stvarnoj duljini.

3.1. Predloženi postupak

Pošto je ispitni predmet obilježen kontrastnim markerima, najbolji način određivanja položaja markera je direktnom obradom horizontalnog (ili vertikalnog) histograma slike ili traženjem rubova markera na slici. Radi veće preciznosti odabran je postupak traženja položaja rubova markera korištenjem gradijentnog operatora, pri čemu su minimum i maksimum gradijenta traženi rubovi.

Ideja je da se područje od interesa na slici (područje na kojem se nalazi samo ispitni predmet sa kontrastnim markerima) promatra kao skup horizontalnih (ili vertikalnih) linija. Svaka linija se zatim prelazi sa jednodimenzionalnim gradijentnim operatorom (vidi poglavlje 3.1.2). Ovom operacijom se dobivaju razlike u intenzitetima susjednih piksela. Ekstremi među tim intenzitetima odgovaraju najvećim skokovima intenziteta piksela, tj. rubovima. Pošto šum visokih frekvencija odgovara naglim skokovima, kako se on ne bi lažno detektirao kao rub, slika se prelazi blagim Gaussovim filtrom. Ovaj postupak se naziva zaglađivanje i objašnjen je u poglavlju 3.1.1.



Slika 3.1 – Odabir ROI-a (crveni pravokutnik, lijevo) i područja s linijama od interesa (desno)

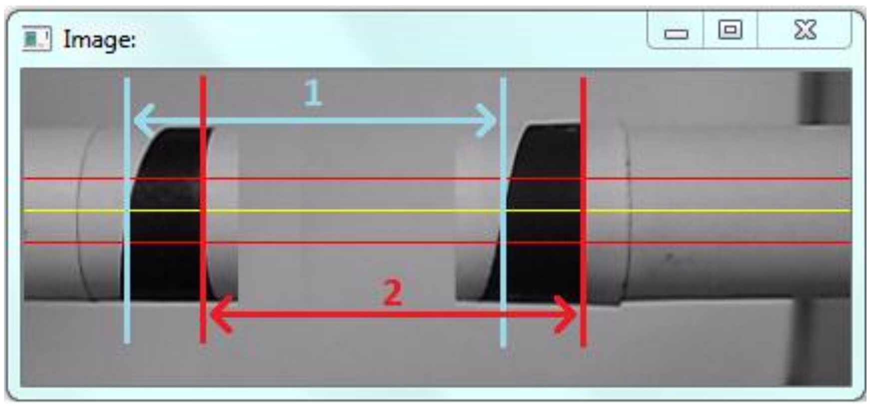
Indeksi na kojima se nalaze pravi ekstremi trenutno odgovaraju položaju piksela, što nije dovoljno za preciznu lokalizaciju ruba. Sljedeći korak je precizno određivanje položaja ruba, pri čemu se u ovom radu razmatraju dva modela ruba:

a) Model temeljen na funkciji rampe

b) Model temeljen na odzivu Gaussovog filtra nad funkcijom skoka

Pošto se na svakoj liniji može nalaziti više rubova, za svaki rub se određuje inicijalni položaj (preciznosti jednog piksela) koji se zatim odabranom metodom precizno lokalizira. Ovaj postupak se ponavlja za sve linije ispitnog područja.

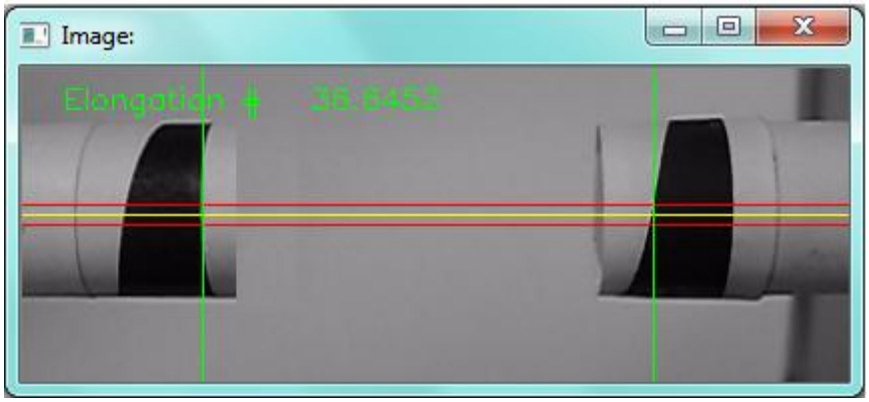
Jednom kad su detektirani i precizno lokalizirani svi rubovi na svim linijama, može se pristupiti sparivanju rubova kako bi se odredila udaljenost između njih, a kasnije odredili i nastali pomaci, tj. promjene udaljenosti između markera.



Slika 3.2 – Primjer detekcije 4 ruba i 2 izmjerene duljine

Inicijalno se na prvih nekoliko slika provodi kalibracija zadane udaljenosti. Ovaj postupak se provodi na centralnoj liniji ispitnog područja jer markeri ne moraju biti paralelni, tj. ako nisu paralelni onda udaljenosti između uparenih markera nisu iste za sve ispitne linije, a time kalibracija postaje neprecizna. Rezultat kalibracije je koeficijent omjera udaljenosti i broja piksela između markera.

Nakon kalibracije slijedi praćenje pomaka markera za svaku liniju. Svi pomaci se zatim usrednjuju, čime se dobiva veći stupanj vjerovanja da izračunati pomak odgovara stvarnom pomaku. Pomak se zatim množi sa spomenutim koeficijentom omjera kako bi se dobio podatak o pomaku izražen u zadanoj mjernoj jedinici. Dodatno je implementirana opcija praćenja pomaka kroz više slika čime se postiže još bolja procjena mjerenja. Kod ove opcije se pomaci zabilježeni na više slika dodatno usrednjuju.



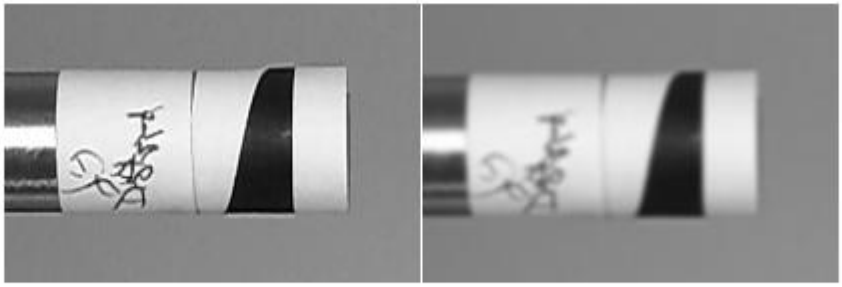
Slika 3.3 – Konačni rezultat mjerenja (prikazani marker i duljina predmeta)

3.1.1. Zaglađivanje

Glađenje je osnovni postupak pripreme slike za daljnju obradu. To je postupak filtriranja slike koji kao rezultat, u odnosu na originalnu sliku, daje zamagljenu sliku smanjene detaljnosti [4]. Glađenjem slike se uklanja šum koji je sastavni dio slike. Šum se uglavnom sastoji od visokih frekvencija pa ga se može ukloniti filtriranjem niskopropusnim filtrom. Za ovakve primjene se koristi diskretno filtriranje, pri čemu se na ulaznu funkciju (***f(x)***) primjenjuje filtar (***h(N)***) te dobiva nova izlazna funkcija (***g(x)***). Ovdje se koristi prednost separabilnog filtra, tj. svojstva da se dvodimenzionalni separabilni filtar može predstaviti sa dva jednodimenzionalna filtra:

$$g(x) = \sum_{N=-\infty}^{+\infty} f(x-N)h(N) \qquad [32]$$

Separabilni filtar koji se koristi u ovom radu je Gaussov filtar. On ovisi o parametru sigma (**σ**), tj. standardnoj devijaciji Gaussove funkcije (rasprostranjenost te funkcije). Za ovaj zadatak je iskorištena gotova implementacija dostupna u biblioteci OpenCV. Korišteno je blago zaglađivanje s parametrom **σ=0.5** i to samo po horizontalnoj dimenziji. Utjecaj zaglađivanja sa dosta većim iznosom parametra **σ** je vidljiv na slici 3.4.



Slika 3.4 – Originalni detalj sa slike te isti detalj sa zaglađene slike (σ=1.5)

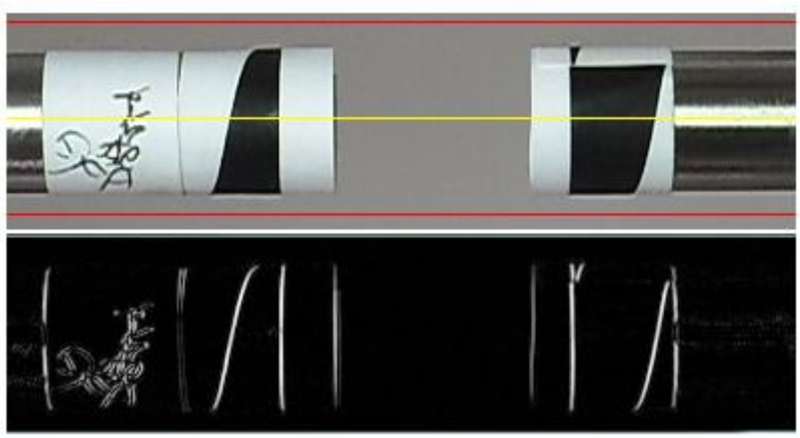
Šum može dosta utjecati na intenzitete piksela i tako pogoršati preciznost algoritma. Iz tog razloga se koristi zaglađivanje kako bi se on uklonio prije detekcije ruba.

3.1.2. Gradijentni operator

Gradijent pruža informaciju o intenzitetu i smjeru najvećeg porasta funkcije [4]. U ovom slučaju se koristi kako bi se pronašle pozicije rubova, jer rubovi odgovaraju skokovima u intenzitetu piksela. Ovim postupkom se može odrediti položaj ruba do grube preciznosti od jednog piksela. Pošto se u ovom radu rubovi pronalaze na linijama i kut gradijenta nije potreban, svaka linija se obrađuje 1D gradijentnim operatorom koji je zapravo maska veličine 1x3 (prikazana maska odgovara 1D Prewittovom gradijentnom operatoru):

$$G_x = [-1 \quad 0 \quad +1] \qquad [33]$$

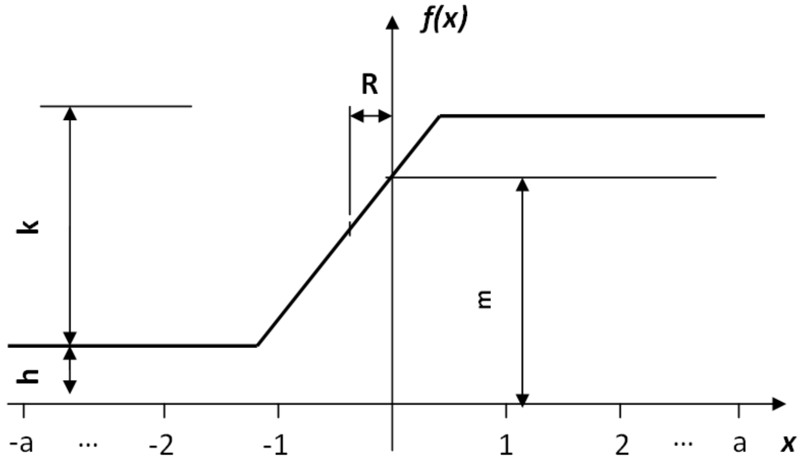
Pri izračunu iznosa gradijenta G_x na poziciji određenog piksela, računa se razlika intenziteta njegovih susjeda neposredno lijevo i desno. Najveće apsolutne razlike odgovaraju najjače izraženim rubovima na liniji, koji zapravo daju položaj kontrastnih markera na slici 3.5. Poželjno je da prije obrade gradijentnim operatorom slika prođe proces zaglađivanja kako bi se uklonio šum, a time i uklonili lažni rubovi. Rezultat ovog postupka je položaj ruba do preciznosti jednog piksela, a taj podatak služi kako bi se ubrzao postupak precizne lokalizacije ruba.



Slika 3.5 – Originalna slika (vidljive su linije odabira ROI-a) i slika s gradijentima

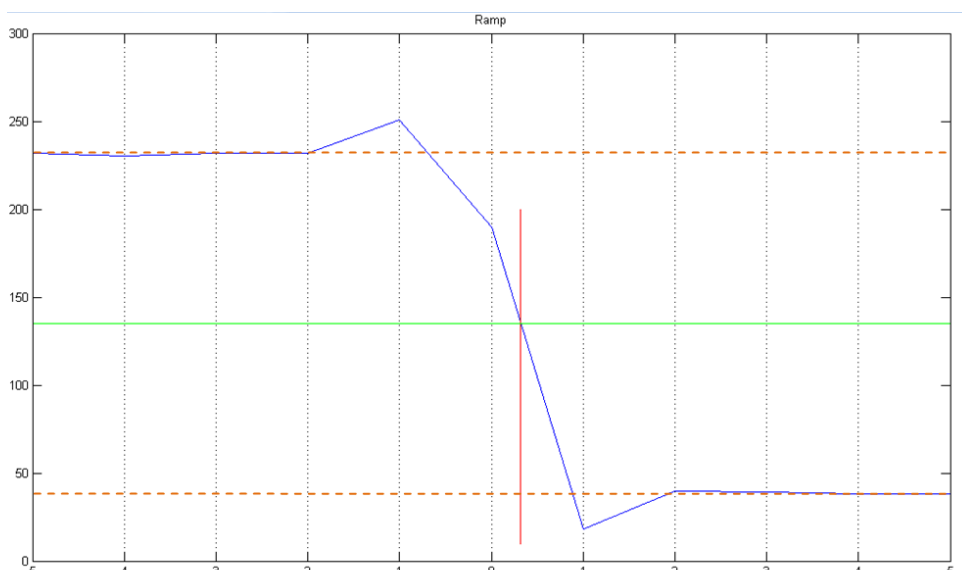
3.1.3. Metoda s modelom ruba temeljenim na funkciji rampe

Metoda s modelom ruba temeljenim na funkciji rampe (u tekstu dalje model s funkcijom rampe) kao osnovu uzima funkciju rampe koja je prikazana na slici 3.6. Model pretpostavlja da je pravi položaj ruba između piksela tj. na većoj rezoluciji od one zadane prostornom rešetkom na koju se mapiraju pikseli.



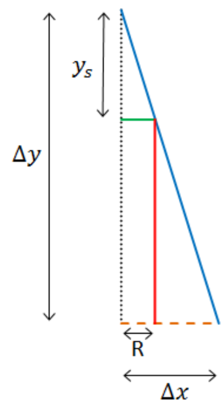
Slika 3.6 – Model s funkcijom rampe

Funkcija ***f(x)*** predstavlja intenzitet piksela na poziciji ***x***. Parametar ***h*** odgovara intenzitetu pozadine, ***k*** odgovara kontrastu intenziteta pozadine i predmeta, ***m*** je intenzitet promatranog piksela s najvećim iznosom gradijenta, dok je ***R*** udaljenost pravog ruba od položaja piksela sa najvećim iznosom gradijenta na zadanom dijelu linije (na slici piksel ***x=0***). Dodatno se uvodi parametar $\boldsymbol{\alpha}$ koji predstavlja udaljenost susjednih piksela koji se uzimaju kao pikseli sa stabilnom vrijednosti dvaju platoa. Platoi su homogene površine s približno jednakim vrijednostima piksela prije i poslije ruba, tj. govorimo o donjem i gornjem platou. Estimacija parametra $\boldsymbol{R}$ se provodi na temelju presjecišta centralnog platoa s pravcem u smjeru najvećeg iznosa gradijenta, i to lijevo ili desno od položaja grubo lokaliziranog ruba. Na slici 3.7 su prikazani gornji i donji plato (gornja i donja narančasta isprekidana linija), centralni plato (puna zelena horizontalna linija) te linija koja prolazi presjecištem centralnog platoa i pravca u smjeru najvećeg iznosa gradijenta.



Slika 3.7 – Prikaz stvarne karakteristike ruba te lokalizacija opisanom metodom

Na slici 3.7 se naziru dva slična trokuta na temelju kojih se može izračunati estimirani $\boldsymbol{R}$:



Slika 3.8 – Izdvojeni slični trokuti

Na temelju pravila o sličnim trokutima se jednostavno dolazi do rješenja problema:

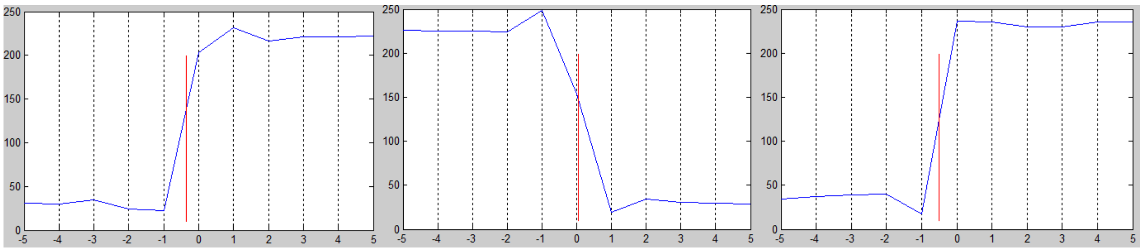
$$\frac{R}{\Delta x} = \frac{y_s}{\Delta y} \quad \overset{\Delta x=1}{\Longrightarrow} \quad R = \frac{y_s}{\Delta y} \qquad [34]$$

Zaključno s prikazanim rješenjem problema, parametar ***R*** se može odrediti na temelju dostupnih podataka o intenzitetu promatranog piksela (kao ulaz u ovu metodu se predaje indeks piksela $\boldsymbol{f_x}$) i intenzitetima susjednih piksela i to na sljedeći način (u prvoj jednadžbi se koriste direktni iznosi piksela, a u drugoj parametri opisanog modela, rezultat je isti):

$$R = \begin{cases} -\dfrac{f_{x-a} - 2f_x + f_{x+a}}{2(f_{x-a} - f_x)}, & ako\ |f_{x-a} - f_x| > |f_x - f_{x+a}| \\ 0, & ako\ |f_{x-a} - f_x| = |f_x - f_{x+a}| \\ -\dfrac{f_{x-a} - 2f_x + f_{x+a}}{2(f_x - f_{x+a})}, & inače \end{cases} \qquad [35]$$

$$R = \begin{cases} -\dfrac{2(h-m)+k}{2(m-h-k)}, & ako\ |h-m| > |m-h-k| \\ 0, & ako\ |h-m| = |m-h-k| \\ -\dfrac{2(h-m)+k}{2(h-m)}, & inače \end{cases} \qquad [36]$$

Kao što se može zaključiti iz jednadžbi [35] i [36], ovaj model koristi intenzitete središnjeg piksela i intenzitete susjednih piksela (koji su od promatranog središnjeg piksela udaljeni za cijeli broj $\boldsymbol{a}$, ovo je ujedno i parametar širine promatranog područja za metodu opisanu u 3.1.4.) kako bi se precizno lokalizirao rub na liniji. Primjer precizne lokalizacije s opisanom metodom je prikazan na slici 3.9.

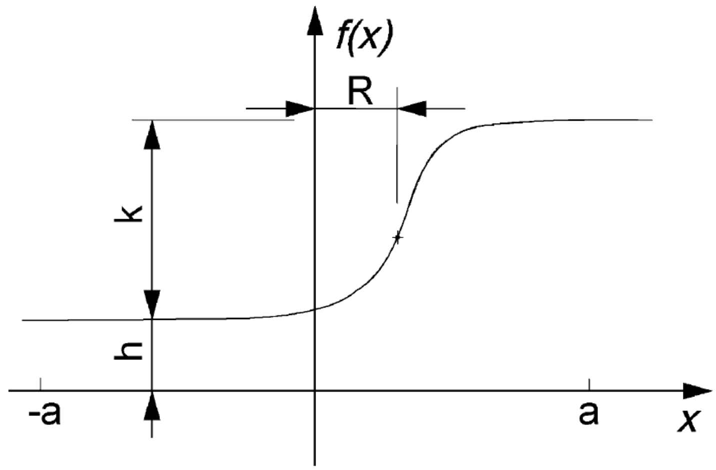


Slika 3.9 – Primjer lokalizacije ruba opisanom metodom (crvena linija označava položaj ruba)

Pošto se ovdje koriste samo tri podatka i jednostavna računska operacija, ovaj postupak je mnogo brži od metode nalijeganja, ali isto tako stoji i pretpostavka da unosi veću pogrešku zbog modela koji nije u potpunosti prilagođen stvarnom rubu.

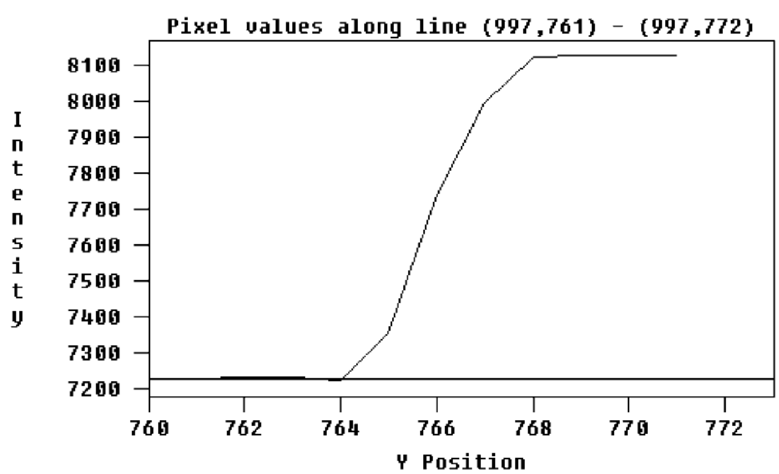
3.1.4. Metoda s modelom ruba temeljenim na odzivu Gaussovog filtra nad funkcijom skoka

Za razliku od pretpostavke da se rub može opisati modelom koji se zasniva na funkciji rampe, ova metoda koristi pretpostavku da je rub opisan nešto složenijom funkcijom Gaussovog modela ruba. Dodatno, parametri funkcije nisu određeni statički, već se za svaki detektirani rub ponovno određuju metodom nelinearne optimizacije [5]. Model ruba zasnovan na odzivu Gaussovog filtra nad funkcijom skoka (dalje u tekstu model s Gaussovom funkcijom) je prikazan na slici 3.10 (slika preuzeta iz [5]).



Slika 3.10 – Model ruba temeljen na odzivu Gaussovog filtra nad funkcijom skoka

Slično kao i u prošlom modelu, funkcija ***f(x)*** predstavlja intenzitet piksela na poziciji ***x***. Parametri ***h***, ***k*** i ***R*** redom odgovaraju intenzitetu pozadine, kontrastu intenziteta pozadine i predmeta te udaljenosti pravog ruba od položaja piksela sa najvećim iznosom gradijenta na zadanom dijelu linije (na slici piksel ***x=0***). Uz spomenute parametre, uvodi se dodatni parametar **σ** koji opisuje oblik (raširenost, brzinu prirasta) Gausssove funkcije. Ovakav model ruba je u [5] potkrijepljen sa profilom ruba na stvarnoj slici (slika 3.11) i bolje opisuje karakteristiku ruba od modela s funkcijom rampe.



Slika 3.11 – Profil ruba očitan s realne slike

Funkcija koja najbolje opisuje ovaj model je konvolucija funkcije skoka s Gaussovom funkcijom pri čemu se intenzitet svjetlosti za određeni položaj ***x*** tada može izraziti kao:

$$I(x) = h + \frac{k}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(t-R)^2}{2\sigma^2}} \, dt \qquad [37]$$

Pošto se kod CCD ili CMOS optičkih senzora ova kontinuirana funkcija uzorkuje i sprema u digitalnom obliku kao diskretna funkcija, intenzitet svakog piksela se izražava kao akumulacija intenziteta na intervalu čija je širina određena frekvencijom uzorkovanja:

$$\tilde{G}(i) = \int_{i-0.5}^{i+0.5} I(x) \, dx \qquad [38]$$

Kao i u modelu s funkcijom rampe, potrebno je izračunati parametar ***R*** kako bi se dobila podpiksel preciznost pri određivanju položaja ruba. Ovaj parametar bi se mogao jednostavno izračunati uvrštavanjem ostalih parametara ako su oni poznati, ali pošto su i oni nepoznanice, pristupa se traženju svih parametara.

Odabrani algoritam određivanja tih parametara je Gauss-Newtonov algoritam [6]. On se koristi za rješavanje nelinearnih problema najmanjih kvadrata. Zapravo je to modifikacja Newton-ove metode traženja minimuma funkcije. Razlika je u tome što se ne zahtijeva računanje drugih derivacija (Hesseove matrice) koje je složenije i povećava vrijeme izvršavanja, ali je zato ovaj algoritam ograničen na minimizaciju sume kvadrata funkcijskih varijabli. Glavna zadaća ovog algoritma je traženje parametara predloženog modela ruba takvih da model što bolje odgovara stvarnim intenzitetima na slici, tj. traži se najbolje nalijeganje modela na stvarne podatke.

Ciljna funkcija koja se minimizira je zapravo suma kvadrata pogreške pri čemu je pogreška predstavljena kao odstupanje modela od stvarnih podataka za svaki promatrani piksel:

$$\Delta = \sum_{i=-a}^{a} [G(i) - \tilde{G}(i)]^2 \qquad [39]$$

gdje $\tilde{\boldsymbol{G}}(\boldsymbol{i})$ predstavlja procjenu intenziteta piksela na temelju opisanog modela, a $\boldsymbol{G}(\boldsymbol{i})$ stvarni iznos intenziteta na slici. Dodatno se uvodi parametar $\boldsymbol{a}$, koji određuje širinu promatranog područja na liniji, tj. broj piksela koji se koriste pri nalijeganju modela. Ovim postupkom lokalizacija ruba postaje višedimenzionalni problem optimizacije koji se može definirati kao:

$$minimiziraj[\Delta(R, \sigma, h, k)] \qquad [40]$$

Rješavanju ovako opisanog problema se sada može pristupiti koristeći spomenuti Gauss-Newtonov algoritam. Općeniti algoritam započinje inicijalnom procjenom traženih parametara a zatim se oni iterativno poboljšavaju dok se ne zadovolji uvjet zaustavljanja. Jednadžbe [41-43] prikazuju općenitu funkcija cilja, vektor s parametrima te funkciju iterativnog postupka nalijeganja parametara.

$$C(\vec{V}) = \sum_{i=1}^{m} f_i^2(\vec{V}) \qquad [41]$$

$$\vec{V} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \qquad [42]$$

$$\vec{V}^{(k+1)} = \vec{V}^{(k)} + \overrightarrow{\Delta V} \qquad [43]$$

Vektor $\vec{\boldsymbol{V}}^{(\boldsymbol{k})}$ predstavlja vektor s traženim parametrima u iteraciji $\boldsymbol{k}$, izraz $\boldsymbol{f_i^2}(\vec{\boldsymbol{V}})$ predstavlja kvadrat funkcija koje se nastoje minimizirati, a $\overrightarrow{\Delta \boldsymbol{V}}$ predstavlja vektor „popravaka“ za svaki od parametara. Pošto $\overrightarrow{\Delta \boldsymbol{V}}$ sadrži male iznose pomaka, može se zapisati sljedeće:

$$C\left(\vec{V}^{(k)}+\overrightarrow{\Delta V}\right)=C\left(\vec{V}^{(k)}\right)+\left[\frac{\partial C\left(\vec{V}^{(k)}\right)}{\partial \vec{V}_i}\right]^T \overrightarrow{\Delta V}+\frac{1}{2}\overrightarrow{\Delta V}^T\left[\frac{\partial^2 C\left(\vec{V}^{(k)}\right)}{\partial \vec{V}_i \partial \vec{V}_j}\right]\overrightarrow{\Delta V} \quad [44]$$

$$J_f=\begin{bmatrix}\frac{\partial f_1(\vec{V})}{\partial v_1} & \cdots & \frac{\partial f_1(\vec{V})}{\partial v_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial f_m(\vec{V})}{\partial v_1} & \cdots & \frac{\partial f_m(\vec{V})}{\partial v_n}\end{bmatrix} \quad [45]$$

Ako se u jednadžbu [44] uvrsti Jacobijeva marica prikazana u [45] te ako se Hessianova matrica drugih parcijalnih derivacija aproksimira s $\boldsymbol{J_f}^{\boldsymbol{T}}\boldsymbol{J_f}$, dolazi se do:

$$C\left(\vec{V}^{(k)}+\overrightarrow{\Delta V}\right)\approx C\left(\vec{V}^{(k)}\right)+f\left(\vec{V}^{(k)}\right)^T J_f\overrightarrow{\Delta V}+\frac{1}{2}\overrightarrow{\Delta V}^T {J_f}^T J_f\overrightarrow{\Delta V} \quad [46]$$

pri čemu je $\boldsymbol{f}(\vec{\boldsymbol{V}}^{(\boldsymbol{k})})$ vektor funkcija $\boldsymbol{f_i}(\vec{\boldsymbol{V}}^{(\boldsymbol{k})})$, a $\boldsymbol{J_f}$ Jacobijeva matrica od $\boldsymbol{f}(\vec{\boldsymbol{V}})$ (eksponent T predstavlja operaciju transponiranja). Zatim se deriviranjem i izjednačavanjem s nulom traži minimum ove jednadžbe čime se dolazi do konačnog izraza za izračun vektora „popravka“:

$$C'\left(\vec{V}^{(k)}+\overrightarrow{\Delta V}\right)\approx\left(f\left(\vec{V}^{(k)}\right)^T J_f\right)^T+{J_f}^T J_f\overrightarrow{\Delta V}=0 \quad [47]$$

$$\overrightarrow{\Delta V}=\left({J_f}^{\mathrm{T}}J_f\right)^{-1}\left(-{J_f}^{\mathrm{T}}\right)f(\vec{V}^{(k)}) \quad [48]$$

Dodatno se ovaj postupak može pojednostaviti ako problem koji se rješava spada u domenu nalijeganja parametara, tj. gdje je funkcija $\boldsymbol{f_i}(\vec{\boldsymbol{V}}^{(\boldsymbol{k})})$ tipa:

$$f_i\left(\vec{V}^{(k)}\right)=y_i-g(x_i,\vec{V}^{(k)}) \quad [49]$$

Pojednostavljenje se odnosi na to da se računa nešto jednostavnija Jacobijeva matrica nad funkcijom $\boldsymbol{g}(\boldsymbol{x_i},\vec{\boldsymbol{V}}^{(\boldsymbol{k})})$. Tada se vektor $\overrightarrow{\Delta \boldsymbol{V}}$ računa kao:

$$\overrightarrow{\Delta V}=\left({J_g}^{\mathrm{T}}J_g\right)^{-1}\left({J_g}^{\mathrm{T}}\right)f(\vec{V}^{(k)}) \quad [50]$$

Ovaj algoritam s pojednostavljenjem se bez poteškoća može primijeniti na zadani model. Pri tome se kao ciljna funkcija i vektor parametara koriste izrazi [39] i [40]. Jacobijeva matrica se može prikazati na sljedeći način (parametar $2a$ predstavlja širinu promatranog područja, tj. broj piksela nad kojima se provodi postupak nalijeganja):

$$J_{\mathrm{g}} = \frac{\partial \tilde{G}\left(\vec{V}^{(k)}\right)}{\partial \vec{V}} = \begin{bmatrix} \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial R} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial \sigma} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial h} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial k} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial R} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial \sigma} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial h} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial k} \end{bmatrix} \qquad [51]$$

Parcijalne derivacije prikazane u [51] se mogu analitički odrediti, ali pošto se radi o parcijalnoj derivaciji složenog dvostrukog integrala, za analitičko određivanje rješenja korišteni su alati *Wolfram Mathematica* i Matlabov *Symbolic Toolbox*. Nakon analitičkog formuliranja Jacobijeve matrice konačno se dolazi do sljedećih koraka algoritma:

1) Inicijalno se postavljaju početne procjene parametara:

$$\vec{V}^{(0)} = \begin{bmatrix} R^{(0)} \\ \sigma^{(0)} \\ h^{(0)} \\ k^{(0)} \end{bmatrix} \qquad [52]$$

2) Na temelju trenutnih parametara i indeksa promatranog piksela računaju se elementi Jacobijeve matrice:

$$J_g = \begin{bmatrix} j_{-a,1} & \cdots & j_{-a,4} \\ \vdots & \ddots & \vdots \\ j_{a,1} & \cdots & j_{a,4} \end{bmatrix} \qquad [53]$$

3) Zatim se na temelju Jacobijeve matrice, transponirane Jacobijeve matrice i inverza umnoška tih dviju matrica računa konačna matrica koja množi vektor trenutnih rezultata pogreške fitanja za pojedini piksel koji se razmatra:

$$\overrightarrow{\Delta V} = \left(J_g{}^T J_g\right)^{-1}\left(J_g{}^T\right)F(\vec{V}^{(k)}) \quad [54]$$

$$F\left(\vec{V}^{(k)}\right) = \begin{bmatrix} (G(-a) - \tilde{G}(-a)) \\ \vdots \\ (G(a) - \tilde{G}(a)) \end{bmatrix} \quad [55]$$

4) Nakon što se izračuna $\overrightarrow{\Delta \boldsymbol{V}}$, računaju se novi iznosi parametara, pri čemu $\boldsymbol{\lambda}$ predstavlja korak popravka (u osnovnom algoritmu vrijedi $\boldsymbol{\lambda} = \mathbf{1}$):

$$\vec{V}^{(k+1)} = \vec{V}^{(k)} + \lambda\overrightarrow{\Delta V} \quad [56]$$

5) Na kraju svake iteracije se provjerava uvjet zaustavljanja koji je izveden u više varijanti, tj. algoritam staje ako je zadovoljen neki od ovih uvjeta:

a) ***R*** se prestao mijenjati:

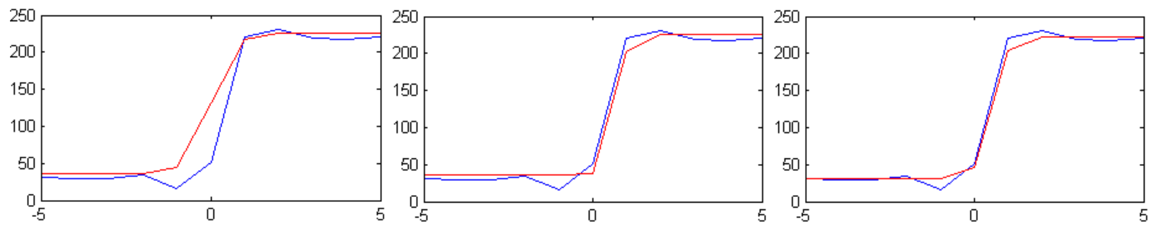
$$\left|R^{(k+1)} - R^{(k)}\right| < \varepsilon \quad [57]$$

b) Prošlo je više iteracija od dopuštenog broja iteracija

c) Tokom cijelog postupka se pamti ***R*** za koji je ispitna funkcija minimalna, kako bi se taj iznos vratio ako postupak kojim slučajem divergira.

Kako bi postupak brže i točnije konvergirao, pristupilo se određivanju parametra $\boldsymbol{\lambda}$ korištenjem postupka zlatnog reza [7]. Spomenuti postupak je implementiran kako bi se optimalno odredila veličina koraka u smjeru gradijenta $\overrightarrow{\Delta \boldsymbol{V}}$.

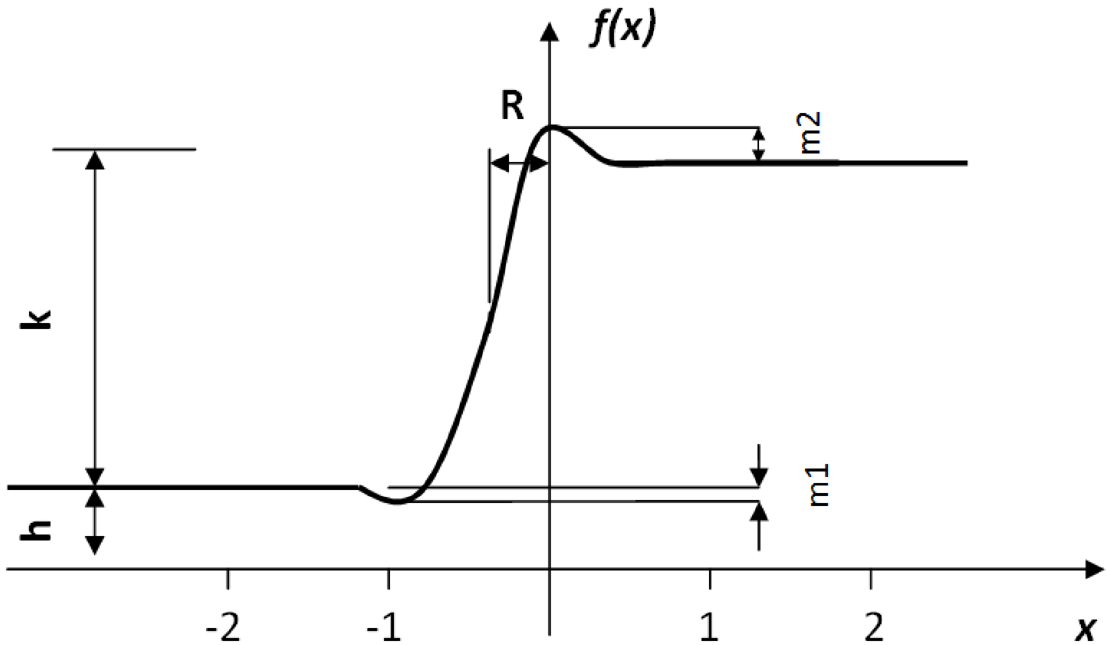
Metoda vraća vrijednost parametra ***R*** za koji je funkcija minimalna, tj. konačni parametar ***R*** kod ostvarivanja konvergencije algoritma. Kao ulaz u ovaj algoritam se predaje ispitna linija sa grubo određenom pozicijom ruba, najveći dopušteni broj iteracija te željena preciznost parametra ***R*** koja je označena kao parametar $\boldsymbol{\varepsilon}$. Primjer rezultata nalijeganja modela na stvarne podate je prikazan na slici 3.12.



Slika 3.12 – Primjer nalijeganja modela (crvena linija) na stvarne podatke (plava linija), prva, druga i treća iteracija

3.1.5. Metoda sa složenim modelom ruba temeljenom na odzivu Gaussovog filtra nad funkcijom skoka

Nakon testiranja metode opisane u poglavlju 3.1.4, uočeno je da Gauss-Newtonov postupak u nekim slučajevima teško dolazi do konvergencije. Rezultat toga je u nepreciznom modelu ruba. Model ruba uočen na ispitnim snimkama je prikazan u poglavlju 3.3, tj. na slici 3.19. Radi ove pojave, pristupilo se promjeni ulaznog modela ruba dok je ostatak postupka ostao isti.



Slika 3.13 – Složeniji model ruba

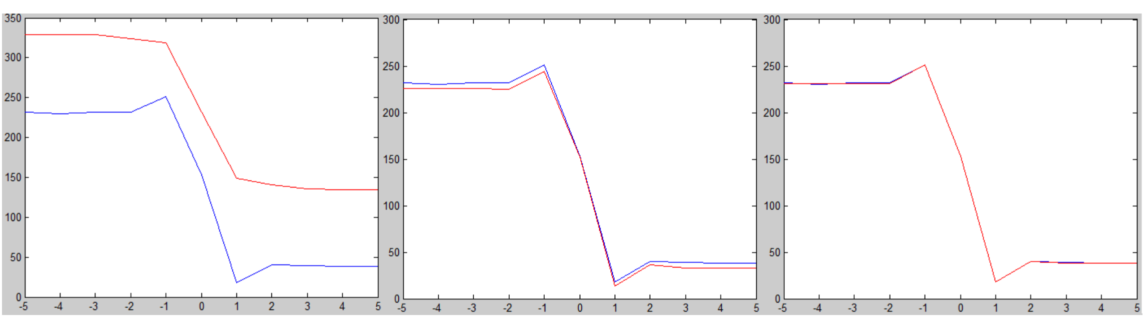
Novi model ruba je prikazan na slici 3.13 na kojoj je vidljivo da su dodana dva nova parametra ($\boldsymbol{m1}$ i $\boldsymbol{m2}$). Ta dva parametra se isto tako traže opisanom Gauss-Newton metodom pri čemu su jedine promjene vidljive u početnoj ispitnoj funkciji $\boldsymbol{I(x)}$, veličini početnog vektora s parametrima i broju stupaca Jacobijeve matrice:

$$I(x) = h - \frac{m1}{\sigma_m\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2\sigma_m^2}} + \frac{m2}{\sigma_m\sqrt{2\pi}} e^{-\frac{(x+1)^2}{2\sigma_m^2}} + \frac{k}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(t-R)^2}{2\sigma^2}} dt \qquad [58]$$

$$\vec{V}^{(t)} = \begin{bmatrix} R^{(t)} \\ \sigma^{(t)} \\ h^{(t)} \\ k^{(t)} \\ m1^{(t)} \\ m2^{(t)} \end{bmatrix} \qquad [59]$$

$$J_g = \begin{bmatrix} \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial R} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial \sigma} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial h} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial k} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial m1} & \frac{\partial \tilde{G}_{-a}(\vec{V})}{\partial m2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial R} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial \sigma} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial h} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial k} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial m1} & \frac{\partial \tilde{G}_{a}(\vec{V})}{\partial m2} \end{bmatrix} \qquad [60]$$

Parametar $\boldsymbol{\sigma_m}$ je izostavljen iz postupka traženja i postavljen je na vrijednost $\boldsymbol{\sigma_m = 0.05}$, iz razloga što se anomalija pojavljuje samo na uskom području širine jednog piksela. Ostatak postupka je identičan već opisanom. Zbog boljeg odgovaranja modela rubovima na slici, lakše se postiže konvergencija postupka:

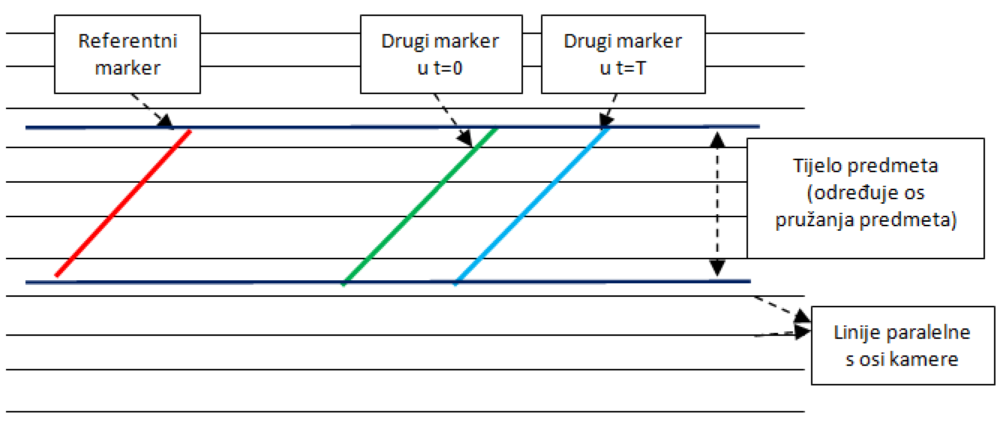


Slika 3.10 – Prikaz konvergencije postupka (prva, druga i peta iteracija), plava linija su stvarni podaci a crvena predstavlja model

3.2. Analiza utjecaja poravnanja kuta osi predmeta i osi kamere

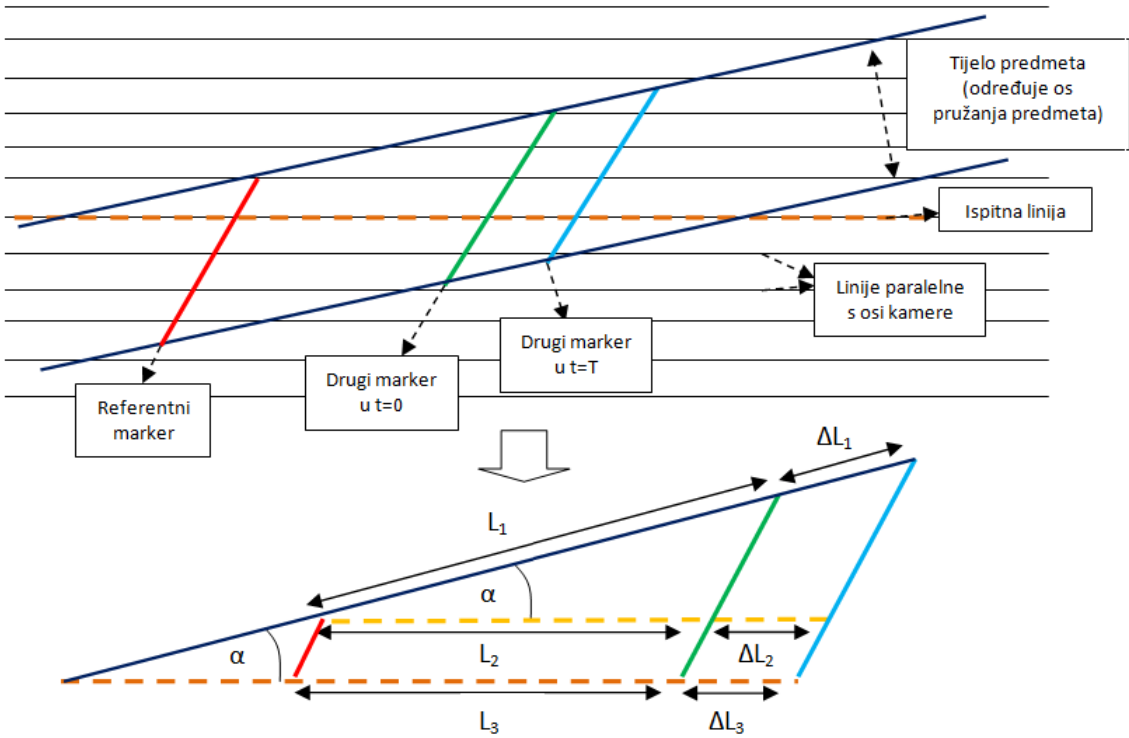
Preciznost postupka ovisi o više faktora koji su pod kontrolom osobe koja postavlja opisani laboratorijski sustav. Neki od tih parametara su npr.: postavljanje i vrsta osvjetljenja, poravnanje osi kamere s osi ispitnog objekta, postavljanje parametara kamere i objektiva (uvećanje, fokus, kontrast) ili recimo pozadina iza ispitnog objekta. Pravilnim osvjetljenjem i postavljanjem kontrastne pozadine te fukusa može se postići bolji kontrast na prijelazima svjetlo-tama, a što je razlika tamnih i svijetlih piksela markera veća time je i preciznost postupka bolja. Uvećanje (zoom) utječe na veličinu predmeta na slici, a time i na broj piksela koji predstavljaju predmet, čime se dobiva veća rezolucija slike predmeta, a time i veća preciznost mjerenja udaljenosti markera. Utjecaj poravnanja osi predmeta i osi kamere nije toliko intuitivno jasan, stoga se pristupa analizi tog utjecaja.

Ako se pretpostavi da je ispitni predmet ravninski (ravna ploha sa pravocrtnim kontrastnim markerima, slika 3.15), tada se izračun utjecaja svodi na sličnost trokuta i pokazuje se da devijacije kuta između osi predmeta i osi kamere nemaju utjecaj na preciznost mjerenja.



Slika 3.15 – Ravninski objekt sa paralelnim markerima

Na sljedećoj slici se vidi predmet čija se os ne poklapa sa osi kamere te se tu naziru slični trokuti kojima se dokazuje ne postojanje utjecaja kuta između dviju osi:



Slika 3.16 – Ravninski objekt pod kutom u odnosu na os kamere

Iz ovako postavljenog sustava se bez poteškoća može pokazati sličnost trokuta, a time i dokazati da kod ravninskih predmeta kut poravnanja osi kamere i predmeta nema utjecaj na iznos mjerenja:

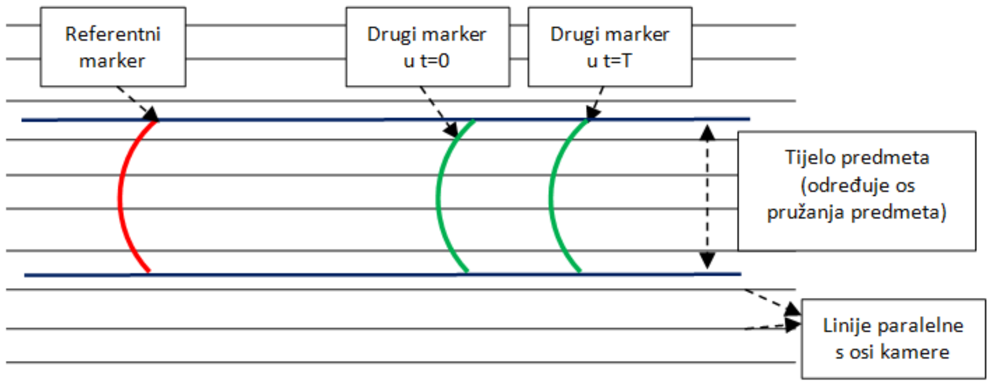
$$L_2 = L_3, \qquad \Delta L_2 = \Delta L_3 \quad [61]$$

$$\frac{L_2 + \Delta L_2}{L_2} = \frac{L_3 + \Delta L_3}{L_3} \quad [62]$$

$$L_3 L_2 + L_3 \Delta L_2 = L_2 L_3 + L_2 \Delta L_3 \quad [63]$$

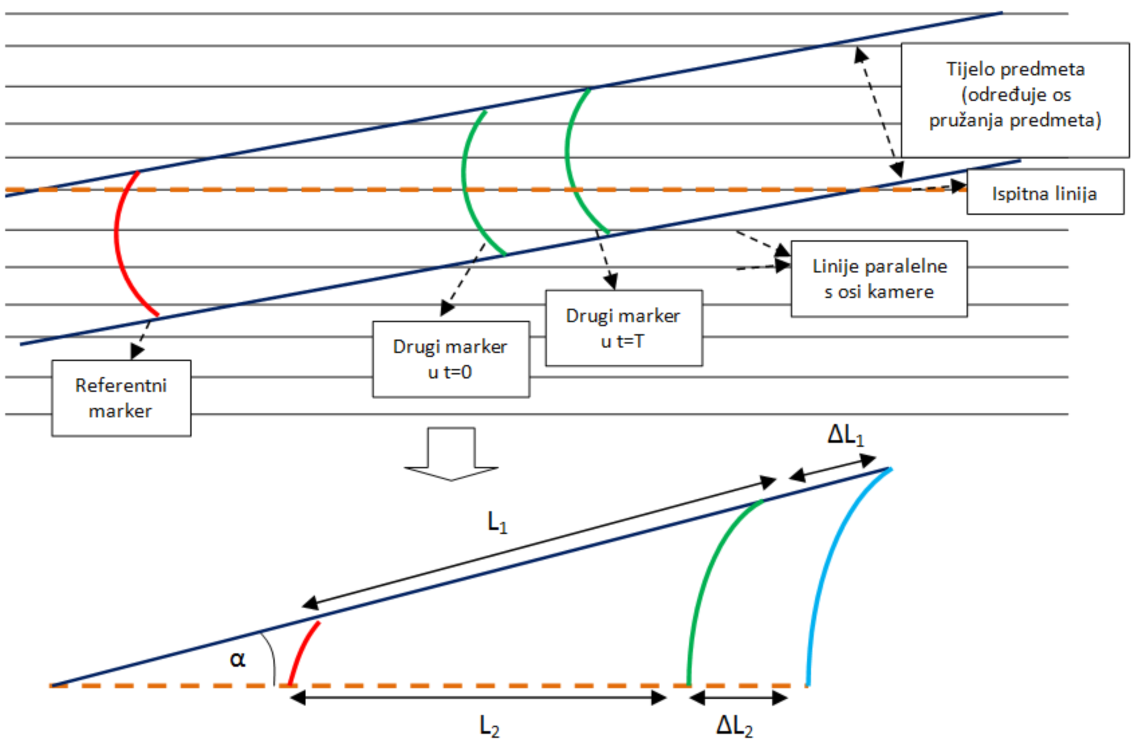
$$\frac{\Delta L_2}{L_2} = \frac{\Delta L_3}{L_3} = \frac{\Delta L_1}{L_1} \quad [64]$$

S druge strane, ako se u obzir uzmu zakrivljeni, tj. obli predmeti, izračun se komplicira i pokazuje se da postoji utjecaj. Pravocrtne linije zbog perspektivne transformacije postaju eliptične tako da se dosta teže određuje utjecaj odstupanja kuta osi predmeta i kamere.



Slika 3.17 – Zaobljeni predmet (npr. cijev) paralelan s osi kamere

Na analizu u ovom slučaju utječe mnogo više parametara. U obzir se uzimaju tri jednadžbe elipse i kut između osi predmeta i osi kamere:



Slika 3.18 – Zaobljeni predmet pod kutom u odnosu na os kamere

Kako bi se lakše izračunao utjecaj i konačna maksimalna pogreška, sustav elipsa i pravca pod kutom su postavljeni u ishodište koordinatnog sustava pri čemu elipsa koja predstavlja referentni marker prolazi ishodištem i centri svih elipsa su na osi apscise. Sustav jednadžbi tada glasi:

$$y = x * tg(\alpha) \tag{65}$$

$$b_1^2(x - p_1)^2 + a_1^2 y^2 = a_1^2 b_1^2 \tag{66}$$

$$b_2^2(x - p_2)^2 + a_2^2 y^2 = a_2^2 b_2^2 \tag{67}$$

$$b_3^2(x - p_3)^2 + a_3^2 y^2 = a_3^2 b_3^2 \tag{68}$$

Rješenja ovog sustava predstavljaju presjecišta pravca pod kutem i elipsa. Na temelju tih točaka mogu se odrediti duljine $\boldsymbol{L_1}$ i $\Delta\boldsymbol{L_1}$. Druge dvije duljine $\boldsymbol{L_2}$ i $\Delta\boldsymbol{L_2}$ se mogu direktno odrediti iz parametara $\boldsymbol{a_i}$ i $\boldsymbol{p_i}$ svake od jednadžbi elipse. Pogreška koja se ovim načinom može izračunati je predstavljena kao pogreška postotka izduljenja predmeta:

$$error = \left|\frac{\Delta L_1}{L_1} - \frac{\Delta L_2}{L_2}\right| \tag{69}$$

Pošto izračun pogreške uključuje složeni model i dosta računanja, implementirana je simulacija u programskom jeziku Matlab. U toj simulaciji se zadaju parametri elipsa i kut pravca, a rezultat je spomenuta pogreška. Za konačnu ocjenu utjecaja simulacija je pokrenuta sa parametrima zateknutim na trenutnim snimkama na kojima se provodi implementacija ovog rada:

$$\alpha = 0.25° = 0°15`$$

$$a_1 = a_2 = a_3 = 1.91$$

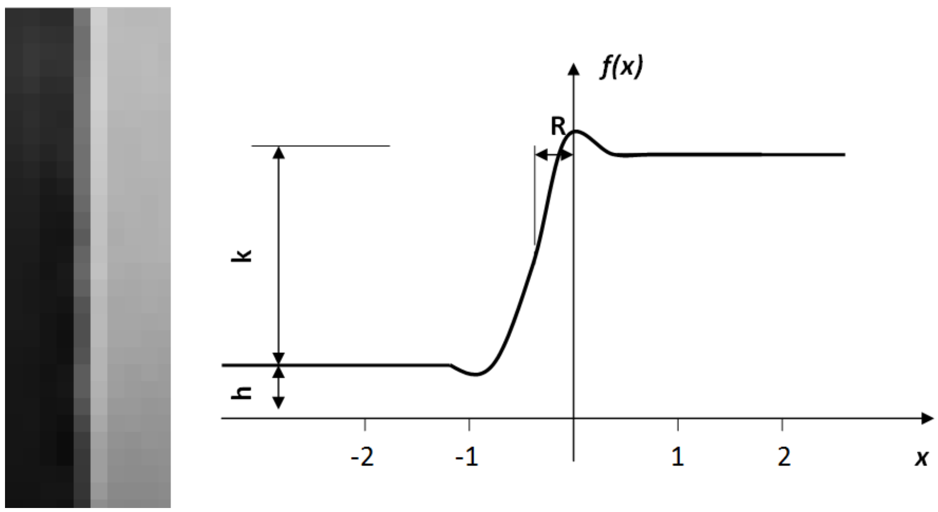
$$b_1 = b_2 = b_3 = 6.05$$

$$p_1 = a_1, \qquad p_2 = p_1 + 25, \qquad p_3 = p_1 + 26$$

Kao rezultat je dobivena relativna pogreška $\boldsymbol{error} = \mathbf{5.2} * \mathbf{10^{-7}}$, a to znači da se na predmet duljine 25mm uz ovaj kut otklona može očekivati pogreška mjerenja od $\mathbf{0.013}\boldsymbol{\mu m}$, što je u okviru ovoga rada i više nego dovoljna preciznost spram očekivane preciznosti.

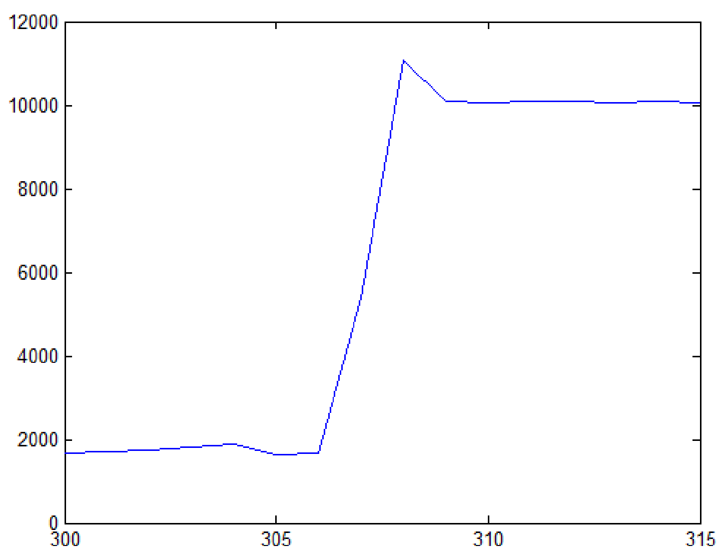
3.3. Mogući utjecaji na preciznost

Uz već spomenute utjecaje u prošlom poglavlju, nameće se još jedan. Mogući razlog gubitka na preciznosti je neprecizni model ruba, tj. realni rub na trenutnim snimkama (slika 3.19) se razlikuje od dva predstavljena modela ruba u poglavljima 3.1.3. i 3.1.4.



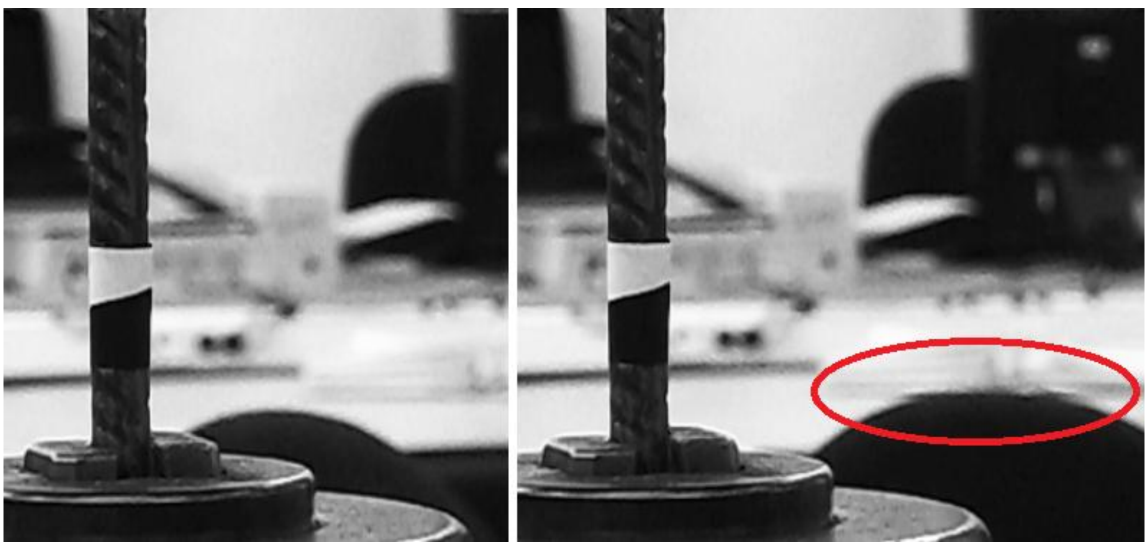
Slika 3.19 – Detalj ruba sa realne slike (lijevo) i model ruba desno

Razlog ove pojave je postprocesiranje na strani kompaktnog fotoaparata, kako bi se naglasio kontrast na rubovima. Stvarna karakteristika ruba je dana na slici (slika 3.20).



Slika 3.20 – Karakteristika ruba (suma intenziteta za 16 piksela, svaki po 66 linija)

Kasnije je uočena još jedna anomalija na snimkama koja služi kao dokaz postojanja nesavršenosti leće. Primjer ove anomalije je vidljiv na slici 3.21. Ovim primjerima je pokazano da je za što preciznija mjerenja potrebna industrijska kamera i leća na kojima su ovakve deformacije manje vjerojatne. Potrošačka kamera kao ova je namijenjena širokoj primjeni i kao takva dolazi sa širokim spektrom funkcija, ali isto tako unosi i deformacije slike koje su ljudskom oku nevidljive ili pak nisu od interesa. Industrijske kamere su detaljno provjerene i testirane te je za njih dostupna velika količina dokumentacije u kojoj su navedene precizne mjere i tolerancije vezane za moguće deformacije (poput radijalne distorzije, tj. TV distorzije).



Slika 3.21 – Normalna slika (lijevo) i slika s deformacijom (desno, horizontalna deformacija unutar područja crvenog markera)

4. Programska implementacija

Konačna implementaciji rješenja je izvedena u programskom jeziku c++ uz korištenje OpenCV i Boost biblioteka. Za prikazivanje slika te operacije nad slikama kao i za matrične operacije je korišten OpenCV, dok je Boost korišten za računanje posebnih funkcija kao što je `erf()` funkcija.

Kôd algoritma je raspodijeljen na četiri komponente. Glavna, odnosno ulazna komponenta je `MeasurementSolution.cpp` u kojoj se nalazi funkcija `main()`. U njoj se nalaze funkcije za inicijalizaciju ROI-a, glavna funkcija za računanje i bilježenje položaja rubova, prikaz video sekvence te osvježavanje prikazane slike s markerima. Ovdje se isto tako iterira kroz glavnu petlju koja prolazi svim slikama iz video sekvence, za svaku sliku se bilježe položaji rubova kao i duljina ispitnog predmeta te se kao rezultat iscrtava slika s izračunatom duljinom ispitnog predmeta. Spomenute funkcije su redom:

```
// Funkcija inicijalizira ROI na prvoj ulaznoj slici.
void on_mouse(int event, int x, int y, int flags, void* param);

// Funkcija priprema svaku ulaznu sliku za daljnju obradu.
void prepInputImage(bool video, int frame_idx, CvVideoWriter*  writer, double
       last_elongation, ofstream &outfile, CvFont font);

// Funkcija za dohvat položaja rubova na pojedinoj slici.
vector<vector<double>> getEdgePositions(BwImage &imgA, BwImageFloat &imgB);

// Glavna funkcija za bilježenje izduljenja ispitnog predmeta na video
sekvenci.
void measure(char* str);
```

U glavnoj komponenti su isto tako deklarirani svi parametri implementiranih metoda koji se kasnije prenose kao parametri funkcija (tablica 4.1).

Parametar	Opis
NUM_OF_MEM_IMGS	Broj slika na kojima se bilježe izmjerene udaljenosti i provodi njihovo usrednjavanje
LINES	Broj linija uzduž kojih se računaju pomaci na pojedinoj slici
SIGMA	Parametar zaglađivanja ispitne slike
KERNEL_SIZE	Širina promatranog područja u okviru kojeg se estimira položaj ruba
USE_GAUSSIAN_EDGE	Odabir metode precizne lokalizacije ruba
NUM_OF_ITERATIONS	Najveći dopušteni broj iteracija gradijentnog postupka
EPSILON	Željena preciznost određivanja položaja ruba
MAX_NUM_GOLDEN_SECTION_ITERATIONS	Najveći dopušteni broj iteracija postupka zlatnog reza

Tablica 4.1 – Popis parametara implementiranih metoda koje su deklarirane u glavnoj komponenti

Funkcija `getEdgePositions()` na temelju opisanih parametara poziva metode precizne lokalizacije te pomoćne funkcije i strukture koje se nalaze u datotekama:

1) `RampModel.cpp`
2) `GaussianModel.cpp`
3) `HelpingFunctions.cpp`

Glavne funkcije su:

```
// Funkcija precizne lokalizacije ruba s modelom funkcije rampe.
double localizeEdgeRampModel(BwImage &lines, int max_idx, int line_idx, int
      kernel_sz);

// Funkcija precizne lokalizacije ruba temeljena na modelu ruba s Gaussovom
funkcijom.
double localizeEdgeGaussianModel(BwImage &lines, int line_idx, int edge_idx,
      int max_Iterations, int kernel_sz, double epsilon,
      int golden_section_iter);

// Funkcija precizne lokalizacije ruba temeljena na složenijem modelu ruba s
Gaussovom funkcijom.
double localizeEdgeGaussianModelComplex(BwImage &lines, int line_idx, int
edge_idx,
      int max_Iterations, int kernel_sz, double epsilon,
```

ruba temeljenim na funkciji rampe je jednostavna, a njena funkcionalnost je opisana u poglavlju 3.1.3. Implementacija metode s modelom ruba temeljenim na odzivu Gaussovog filtra nad funkcijom skoka se nalazi u datoteci `GaussianModel.cpp`, a dodatno poziva i sljedeće funkcije koje su smještene u istoj datoteci:

```
// Funkcija evaluira ciljnu funkciju i kao rezultat daje ukupnu pogrešku
modela s Gausssovom funkcijom u odnosu na stvarne podatke.
double evalFuncGN(BwImage &lines, int line_idx, int edge_idx, int kernel_sz,
       double R, double Sigma, double h, double k);

// Funkcija provodi postupak zlatnog reza nad parametrom lambda. Rezultat je
lambda za koji je pogreška modela u odnosu na stvarne podatke minimalna.
double goldenSectionSearch(BwImage &lines, int line_idx, int edge_idx, int
       kernel_sz, double last_R, double last_Sigma, double last_h, double
       last_k, CvMat* delta, int numberOfIterations);

// Evaluacijska funkcija ali za složeniju funkciju ruba zasnovanu na
Gaussovoj funkciji.
double evalFuncGN(BwImage &lines, int line_idx, int edge_idx, int kernel_sz,
       double R, double Sigma, double h, double k);

// Funkcija provodi line-search algoritam zlatnog reza, ali na modelu ruba sa
složenijom Gaussovom funkcijom.
double goldenSectionSearch(BwImage &lines, int line_idx, int edge_idx, int
       kernel_sz, double last_R, double last_Sigma, double last_h, double
       last_k, CvMat* delta, int numberOfIterations);
```

Svaka od ove četiri datoteke ima pripadajuću *header* datoteku u skladu s konvencijom o pravilnom strukturiranju kôda.

5. Eksperimentalni rezultati

5.1. Organizacija rezultata

Jedini kriterij vrednovanja pri testiranju ovog algoritma je preciznost mjerenja. Pojedino mjerenje odgovara izmjerenoj udaljenosti između markera nalijepljenih na ispitnom objektu, dok se pojam preciznost mjerenja odnosi na podatak o odstupanju te izmjerene udaljenosti od stvarne, deklarirane udaljenosti. Uz spomenuto odstupanje mjeri se standardna devijacija više mjerenja izmjerenih na video sekvenci, kako bi se dobio podatak o oscilacijama mjerenja.

Prilikom izračuna udaljenosti između markera potrebno je odrediti položaje rubova markera. Ti položaji se prvo određuju grubom lokalizacijom, koja se provodi traženjem ekstrema gradijenata slike i precizna je do reda veličine jednog piksela. Nakon određivanja grubog položaja, pristupa se metodi precizne lokalizacije ruba koja računa podpikselski položaj tog ruba. Implementirane su i usporedno testirane dvije metode precizne lokalizacije koje su objašnjene u poglavljima 3.1.3 i 3.1.4:

1) Metoda s modelom ruba temeljenim na funkciji rampe (dalje u tekstu se koristi naziv: metoda s modelom ruba rampe)

2) Metoda s modelom ruba temeljenim na odzivu Gaussovog filtra nad funkcijom skoka (dalje u tekstu se koristi naziv: metoda s Gaussovim modelom ruba)

Osnovni algoritam sadrži pet parametara koji utječu na mjerenja:

- **a)** $\boldsymbol{nS}$ - Broj slika na kojima se bilježe izmjerene udaljenosti i provodi njihovo usrednjavanje
- **b)** $\boldsymbol{nL}$ - Broj linija uzduž kojih se računaju pomaci na pojedinoj slici
- **c)** $\boldsymbol{\sigma_I}$ - Parametar zaglađivanja ispitne slike
- **d)** $\boldsymbol{2a+1}$ - Širina promatranog područja u okviru kojeg se estimira položaj ruba
- **e)** Odabir metode precizne lokalizacije ruba

Ako se kao metoda precizne lokalizacije ruba odabere metoda s Gaussovim modelom ruba, tada se uvode dodatni parametri:

- **f)** $\boldsymbol{MaxIter}$ - Najveći dopušteni broj iteracija gradijentnog postupka
- **g)** $\boldsymbol{\varepsilon}$ - Željena preciznost određivanja položaja ruba
- **h)** $\boldsymbol{MaxGSIter}$ – Najveći dopušteni broj iteracija postupka zlatnog reza

Testiranje preciznosti se provodi na ispitnim snimkama s mikrovijkom. Na mikrovijak se nalijepe markeri, inicijalno se odredi udaljenost između njih (kalibracija), a zatim se na video sekvenci prate njihovi pomaci. Precizno određeni pomaci se izvode mikrovijkom s koracima $1mm$, $1mm$ i $2mm$. Kako se na ispitni predmet (u ovom slučaju mikrovijak) markeri ručno lijepe, njihova inicijalna udaljenost se mora precizno odrediti. Pošto je poznat korak pomaka mikrovijka, na temelju njega se može odrediti prava početna udaljenost između markera. Ovaj postupak se provodi tako da se iterativno korigira početna kalibracija dok se prvi korak izmjeren na video sekvenci ne poklopi sa pomakom od $1mm$.

Objekt se na ispitnoj video sekvenci nalazi u području širine ≈ 146 piksela u smjeru izduženja mredmeta. Ovaj podatak je važan jer daje uvid u relativni odnos duljine i broja piksela na slici. Početna kalibrirana udaljenost se kreće oko $\approx 24.5mm$, što znači da je preciznost na razini piksela $\approx 167.8\mu m$. Ovo je dobar procjenitelj preciznosti ostvarenih metoda, jer se s njim može pokazati faktor povećanja preciznosti za pojedinu metodu.

Primjer rezultata mjerenja na video sekvenci za metodu s Gaussovim modelom ruba je prikazan u tablici 5.1. Parametri su: $\boldsymbol{nS = 16}$, $\boldsymbol{nL = 3}$, $\boldsymbol{\sigma_I = 1.0}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.32992277	24.3302814	-0.358628	1.162315
2	25.33202871	25.3302814	1.747313	1.055399
3	26.33520495	26.3302814	4.92355	1.011768
4	28.32738317	28.3302814	-2.898232	0.784738

Tablica 5.1 – Primjer rezultata za video sekvencu uz pomake od 1+1+2mm

Srednja vrijednost prikazana u tablici predstavlja srednju vrijednost stotinu mjerenja izračunatih na slikama (odnosno izračunatih u vremenu) na kojima ispitni predmet miruje, tj. ne izdužuje se. Očekivana vrijednost je zbroj kalibrirane vrijednosti i deklariranog pomaka mikrovijka. Iz ovog mjerenja se može zaključiti da je najveća pogreška vidljiva na mjerenju odstupanje od $4.924\mu m$, dok su najmanje zabilježeno i srednje odstupanje redom $1.747\mu m$ i $3.190\mu m$ (u izračun odstupanja se ne ubraja prvo mjerenje jer ono odgovara stacionarnom mjerenju nakon kalibracije kad se predmet još nije izduljio).

5.2. Usporedba dviju metoda precizne lokalizacije ruba

Pošto su razvijene dvije metode precizne lokalizacije ruba, potrebno ih je i usporedno testirati. Za parametre algoritma su korišteni isti parametri u oba testa: $\boldsymbol{nS = 16}$, $\boldsymbol{nL = 3}$, $\boldsymbol{\sigma_I = 0.5}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.35315248	24.35333083	-0.178355	1.104
2	25.35205149	25.35333083	-1.279345	0.909
3	26.35445941	26.35333083	1.128576	1.109
4	28.35420099	28.35333083	0.87016	1.153

Tablica 5.2 – Rezultati mjerenja izvedenih metodom s modelom temeljenim na modela ruba s funkcijom rampe

U tablici 5.2 su vidljivi rezultati mjerenja za metodu s funkcijom rampe, dok su u tablici 5.3 prikazani rezultati za metodu s Gaussovom funkcijom:

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.32442475	24.3245264	-0.101648	0.947144
2	25.32694059	25.3245264	2.414194	0.944582
3	26.33123267	26.3245264	6.706273	1.0047
4	28.32042475	28.3245264	-4.101648	0.771804

Tablica 5.3 - Rezultati mjerenja s metodom modela ruba temeljenom na odzivu Gaussovog filtra nad funkcijom skoka

Već na prvi pogled je vidljivo da je metoda s funkcijom rampe preciznija od složenije metode s Gaussovom funkcijom. Srednja odstupanja prve i druge metode su $1.093\mu m$ i $4.407\mu m$, što vodi do činjenice da je metoda s funkcijom rampe više nego četiri puta preciznija. Standardne devijacije povezane uz obje metode su gotovo iste. Kako bi se dodatno provjerila ispravnost implementacije metode, ista metoda je provedena u Matlab-u, ali s korištenjem Matlab-ovih ugrađenih funkcija nelinearne optimizacije. Usporedni rezultati implementacije u c++ i implementacije u Matlab-u su prikazani u tablicama 5.4 i 5.5. Korišteni su parametri: $\boldsymbol{nS = 16}$, $\boldsymbol{nL = 3}$, $\boldsymbol{\sigma_I = 0.0}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.24545545	24.24529265	0.162796	1.148693
2	25.23784158	25.24529265	-7.451066	2.647462
3	26.23944356	26.24529265	-5.849086	1.11968
4	28.25304158	28.24529265	7.748934	10.082949

Tablica 5.4 – Rezultati mjerenja za metodu s Gaussovom funkcijom (implementacija u c++)

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.24078152	24.2409809	-0.199379	0.425465
2	25.23932755	25.2409809	-1.653356	0.80139
3	26.24356368	26.2409809	2.58278	1.703755
4	28.23320199	28.2409809	-7.778919	0.464298

Tablica 5.5 – Rezultati mjerenja za metodu s Gaussovom funkcijom (implementacija u Matlab-u)

Rezultati za obje implementacije su usporedivi, što otklanja mogućnost pogrešne implementacije u c++. Nakon ovog ispitivanja ostaje jedino mogućnost da je model neprimjereno postavljen spram podataka. Radi ispitivanja i ove mogućnosti implementiran je složeniji model ruba opisan u poglavlju 3.1.5. Rezultati opisane metode sa složenijim modelom ruba su dani u tablici 5.6. Korišteni parametri su: $\boldsymbol{nS = 16}$, $\boldsymbol{nL = 3}$, $\boldsymbol{\sigma_I = 0.5}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Redni broj mjerenja	Srednja izmjerena duljina $[mm]$	Stvarna duljina $[mm]$	Odstupanje $[\mu m]$	Standardna devijacija $[\mu m]$
1	24.47404752	24.47253733	1.510195	1.824258
2	25.46846733	25.47253733	-4.070003	2.562893
3	26.46998119	26.47253733	-2.556142	1.706266
4	28.4769802	28.47253733	4.442868	4.070013

Tablica 5.6 - Rezultati mjerenja s metodom složenijeg modela ruba temeljenom na odzivu Gaussovog filtra nad funkcijom skoka

Najveće i srednje odstupanje su manji u odnosu na metodu s jednostavnijim modelom temeljenim na odzivu Gaussove funkcije nad funkcijom skoka i iznose $4.443\mu m$ i $3.690\mu m$ (naspram $6.706\mu m$ i $4.407\mu m$). Iz ovih podataka se da zaključiti kako složeniji model bolje odgovara karakteristici stvarnog ruba. U ostalim testovima se uz navođenje rezultata dviju opisanih metoda navode i rezultati za metodu s Gaussovom funkcijom ali uz složeniji model ruba.

5.3. Utjecaj parametra zaglađivanja

Zaglađivanje je osnovni postupak predprocesiranja ulazne slike, što znači da direktno utječe na daljnje odvijanje metoda lokalizacije ruba. U ovom djelu se iznose testovi s različitim iznosima parametra $\boldsymbol{\sigma_I}$, kako bi se utvrdio utjecaj predprocesiranja slike na konačnu preciznost opisanih metoda. Korišteni fiksni parametri su: $\boldsymbol{nS = 16}$, $\boldsymbol{nL = 3}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Korištena metoda	Srednje odstupanje uz određeni parametar σ_I (μm)			
	$\sigma_I = 0$	$\sigma_I = 0.5$	$\sigma_I = 1.0$	$\sigma_I = 1.5$
Model s funkcijom rampe	1.236	1.093	0.915	1.195
Model sa zaglađenim skokom	7.016	4.407	3.190	3.323
Složeniji model sa zaglađenim skokom	9.188	3.690	2.425	3.281
	Srednja standardna devijacija uz određeni parametar σ_I (μm)			
Model s funkcijom rampe	1.310	1.057	0.625	0.592
Model sa zaglađenim skokom	4.617	0.907	0.951	1.281
Složeniji model sa zaglađenim skokom	4.346	2.780	1.274	1.377

Tabela 5.7 – Utjecaj parametra zaglađivanja σ_I na konačnu preciznost metoda precizne lokalizacije ruba

Tablica 5.7 prikazuje srednja odstupanja opisanih metoda u ovisnosti o iznosu parametra $\boldsymbol{\sigma_I}$. Može se uočiti da su preciznosti svih metoda nešto lošije ako se postupak zaglađivanja uopće ne provodi ($\boldsymbol{\sigma_I = 0}$). Razlog tomu je prisutnost šuma na slici, koji se zaglađivanjem otklanja. Isto tako se može primijetiti da se povećanjem parametra $\boldsymbol{\sigma_I}$ postižu veće preciznosti s manjim oscilacijama (manja standardna devijacija), ali samo do određene granice kada se zbog jakog zaglađivanja počinje gubiti informacija o rubu (gubi se oštrina ruba). Za metodu s Gaussovom funkcijom se pokazalo da je osjetljivija na šum od metode s funkcijom rampe. Najbolji rezultati su ostvareni sa $\boldsymbol{\sigma_I = 1.0}$, tako da se ovaj parametar koristi kod daljnjeg testiranja.

5.4. Utjecaj broja ispitnih linija

Ostvarivanje dodatne informacije za određivanje podpikselski precizne udaljenosti između markera se temelji na praćenju položaja rubova na više linija. Iz ovog razloga je potrebno ispitati koliki točno utjecaj na preciznost ima broj ispitnih linija na kojima se bilježe spomenuti pomaci. Korišteni fiksni parametri su: $\boldsymbol{nS = 16}$, $\boldsymbol{\sigma_I = 1.0}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Korištena metoda	Srednje odstupanje uz određeni broj ispitnih linija			
	3 linije	5 linija	9 linija	17 linija
Model s funkcijom rampe	0.915	0.877	0.719	1.488
Model sa zaglađenim skokom	3.190	2.331	3.524	3.849
Složeniji model sa zaglađenim skokom	2.425	2.269	2.762	2.536
	Srednja standardna devijacija uz određeni broj ispitnih linija			
Model s funkcijom rampe	0.625	0.536	0.427	0.496
Model sa zaglađenim skokom	0.951	0.891	0.733	0.797
Složeniji model sa zaglađenim skokom	1.274	1.066	0.807	1.038

Tablica 5.8 – Utjecaj broja ispitnih linija na preciznost postupka

Tablica 5.8 prikazuje konačne preciznosti za određeni broj ispitnih linija. Zamjećuje se slična pojava kao i kod testiranja utjecaja zaglađivanja: povećanjem broja ispitnih linija povećava se i preciznost, ali samo do određene granice. Što se više linija uzima u obzir, preciznost bi trebala biti bolja, a rezultati stabilniji, ali uzimanje većeg broja linija ima posljedicu da se dovodi više mogućnosti za pojavu anomalija. Može se reći da više linija unosi dodatni šum u same rezultate, pa oni postaju nepouzdaniji. Za većinu ispitanih metoda se pokazalo da je optimalni broj linija $\boldsymbol{nL = 5}$ stoga se ovaj parametar koristi u daljnjim testiranjima.

5.5. Utjecaj broja ispitnih slika

Broj ispitnih slika odgovara broju slika na kojima se bilježe pomaci i na kojima se zatim provodi usrednjavanje kako bi se dobio jedan podatak o izduljenju predmeta kojem se više vjeruje od onog na samo jednoj slici. Ovim načinom se više mjerenja s više slika usrednjuje, a time se dobivaju manje oscilacije među rezultatima te veći stupanj vjerovanja da se izmjereno izduljenje poklapa sa stvarnim izduljenjem. U tablici 5.9 su prikazane preciznosti postupka za različit broj ispitnih slika. Korišteni su parametri: $\boldsymbol{nL = 5}$, $\boldsymbol{\sigma_I = 1.0}$, $\boldsymbol{a = 5}$, $\boldsymbol{MaxIter = 50}$ i $\boldsymbol{\varepsilon = 10^{-5}}$.

Korištena metoda	Srednje odstupanje uz određeni broj slika za usrednjavanje			
	1 slika	4 slike	16 slike	32 slike
Model s funkcijom rampe	0.941	0.919	0.877	0.711
Model sa zaglađenim skokom	2.792	2.888	2.229	2.455
Složeniji model sa zaglađenim skokom	2.688	2.589	2.269	2.049
	Srednja standardna devijacija uz određeni broj slika za usrednjavanje			
Model s funkcijom rampe	1.557	0.878	0.536	0.378
Model sa zaglađenim skokom	2.666	1.563	0.891	0.588
Složeniji model sa zaglađenim skokom	3.148	1.805	1.066	0.807

Tablica 5.9 – Utjecaj broja ispitnih slika na preciznost postupka

Ovdje se zamjećuje blagi porast preciznosti s porastom broja ispitnih slika, ali je veći napredak vidljiv u pogledu stabilnosti rezultata. S povećanjem broja slika opada standardna devijacija, što znači da mjerenja sve manje osciliraju, tj. da su pouzdanija. Za parametar $\boldsymbol{nS} = \mathbf{16}$ je uočen najbolji omjer preciznosti i stabilnosti rezultata.

6. Zaključak

Predstavljen je postupak za precizno mjerenje istezanja materijala. Postupak se temelji na pretpostavci da je ispitni materijal (ispitni objekt) obilježen kontrastnim markerima te da se isteže ili skuplja pod utjecajem vlačnih ili tlačnih sila hidraulične preše. Ispitni objekt se snima jednom kamerom čija je horizontalna os paralelna s uzdužnom osi objekta. Slike pribavljene kamerom služe kao jedini ulaz opisanom postupku, dok je izlaz postupka podatak o izduljenju, tj. promjeni izduljenja ispitnog objekta.

Postupak se zasniva na dvije metode precizne lokalizacije ruba koje se temelje na dva modela ruba. Prvi model ruba se temelji na funkciji rampe gdje se položaj ruba određuje jednostavnim matematičkim operacijama. Drugi model se temelji na odzivu Gaussovog filtra nad funkcijom skoka, a parametri modela se određuju metodom gradijentne optimizacije gdje se koristi Gauss-Newtonova metoda. Za obje metode su provedeni testovi preciznosti.

Testovi su izvedeni na video sekvencama s ispitnim predmetom na kojem je početna kalibrirana udaljenost između markera $\approx 24.5mm$, pri čemu ta udaljenost na slici odgovara širini od ≈ 146 piksela što dovodi do preciznosti $\approx 167.8\mu m$ ako se radi o određivanju položaja ruba do razine piksela. Metoda precizne lokalizacije ruba temeljena na funkciji rampe se pokazala preciznijom s najmanjim srednjim odstupanjem od stvarne vrijednosti koje iznosi $\approx 0.711\mu m$, dok je složenija metoda temeljena na odzivu Gaussovog filtra nad fun$kcij$om skoka ostvarila najmanje srednje odstupanje od $\approx 2.049\mu m$.

Prosječna brzina obrade slike je za metodu s funkcijom rampe $\approx 9.622ms$, dok je za metodu s Gaussovom funkcijom $\approx 79.402ms$, što omogućava obradu ≈ 103.93 slike, tj. ≈ 12.59 slika u sekundi ($\approx 103fps$ i $\approx 12fps$). Ovakve brzinu su dostatne za korištenje postupka u stvarnom vremenu.

Uzimajući u obzir kvalitetu kamere korištene u ovom radu (u usporedbi s industrijskom kamerom visoke rezolucije), čak je i lošiji rezultat, dobiven složenijom metodom, dovoljno dobar za primjenu u ispitivanju svojstava materijala gdje je potrebna određena razina preciznosti, dok je preciznija metoda već i s ovakvom kamerom blizu preciznosti postojećih komercijalnih sustava.

Iako su ostvarene preciznosti zadovoljavajuće, postoji prostor za poboljšanja. Jedno od mogućih poboljšanja je korištenje složenijeg modela ruba koji bi u potpunosti odgovarao rubovima na slici ili korištenje industrijske kamere visoke rezolucije. Još neka od mogućih poboljšanja su korištenje gradijentne optimizacije za određivanje parametara modela ruba s funkcijom rampe, korištenje više kamera pod različitim kutovima uz postupak registracije slika ili pak korištenje više markera na ispitnim objektima.

Dodaci

Dodatak A - Tehničke specifikacije korištenog kompaktnog fotoaparata

U ovom radu je za dobivanje ispitnih video sekvenci korišten kompaktni fotoaparat Samsung PL60. Značajnije karakteristike su navedene u sljedećoj tablici:

Tip senzora	1/2.33" (približno 1.09cm) CCD
Efektivni broj piksela	Približno 10.2 Mega-piksela
Fokalna duljina leće	f = 6.3 - 31.5mm (35mm film ekvivalent: 35 - 175mm)
Digitalni zum leće	1.0X - 5.0X
Tip fokusa	TTL automatski fokus (Multi AF, Centre AF)
ISO ekvivalent ekspozicije	Auto, 80, 100, 200, 400, 800, 1,600, 3,200 (up to 3M)
Stabilizacija slike	Dual IS (optička + digitalna stabilizacija)
Video	Moguće rezolucije: 640 x 480, 320 x 240 Frame Rate: 30fps, 15fps Tip datoteke: AVI (MJPEG) Audio: WAV

Tabela 2 – Specifikacije korištene kamere

Testne snimke su rađene u rezoluciji 640x480 uz 30 slika u sekundi.

Dodatak B - Protokol komunikacije klijentskog računala s programom

Klijentsko računalo s računalom na kojem je pokrenut program komunicira preko serijskog sučelja (RS 232). Osmišljeni protokol se sastoji od jednoslovnih poruka (naredbi) koje program interpretira na sljedeće načine:

Naredba	Trenutak za slanje	Funkcija
„I“	Prije početka mjerenja	Pokreće proces traženja markera
„S“	Pri početku mjerenja	Pokreće postupak mjerenja
„T“	Tijekom mjerenja	Okida/zatražuje mjerenje
„X“	Na kraju mjerenja	Zaustavlja postupak mjerenja
„Q“	Ostatak vremena	Izlaz iz mjernog načina rada (radi kalibracije)

Tablica 2.1 – Protokol komunikacije klijentskog računala s programom

Ove naredbe ne smiju biti završene sa CR i/ili LF terminatorima. Program ne šalje nikakve potvrde o primljenim porukama.

Kad program primi naredbu „T“, računa se udaljenost između markera u zadanim mjernim jedinicama. Izračunata udaljenost se zatim vraća klijentskom računalu u sljedećem formatu:

Format	Simboli
Laaaa.bbbbb	a = {' ' || '-' || '0'-'9'}, b = {'0'-'9'}

Tablica 2.2 – Povratna poruka s mjerenjem

Podatak „L-999.99999“ se šalje kad dogodi pogreška kao što je npr. slučaj kad se izgubi fokus s markera.

7. Literatura

[1] Fryer, J., Mcintosh, K.: *Enhancement of Image Resolution in Digital Photogrammetry*, Photogrammetric Engineering & Remote Sensing, Vol. 67, No. 6, June 2001, pp. 741-749.

[2] Ebrahimi, M., Vrscay, E.,R.: *Multi-frame super-resolution with no explicit motion estimation*, IPCV 2008

[3] Vajapeyazula, P.K., Rajagopalan, A.N.: *Motion-Free Superresolution*, Department of Electrical Engineering Indian Institute of Technology – Madras Chennai – 600 036, India

[4] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligende, 8(6):679-698, 1986.

[5] J., Ye, J., G., Fu, U.P., Poudel: *High-accuracy edge detection with Blurred Edge Model*, Image and Vision Computing, Volume 23, Issue 5, 1 May 2005, Pages 453-467, ISSN 0262-8856

[6] Björck, A.: *Numerical methods for least squares problems,* SIAM, Philadelphia., 1996., ISBN 0-89871-360-9

[7] Press, W. H., Teukolsky, S. A, Vetterling, W. T., Flannery, B. P.: *Numerical Recepies 3rd Edition: The Art od Scientific Computing,* Cambridge University Press, 2007-09-10, ISBN 0521880688

[8] Messphysik Materials Testing GMBH, Altenmarkt 180, 8280 Fürstenfeld, Austrija, http://www.messphysik.com

Razvoj sustava za laboratorijsko mjerenje računalnim vidom

Sažetak

Razvijen je postupak za precizno mjerenje istezanja materijala, pri čemu su ispitni objekti označeni kontrastnim markerima te se na temelju video sekvence s jedne kamere određuje izduljenje objekta. Implementirane su dvije metode precizne lokalizacije ruba koje se temelje na dva modela ruba: model ruba s funkcijom rampe i model ruba temeljen na odzivu Gaussovog filtra nad funkcijom skoka. U izvedbi se koriste algoritmi glađenja, Gauss-Newtonova metoda gradijentne optimizacije i postupak zlatnog reza. Implementacija je izvedena u programskom jeziku C++ uz korištenje OpenCV biblioteka. Metode su detaljno pojašnjene kao i njihova implementacija, a eksperimentalni rezultati testa preciznosti su prikazani i komentirani.

Ključne riječi

Računalni vid, laboratorijsko mjerenje, fotogrametrija, glađenje, Gauss-Newtonov algoritam, postupak zlatnog reza.

Development of computer vision laboratory mesurement system

Summary

A procedure for accurately measuring the elongation of material is developed, in which the test objects are marked with contrast markers and a video sequence from a single camera is used to determine the elongation property. Two methods for precise edge localization are implemented that are based on two edge models: the edge model with the ramp function and the edge model based on the convolution of Gausssian filter with the step function. Developed procedure uses smoothing algorithm, Gauss-Newton method for gradient optimization and a golden section search algorithm. The implementation was done in C++ using the OpenCV library. Developed methods are described in detail as well as their implementation, and experimental results of accuracy tests are presented and discussed.

Keywords

Computer vision, laboratory measuring, photogrammetry, Gaussian blur, Gauss-Newton algorithm, golden section search algorithm.