

Primjena računalnog vida u prediktivnom održavanju

Crnko, Luka

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:346884>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering
and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Luka Crnko

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Dragutin Lisjak, dipl. ing.

Student:

Luka Crnko

Zagreb, 2019.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru prof. dr. sc. Dragutinu Lisjaku na ukazanom povjerenju, pruženoj pomoći i danim savjetima koji su me usmjeravali tijekom izrade rada

Najviše zahvaljujem svojoj obitelji na pruženoj podršci tijekom studiranja.

Luka Crnko



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	
Ur. broj:	

DIPLOMSKI ZADATAK

Student: **LUKA CRNKO** Mat. br.: **0035193288**

Naslov rada na hrvatskom jeziku: **Primjena računalnog vida u prediktivnom održavanju**

Naslov rada na engleskom jeziku: **Application of machine vision in predicting maintenance**

Opis zadatka:

Pod pojmom prediktivnog održavanja podrazumijeva se proces primjene strojnog učenja i IoT senzora za nadzor trenutnog stanja strojeva i njegovih komponenti s ciljem poduzimanja korektivnih radnji za sprječavanje nastanka potencijalnih kvarova. U skladu s navedenim, u radu je potrebno:

1. Opisati strategiju prediktivnog održavanja.
2. Opisati pojam i mogućnosti primjene računalnog vida.
3. Na konkretnom primjeru prediktivnog održavanja prikazati primjenu računalnog vida.
4. Analizom dobivenih rezultata izvesti zaključke.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
02. svibnja 2019.

Rok predaje rada:
04. srpnja 2019.

Predviđeni datum obrane:
10. srpnja 2019.
11. srpnja 2019.
12. srpnja 2019.

Zadatak zadao:
prof. dr. sc. Dragutin Lisjak

Predsjednica Povjerenstva:
prof. dr. sc. Biserka Runje

SADRŽAJ

POPIS SLIKA	III
POPIS TABLICA.....	V
POPIS OZNAKA	VI
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
2. ODRŽAVANJE.....	3
2.1. Povijest održavanja	3
2.2. Metode upravljanja održavanjem.....	4
2.2.1. Poboljšanje.....	5
2.2.2. Korektivno održavanje.....	6
2.2.3. Preventivno održavanje.....	6
2.2.4. Zadano održavanje	8
2.2.5. Održavanje po stanju.....	8
2.2.6. Prediktivno održavanje	10
2.3. Dijagnostika kvara.....	11
2.3.1. Zahtjevi sustava dijagnostike	12
2.3.2. Dijelovi dijagnostičkog sustava	12
2.4. Prediktivno održavanje 4.0.	15
2.4.1. Identificiranje opreme	16
2.4.2. Prikupljanje podataka.....	17
2.4.3. Dizajniranje prediktivnog modela.....	18
2.4.4. Upravljanje zadatcima.....	20
2.4.5. Učenje	20
3. STROJNI I RAČUNALNI VID.....	21
3.1. Općenito o osjetilu vida	21
3.2. Sličnosti i razlike strojnog i računalnog vida.....	23
3.3. Područja primjene računalnog vida.....	23
3.3.1. Obrada fotografija	23
3.3.2. Proces prepoznavanja uzoraka	24
3.3.3. Praćenje objekata	26
3.4. Prednosti upotrebe strojnog vida.....	26
3.5. Vrste strojnog vida	27
3.5.1. Jednodimenzionalni sustavi strojnog vida	27
3.5.2. Dvodimenzionalni sustavi strojnog vida.....	28
3.5.3. Trodimenzionalni sustavi strojnog vida.....	29
4. ALATI ZA OBRADU SLIKE I RAČUNALNI VID.....	30
4.1. Alati za obradu slike	30
4.1.1. Vrste slika u alatu za obradu slike	31
4.1.2. Usklađivanje fotografija.....	33
3.2. Alat za računalni vid	34

3.3. Korištene funkcije MATLAB-ovih alata	34
3.3.1. Osnove učitavanja i prikaza fotografija	34
3.3.2. Funkcije za usklađivanje fotografija	36
3.3.3. Funkcije za pronalaženje razlika između fotografija	38
3.3.4. Ostale funkcije	39
5. PRIMJENA RAČUNALNOG VIDA NA PRIMJERU ODRŽAVANJA ZRAKOPLOVA	41
5.1. Prikaz dijagrama toka programa za otkrivanje promjena	41
5.2. Učitavanje fotografija	42
5.3. Prikaz rezultata.....	43
5.4. Pretvaranje fotografije u boji u sive tonove	46
5.5. Usklađivanje odmak na fotografijama	53
5.5.1. Stvaranje optimizatora i metrike	53
5.5.2. Podešavanje svojstva optimizatora	53
5.5.3. Korištenje dobivenih vrijednosti optimizatora i metrike radi usklađivanja fotografija.....	54
5.6. Pronalazak promjena između fotografija	55
5.7. Pronalaženje najveće vrijednosti promjene između fotografija	56
5.8. Određivanje praga osjetljivosti	57
5.9. Čišćenje šumova slike	60
5.10. Stvaranje maske s pronađenim promjenama.....	61
5.11. Pozivanje funkcije za prebrojavanje pronađenih razlika	62
5.11.1. Opis funkcije filterRegions	62
5.11.2. Opis funkcije sortProperties.....	64
5.12. Konačni prikaz rezultata	64
5.12.1. Točnost prepoznavanja anomalija površine.....	67
ZAKLJUČAK	70
Literatura	72
PRILOZI.....	74

POPIS SLIKA

Slika 1.	Vrste održavanja [1]	5
Slika 2.	Krivulja kade	7
Slika 3.	Princip održavanja po stanju [4].....	9
Slika 4.	Blok shema računalom upravljanog dijagnostičkog sustava [6]	11
Slika 5.	Faze dijagnostičkog sustava [7]	13
Slika 6.	Razine prediktivnog održavanja [9]	15
Slika 7.	Koraci uvođenja prediktivnog održavanja temeljenog na internetu stvari [10]	16
Slika 8.	Karakteristična krivulja pouzdanosti [10]	19
Slika 9.	Primjer prebrojavanja uzoraka [13].....	22
Slika 10.	Primjer stvaranja 3D modela prostora na temelju velikog broja 2D fotografija [13]	22
Slika 11.	Prepoznavanje uzoraka [12]	25
Slika 12.	Prikaz linijskog konvejera s jednodimenzionalnim sustavom strojnog vida [11].	28
Slika 13.	Prikaz linijskog konvejera s dvodimenzionalnim sustavom strojnog vida [11]....	28
Slika 14.	Trodimenzionalno snimanje površine [11].....	29
Slika 15.	Vrijednosti piksela na binarnoj slici [18]	31
Slika 16.	Vrijednost piksela na slici sivih tonova [18]	32
Slika 17.	Vrijednosti piksela slike u boji [18]	33
Slika 18.	Dijagram toka programa za otkrivanje promjena na trupu zrakoplova.....	41
Slika 19.	Početna fotografija	44
Slika 20.	Nova fotografija	45
Slika 21.	Uvećano područje razlika.....	46
Slika 22.	Usporedni prikaz početne i nove fotografije bez translacije	47
Slika 23.	Usporedni prikaz početne i nove fotografije s translacijom prema gore.....	48
Slika 24.	Usporedni prikaz početne i nove fotografije s translacijom prema dolje.....	48
Slika 25.	Usporedni prikaz početne i nove fotografije s translacijom prema lijevo.....	49
Slika 26.	Usporedni prikaz početne i nove fotografije s translacijom prema desno	50
Slika 27.	Usporedni prikaz početne i nove fotografije s translacijom prema gore-lijevo	50
Slika 28.	Usporedni prikaz početne i nove fotografije s translacijom prema gore-desno	51
Slika 29.	Usporedni prikaz početne i nove fotografije s translacijom prema dolje-lijevo ...	52
Slika 30.	Usporedni prikaz početne i nove fotografije s translacijom prema dolje-desno ...	52

Slika 31.	Rezultat nakon provedenog usklađivanja.....	54
Slika 32.	Promjene nastale kao rezultat usporedbe fotografija.....	55
Slika 33.	Točke s najvećom vrijednošću promjene	57
Slika 34.	Prag osjetljivosti za područja veća od 2	58
Slika 35.	Prag osjetljivosti za područja veća od 20	59
Slika 36.	Prikaz fotografije nakon određivanja praga osjetljivosti.....	60
Slika 37.	Prikaz slike bez utjecaja šuma	61
Slika 38.	Prikaz područja različita (zelena boja) od početne fotografije	62
Slika 39.	Prikaz nove fotografije koja nema vidljivih promjena u odnosu na postojeću	65
Slika 40.	Prikaz nove fotografije s označenim promjenama i brojem promjena u odnosu na početnu fotografiju	66
Slika 41.	Uvećan prikaz pronađenih promjena na novoj slici	67
Slika 42.	Usporedni prikaz nove fotografije s prikazanim razlikama i pronađenih razlika uz pomoć algoritma.....	69

POPIS TABLICA

Tablica 1. Prednosti i nedostaci preventivnog održavanja.....	10
Tablica 2. Upotreba strojnog vida prilikom ostvarivanja strateških ciljeva.....	27
Tablica 3. Vrste metoda za prikaz slika	35
Tablica 4. Vrste geometrijskih transformacija slika.....	38
Tablica 5. Prikaz korištenih fotografija.....	43

POPIS OZNAKA

Oznaka	Opis
MTTF	<i>Mean Time to Failure</i>
RCM	<i>Reliability Centered Maintenance</i>
TPM	<i>Total Productive Maintenance</i>

SAŽETAK

Cilj ovog diplomskog rada jest prikazati primjenu računalnog vida u procesu prediktivnog održavanja. Prediktivno održavanje je metoda održavanja koja ima za cilj predvidjeti kvarove prije same pojave kvara. Kako bi se održavanju moglo pristupiti u najpogodnijem trenutku, u kojem se ne ometa normalno funkcioniranje rada stroja, koristi se prediktivna metoda. Računalni vid je jedan od najsloženijih procesa kojim se na temelju ulaznih parametara (slike ili videa) pokušava dobiti što je moguće više informacija o stanju sustava. U radu su navedene metode održavanja i njihov razvoj kroz povijest, način određivanja kvara pomoću dijagnostičkog sustava te uloga, karakteristike i područja primjene računalnog vida. Opisan je alat za obradu fotografija programskog paketa MATLAB te funkcije MATLAB-ovih alata koje su korištene za izradu aplikacije koja omogućava pronalazak promjena na fotografijama nastalih zbog oštećenja trupa zrakoplova. Nakon dobivenih rezultata osoba zadužena za održavanje, temeljem pregleda fotografija s pronađenim razlikama, može donijeti početnu odluku treba li se pristupiti procesu održavanja ili je registrirana pogreška nastala kao rezultat šuma. Ovim alatom ne osigurava se stopostotna sigurnost u otkrivanju pogrešaka, već održavatelji mogu dobiti brzi uvid u područja na trupu zrakoplova na kojima bi mogla nastati promjena u odnosu na prethodni pregled.

Ključne riječi: prediktivno održavanje, dijagnostika kvara, računalni vid, MATLAB.

SUMMARY

The aim of this study is to demonstrate the use of machine vision in the process of predictive maintenance. Predictive maintenance is a maintenance method that aims to predict failures before the failure occurs. In order to keep the maintenance at the most convenient moment, in which the normal operation of the machine does not interfere, the predictive method is used. Machine vision is one of the most complex processes that attempts to get as much information as possible about the state of the system based on the input parameters (images or videos).

The paper describes maintenance methods and their development through history, how to determine failure using the diagnostic system and the role, characteristics and areas of application of machine vision. The MATLAB Image Processing Toolbox is described, as well as the MATLAB tools used to create an application that allows discovery of changes in photos caused by damage to the aircraft hull. After the obtained results, the person in charge of maintenance, based on the photo review with found differences, can make the initial decision to access the maintenance process or a registered fault appeared as a result of noise. This tool does not provide absolute security in fault detection, but maintainers can get quick insight into areas on the aircraft hull where a change may occur in regard to previous review.

Key words: predictive maintenance, diagnostic fault, machine vision, MATLAB.

1. UVOD

Od samih početaka ljudi izrađuju alate koji bi im olakšali svakodnevni život. Napretkom tehnologije danas ljudi imaju različite strojeve koji mogu samostalno obavljati poslove puno brže i točnije nego prije. Jedan od glavnih nedostataka današnjih strojeva je ovisnost o čovjeku koji im govori što i kako moraju raditi. Danas, u velikoj mjeri, održavanje ovisi o radu čovjeka i njegovom iskustvu. Posljednjih godina uz pomoć senzora, kamera i slično, sakupljaju se velike količine podataka koji imaju svrhu praćenja i kontrole procesa. Postoji težnja da se cijeli proces što je moguće više automatizira i strojevi što bolje iskoriste. Održavanje je dugi niz godina bilo zanemareno i vlasnici poduzeća nisu shvaćali prednosti kvalitetnog održavanja. Stoga su koristili opremu do pojave kvara. Danas se na temelju metode održavanja opreme može puno saznati o uspješnosti poduzeća. Korištenjem neadekvatnih metoda održavanja, osim značajnog utjecaja na troškove poduzeća, može se utjecati i na zadovoljstvo krajnjih korisnika usluga kao i na zadovoljstvo radnika. U radu će biti spomenute metode održavanja i pobliže objašnjen nastanak prediktivnog održavanja, kao jedne od trenutno najmodernijih metoda. Korištenjem prediktivnog održavanja, pokušava se, pomoću različitih hardverskih i softverskih rješenja, pronaći zakonitosti koje dovode do nastanka kvara. To bi u budućnosti moglo pomoći otklanjanju kvara samo onda kada je to zbilja potrebno.

Predmet i cilj rada je pronaći i predložiti način primjene računalnog vida vezan uz problem održavanja zrakoplova Croatia Airlinesa. U radu su korištene fotografije prikupljene bespilotnom letjelicom koje su pomoću programskog paketa MATLAB obrađene radi dobivanja podataka o promjenama na trupu zrakoplova.

Za izradu rada korišteni su sekundarni izvori podataka, to jest podaci preuzeti iz ranijih istraživanja. Izvori podataka preuzeti su iz sveučilišnih udžbenika i članaka u znanstvenim i stručnim časopisima.

Rad je podijeljen u šest poglavlja u koje su uključena uvodno poglavlje i zaključak. U uvodnom dijelu iznosi se predmet i cilj rada, struktura i sadržaj rada te izvori prikupljenih podataka. U drugom poglavlju definiran je pojam održavanja, povijest, način razvoja metoda održavanja kao i metode dijagnosticiranja kvara. Treće poglavlje definira razliku između strojnog i računalnog vida, prikazane su vrste strojnog vida, nabrojane su prednosti i područje njegove

primjene. Četvrto poglavlje sadrži opis programskog paketa MATLAB i funkcionalnosti koje omogućuju obradu fotografija. U petom poglavlju dan je primjer načina održavanja primjenom računalnog vida. Posljednje, šesto poglavlje sadrži temeljne zaključke izvedene kroz pisanje rada i predložena buduća poboljšanja.

2. ODRŽAVANJE

Prema normi EN 13306:2017 Održavanje je proces koji koristi kombinaciju svih tehničkih, administrativnih i upravljačkih aktivnosti tijekom životnog ciklusa proizvoda. Cilj održavanja je zadržati ili vratiti proizvod u stanje u kojem može obavljati traženu funkciju. [1]

2.1. Povijest održavanja

Kroz povijest održavanju se nije pridavala velika važnost. Ljudi su smatrali da održavanje nije potrebno te su pristupali popravljaju tek kada bi naprava prestala funkcionirati. Naprave koje su se koristile bile su jednostavne građe. Pojavom složenijih i tehnološki naprednijih naprava uvidjela se potreba za održavanjem. Strojevi koji su se pojavili početkom industrijske revolucije i prije prvog svjetskog rata bili su vrlo robusni i teški za upravljanje. Ti strojevi bili su pouzdani jer su imali malo pokretnih dijelova i zahtijevali su malo zahvata vezanih uz održavanje. Pedesetih godina 20. stoljeća, nakon II. svjetskog rata, bilo je potrebno obnoviti uništenu industriju pa se u Japanu i Njemačkoj razvija konkurentnije tržište koje nije toleriralo zastoje. Troškovi radne snage postali su sve značajniji dio proizvoda te dolazi do automatizacije proizvodnje. Strojevi su postali lakši, boljih performansi, a tržište je zahtijevalo veću količinu proizvoda što dovodi do pojave kvarova na strojevima. Kako su strojevi bili važni za proizvodnju, a svaki je zastoj povećavao troškove, dolazi do razvoja planiranog i preventivnog održavanja kojima se pokušao smanjiti utjecaj zastoja na cjelokupni proces. Tijekom 1970-ih proizvodni pogoni postaju još složeniji i automatiziraniji. Kao glavne osobine održavanja smatrale su se pouzdanost, dostupnost, održivost, a kvaliteta, sigurnost, zaštita okoliša i prilagodljivost smatraju se glavnim prednostima. Prilikom upravljanja i održavanja proizvodnih pogona počeli su se koristiti informacijski sustavi pomoću kojih je bilo lakše pratiti određene strojeve. Tada se javlja održavanje po stanju kojem je cilj pratiti promjene na dijelovima i zamijeniti oštećene dijelove prije nego dođe do kvara strojnog dijela i zaustavljanja linije. Početkom 1980-ih godina dolazi do pojave novih koncepata u održavanju kao što su Cjelovito produktivno održavanje (engl. *Total Productive Maintenance* – dalje u tekstu: TPM) i Održavanje temeljeno na pouzdanosti (engl. *Reliability Centered Maintenance* – dalje u tekstu: RCM). Glavna prednost TPM-a je što obuhvaća širok raspon procesa, a ne samo održavanje, dok je prednost RCM-a što strogo određuje kada i kako se provodi održavanje. Tada, još većom upotrebom tehnologije, dolazi do promjena u načinu razmišljanja vodećeg kadra koji je smatrao da se ništa ne može učiniti kako bi se utjecalo na troškove održavanja. Razvojem računala i mikroprocesora omogućeno je lakše praćenje i upravljanje održavanjem čime je smanjen broj

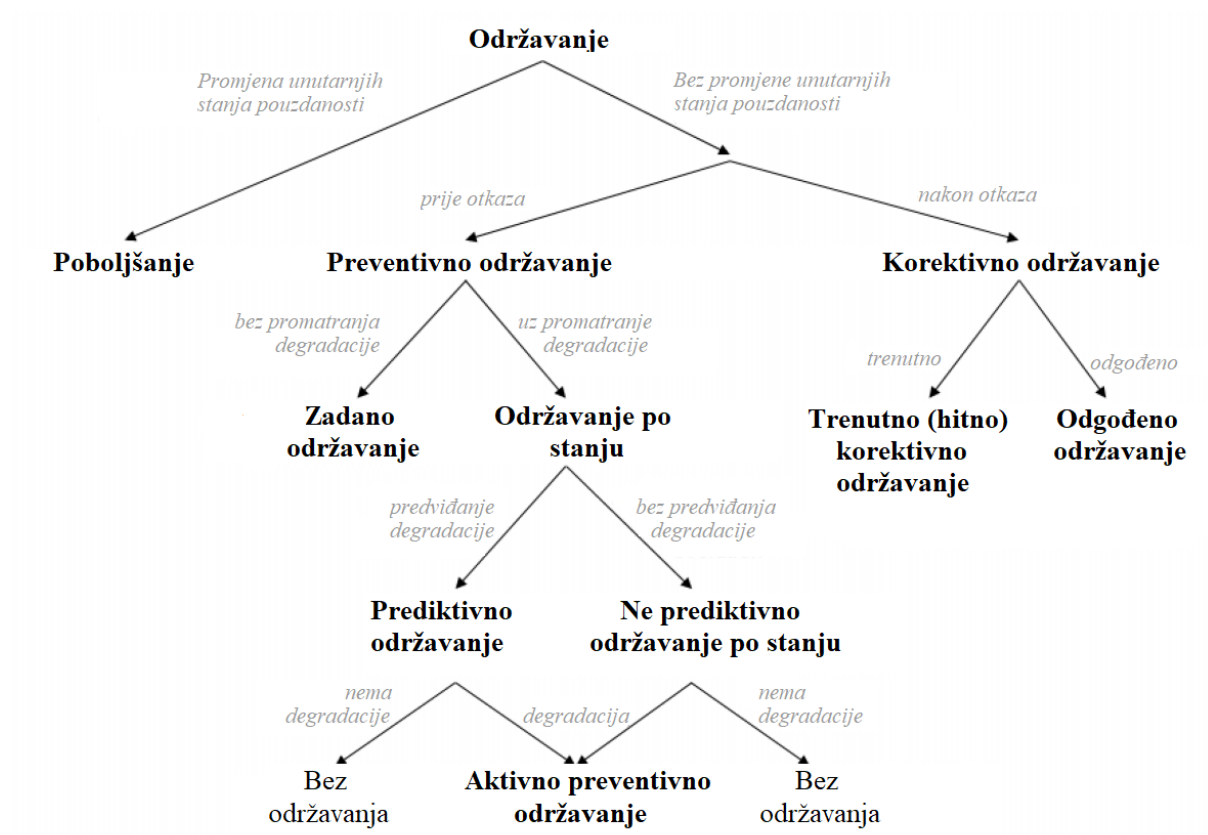
nepotrebnih popravaka te povećan broj popravaka onih kvarova koji dovode do kvara. Danas je održavanje prepoznato kao generator dobiti. U nekim djelatnostima osoblje koje se bavi održavanjem predstavlja većinu zaposlenih. Održavanje postaje bitan dio lanca opskrbe jer nakon godina zanemarivanja, poduzeća prihvaćaju održavanje kao područje u kojem je moguće ostvariti velike uštede. [2]

2.2. Metode upravljanja održavanjem

Razvojem različitih metoda odražavanja pokušavalo se utjecati na produženje životnog vijeka stroja i smanjenje troškova vezanih uz održavanje. Podjelu održavanja najbolje je opisati prema normi EN 13306:2017 koju prikazuje slika 1.

Za razumijevanje slike 1. potrebno je definirati osnovne pojmove: [1]

- **Pouzdanost** (engl. *Dependability*) predstavlja sposobnost obavljanja zadatka kako i kada je potrebno. Pouzdanost obuhvaća pojmove kao što su: dostupnost, sigurnost, trajnost, ekonomičnost i ostale faktore koji utječu na njih - održivost, podrška održavanju, način uporabe.
- **Kvar** (engl. *Failure*) označava nemogućnost obavljanja traženog zadatka. Poslije kvara cjelina ima grešku (neispravnost), koja može biti djelomična ili kompletna. Kvar predstavlja događaj i on je različit od greške koja predstavlja stanje.
- **Greška** (engl. *Fault*) opisuje stanje koje karakterizira nemogućnost obavljanja potrebne funkcije, ali ne uključuje nemogućnost tijekom preventivnog održavanja, drugih planiranih radnji ili nedostatka vanjskih resursa. Greška uobičajeno nastaje zbog kvara, ali u nekim slučajevima greška može postojati od prije. Primjer takve greške su pogreške u projektiranju, proizvodnji ili održavanju.
- **Degradacija** (engl. *Degradation*) naziva se još i zamorom materijala, a označava nepovratan proces pogoršanja jednog ili više svojstava, a može biti prouzrokovana vremenom, upotrebom ili utjecajem vanjskih faktora. Degradacija može dovesti do kvara.



Slika 1. Vrste održavanja [1]

2.2.1. Poboljšanje

Pojam poboljšanja (engl. *Improvement*) u podjeli vrsta održavanja relativno je nov. Pod pojmom poboljšanja smatra se poboljšanje unutarnjih stanja pouzdanosti. To je kombinacija svih tehničkih, administrativnih i promjena menadžmenta koje bez promjene izvornih funkcija rezultiraju poboljšanjem stvarne pouzdanosti i/ili mogućnosti održavanja i/ili poboljšanju sigurnosti. Poboljšanja se provode kako bi se spriječila pogrešna upotreba i izbjegle greške. [1]

2.2.2. *Korektivno održavanje*

Korektivno održavanje (engl. *Corrective maintenance*) je način održavanja u kojem uopće nema održavanja i funkcionira na principu „popravi ga kada se slomi“. Ovaj tip održavanja je neplaniran i hitan. Dugi niz godina ovakva metoda bila je jedini način održavanja, strojeve se iskorištavalo do samih granica i održavanju se pristupalo samo u slučaju ako je došlo do kvara. Na taj način nije se htjelo ulagati dodatna sredstva, za popravak, u proizvod koji „radi“. Takva metoda ujedno je i najskuplja jer dolazi do dužih zastoja u proizvodnji, velikog skladišta rezervnih dijelova, nepotrebnog trošenja ispravnih dijelova te se ne mogu predvidjeti sredstva potrebna za popravak, a mogu se javiti i problemi sa sigurnosti prilikom rada na takvoj opremi. Prednost ove metode održavanja je što nema potrebe za ljudima koji bi se brinuli planiranju i provođenju zahtjeva za održavanjem. Prema analizama, korektivno održavanje je u prosjeku tri puta skuplje od preventivnog održavanja [2].

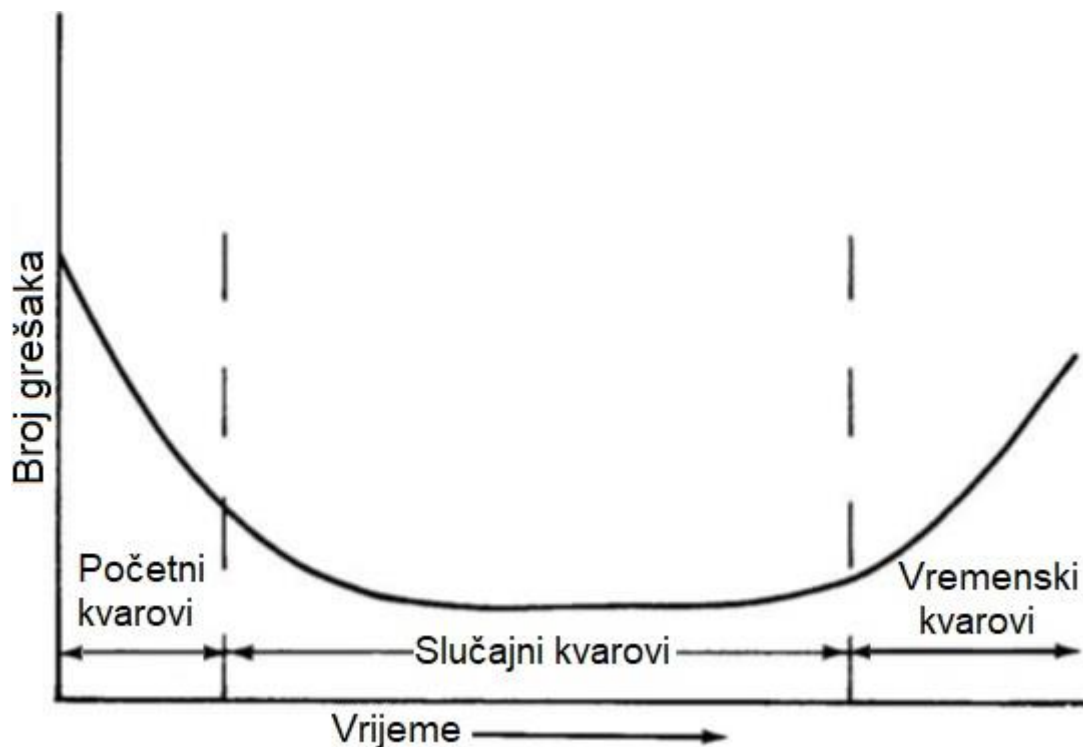
Korektivno održavanje dijeli se na: [1]

1. **trenutno (hitno) korektivno održavanje** (engl. *Immediate Corrective Maintenance*) - kod ove vrste održavanja otklanjanju kvara pristupa se odmah nakon otkrivanja greške kako bi se izbjegle neprihvatljive posljedice.
2. **odgođeno korektivno održavanje** (engl. *Deferred Corrective Maintenance*) - kod ove vrste održavanja popravku se ne pristupa odmah, nakon otkrivanja greške, već je odgođeno za neki vremenski period.

2.2.3. *Preventivno održavanje*

Preventivno održavanje (engl. *Preventive maintenance*) predstavlja prvu stepenicu prilikom uvođenja održavanja. Prilikom preventivnog održavanja potrebno je planirati procese koji prethode održavanju i donijeti odluku kada je najbolji trenutak za provođenje održavanja jer je potrebno zaustaviti rad strojeva. Zadaci preventivnog održavanja temelje se na proteklom vremenu ili radnim satima koji se temelje na statističkim ili povijesnim podacima za određenu vrstu opreme. Preventivno održavanje bazirano je na krivulji kade koju prikazuje slika 2. Prema dijagramu krivulje kade najveća vjerojatnost pojave kvara nastaje u prvih nekoliko dana ili tjedana nakon instaliranja opreme, a uzrok tome su problemi vezani uz proizvodnju ili instalaciju opreme. Nakon početnog razdoblja, vjerojatnost neuspjeha je relativno niska tijekom duljeg vremenskog razdoblja. Nakon proteka normalnog vremena eksploatacije opreme vjerojatnost kvara naglo se povećava s protekom vremena ili radnih sati. Kod preventivnog

održavanja, pregled strojeva, podmazivanje, popravci ili nadogradnje planirani su prema statističkim podacima koji se temelje na krivulji srednjeg vremena do pojave kvara (engl. *Mean Time to Failure* – MTTF) . [3]



Slika 2. Krivulja kade

Za vrijeme normalne eksploatacije opreme, opremi je potrebno minimalno održavanje što uključuje aktivnosti kao što su: podmazivanje, zamjenu potrošnih dijelova i slične sitne popravke.

Glavni nedostatak preventivnog održavanja je što se ista vrsta opreme ne koristi pod istim uvjetima te zbog toga korištenje statističkih podataka ne daje u potpunosti točne podatke kada je potrebno pristupiti procesu održavanja. Uz sve navedene nedostatke, preventivnim održavanjem pospješuje se otkrivanje i otklanjanje kvara prije njegovog nastanka. U odnosu na korektivno, održavanje smanjuje troškove i produžuje vijek trajanja opreme.

2.2.4. Zadano održavanje

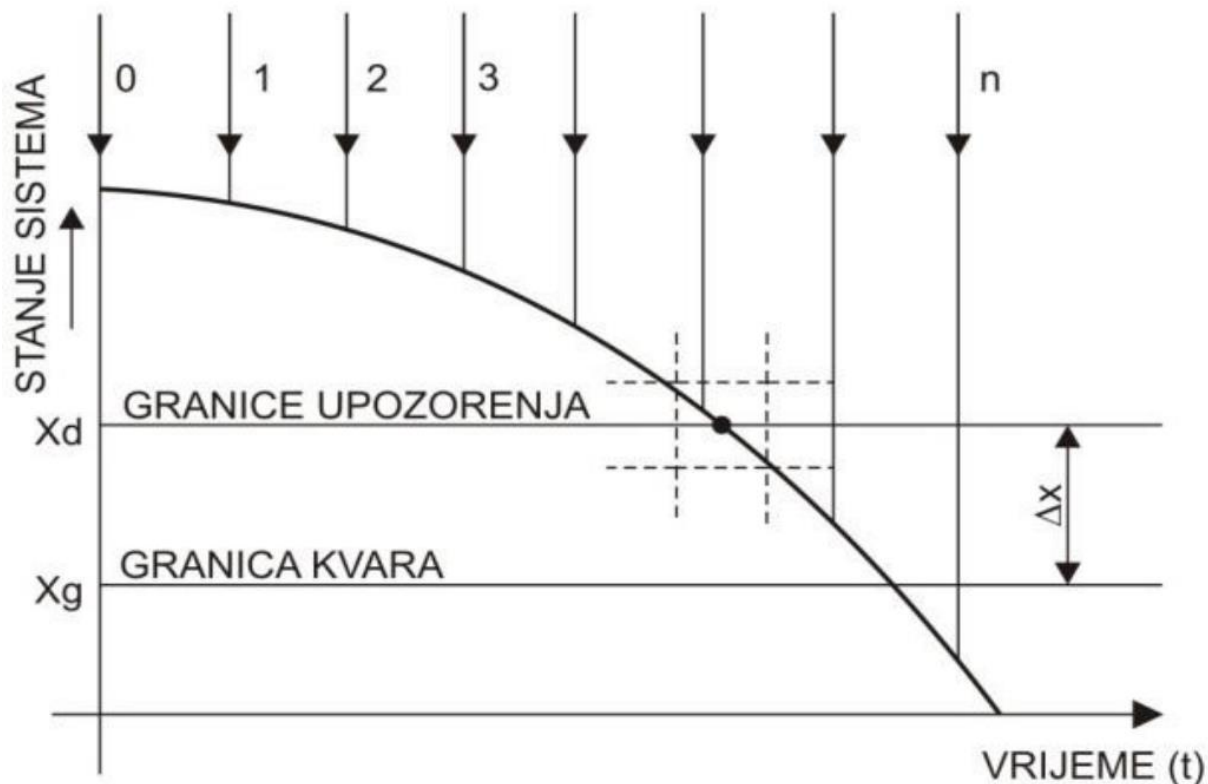
Preventivno održavanje bez promatranja degradacija naziva se zadano održavanje (engl. *Predetermined maintenance*). Ono se temelji na unaprijed određenim intervalima i provodi se periodički bez prethodnog pregleda. [1]

Zadano održavanje može se podijeliti na:

1. radno orijentirano – ako se održavanje izvodi nakon proteka određenog vremena rada ili određenim brojem jedinica upotrebe,
2. vremenski orijentirano – ako se održavanje izvodi nakon proteka određenog vremenskog perioda neovisno koristi li se oprema ili ne.

2.2.5. Održavanje po stanju

Održavanje po stanju (engl. *Condition based maintenance*) oblik je preventivnih aktivnosti koje se izvode prije nastanka kvara, a nastaje kao rezultat poznavanja stanja opreme. Kod održavanja po stanju kontinuirano se prate razni definirani parametri, a održavanju se pristupa samo ukoliko je određena mjera izvan određenih granica. Održavanje po stanju zasniva se na „praćenju stanja“ sustavnih elemenata sustava te na uočavanju ili prognoziranju vremenskog trenutka dostizanja graničnih vrijednosti koje dovode do greške, to jest kvara. Prema rezultatima pregleda poduzimaju se postupci održavanja s odgodom. Princip održavanja po stanju prikazan je na slici 3. Prilikom prve „provjere stanja“ bitno je određivanje „početnog stanja“ koje predstavlja temelj u odnosu na koju se rade sve ostale „provjere stanja“. Definiraju se granica upozorenja (X_d) i granica kvara (X_g) koje su utvrđene pokusima i prezentirane u normativno-tehničkoj dokumentaciji sustava. Granica upozorenja predstavlja dopuštenu vrijednost parametra stanja, a prelaskom te granice pristupa se održavanju po stanju. Vremenski period između granice upozorenja i granice kvara mora biti dovoljno dug da bi se uspjelo pristupiti održavanju i spriječila pojava stanja „u kvaru“. Povećanjem vremenskog perioda dolazi do povećanja broja i troškova održavanja čime model održavanja po stanju ne bi imao smisla. Sprječavanje parametara stanja da izađu iz dopuštenih granica naziva se „signalizacijska tolerancija“ (Δx) koja određuje stupanj osjetljivosti odabrane dijagnostičke metode. [4]



Slika 3. Princip održavanja po stanju [4]

Održavanje po stanju najčešće se dijeli na: [4]

1. **održavanje prema stanju s kontrolom parametara** - predviđa stalno ili periodičnu kontrolu i mjerenje parametara kojima se određuje funkcionalno stanje sustava ili dijelova sustava. Održavanju se pristupa kada vrijednost kontroliranih parametara dosegne „granicu upotrebljivosti“, odnosno pred kritičnu razinu.
2. **održavanje prema stanju s kontrolom razine pouzdanosti** – sastoji se od prikupljanja, obrade i analize podataka o razini pouzdanosti komponenta ili sustava. Kod ove vrste održavanja ispitivanjem se određuje dopuštena razina pouzdanosti, koja se najpotpunije izražava intenzitetom kvara. Sustav se koristi bez ograničenja resursa za održavanje sve dok je stvarna razina pouzdanosti veća od dopuštene razine pouzdanosti.

2.2.6. Prediktivno održavanje

Glavna premisa prediktivnog održavanja je praćenje trenutnog mehaničkog stanja oprema, učinkovitosti kao i svih ostalih pokazatelja koji utječu na performanse. Cilj prediktivnog održavanja je maksimalno povećanje intervala između popravaka, minimiziranje broja popravka, kao i smanjenje troškova neplaniranih zastoja i kvara. Prediktivno održavanje sastoji se od niza raznih aktivnosti (kao što je nadzor vibracija, termografska mjerenja, tribologija i slično) pomoću kojih se prate promjene svojstava i promjene u radu. Temeljem toga poduzimaju daljnje aktivnosti. Upotrebom prediktivnog održavanja, u kombinaciji s ostalim metodama održavanja, optimizira se dostupnost strojeva i u velikoj mjeri smanjuju troškovi održavanja. Prediktivno održavanje utječe na konačnu, bolju, kvalitetu proizvoda, produktivnost i profitabilnost proizvodnog pogona.

Prediktivno održavanje u uskoj je vezi s metodom preventivnog održavanja. Umjesto da se prilikom održavanja oslanja na statističke podatke o prosječnom životnom vijeku proizvoda, prediktivno održavanje koristi izravno praćenje mehaničkog stanja i učinkovitosti kako bi se u svakom trenutku mogla odrediti stvarna vrijednost stanja sustava. Kod preventivnog održavanja, odabir vremena popravka, održavatelj donosi na temelju intuicije i iskustva. Kod prediktivnog održavanja postoje stvarni pokazatelji koji upućuju na promjenu. Na taj način, prilikom svakog preventivnog pregleda, se utvrđuju nedostaci na temelju kojih se može pristupiti pripremi i otklanjanju kvara već u ranoj fazi nastanka. Tako se prevenira nastanak ozbiljnijih problema uz manje konačne troškove. [5]

Prednosti i nedostaci preventivnog održavanja prikazani su tablicom 1. [6]

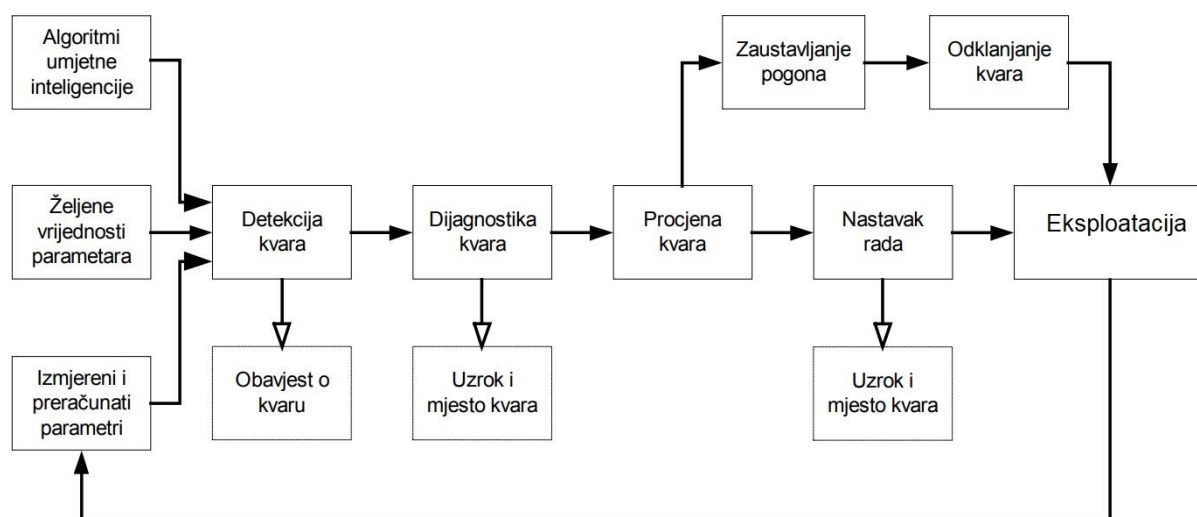
Tablica 1. Prednosti i nedostaci preventivnog održavanja

Prednosti	Nedostaci
Povećana dostupnost i radni vijek komponenti	Visoki inicijalni troškovi instaliranja opreme
Smanjenje vrijeme nedostupnosti opreme	Povećanje troškova zbog obuke zaposlenika
Smanjenje troškova opreme i rada	Uštede nisu jasno vidljive menadžmentu
Veća kvaliteta proizvoda	
Povećanje sigurnosti na radu i zaštite okoliša	
Ušteda energije	
Procijenjena ušteda u odnosu na preventivno održavanje iznosi između 8 i 12 posto	

2.3. Dijagnostika kvara

Dijagnostika je pravovremeno ili periodičko određivanje stanja nekog stroja s ciljem procjene pouzdanosti daljnjeg pogona i predlaganja načina i obima servisiranja. Dijagnostika ima ulogu otkriti kvar u najranijoj mogućoj fazi nadziranjem stanja stroja. Na taj način olakšava se planiranje održavanja i popravaka, čime se smanjuje vrijeme u kojem je stroj izvan funkcije te na se tako smanjuju troškovi prouzrokovani zastojem. Osnova dijagnostike je usporedba stvarnih i željenih ponašanja parametara stroja. Osim teorijskih znanja vrlo je važno i iskustveno poznavanje ponašanja stroja u pojedinim kvarnim režimima. [7]

Proces dijagnostike prikazan je blok shemom na slici 4.



Slika 4. Blok shema računalom upravljanog dijagnostičkog sustava [6]

U automatiziranom dijagnostičkom sustavu za kvalitetnu dijagnostiku najznačajniji su blokovi koji sadrže „željene vrijednosti parametara“ i „algoritme umjetne inteligencije“. Željene vrijednosti parametara dobivene su iz matematičkih modela, teoretskih znanja i provedenih simulacija. One predstavljaju tražene vrijednosti koje sustav mora zadovoljiti prilikom ispravnog rada. Algoritmima umjetne inteligencije određuju se metode kojima će se pomoću ekspertnog sustava te unaprijed izmjerenih i proračunatih veličina, detektirati pojava nekog kvara.

Veličine značajne za rad sustava neprestano se mjere. Izmjerene veličine se preračunavaju i uspoređuju sa željenim vrijednostima parametara. Prilikom normalnog rada, izmjerene veličine

se trebaju nalaziti unutar unaprijed određenih granica. Kada nastupi odstupanje i izmjerene veličine izađu izvan dozvoljenih granica, sustav upozorava o pojavi mogućeg kvara. Pomoću ekspertnog sustava, na osnovi ponašanja sustava, donosi se zaključak o mogućem mjestu i uzroku kvara (dijagnostika kvara). Procjenjuje se dijagnosticirani kvar te se donosi odluka o nastavku ili prekidu rada kako bi se pristupilo procesu otklanjanja kvara (održavanju). U slučaju kada je odstupanje od željenih veličina malo i ne postoji opasnost od većih oštećenja, sustav nastavlja s radom, a dobiveni podaci se koriste za planiranje održavanja i ukoliko je potrebno nabavi rezervnih dijelova. Kod većih odstupanja procesu održavanja pristupa se u najkraćem mogućem roku. [7]

Veliki dio procesa prikupljanja podataka, kao i inicijalna provjera stanja obavlja se ručno, to jest obavlja ju čovjek. Prilikom takvog načina prikupljanja mogu nastati odstupanja u vidu nepažljive promjene stanja ili neispravne mjerne opreme.

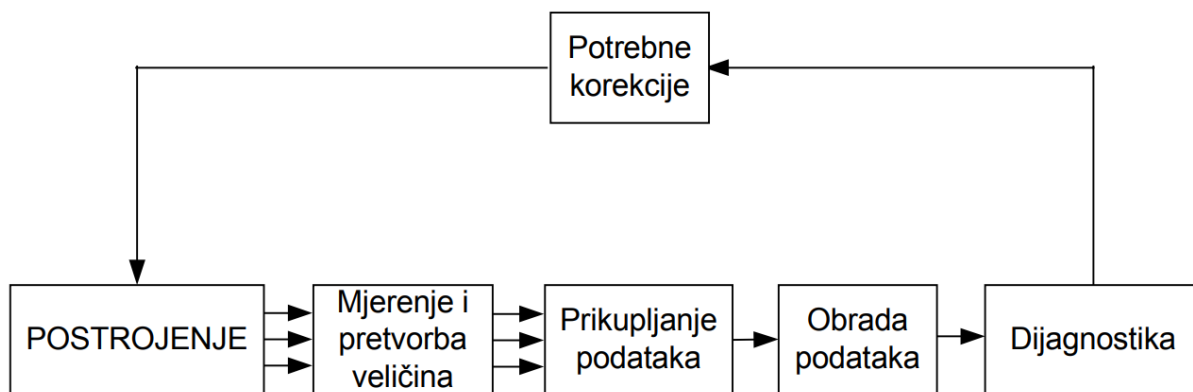
2.3.1. Zahtjevi sustava dijagnostike

Za uspješno obavljanje zadaće dijagnostike, dijagnostički sustav bi trebao zadovoljiti sljedeće uvjete: [8]

- **pouzdanost** – da bi se pouzdano dijagnosticirao kvar signal koji se analizira mora sadržavati sve potrebne informacije
- **jednostavnost** – senzori pomoću kojih se dijagnosticiraju promjene ne smiju biti ometati normalni rad pogona ili biti smješteni na nedostupnim mjestima
- **cijena** – sustava za dijagnostiku može se smanjiti uporabom već postojećih senzora i opreme koja služi za upravljanje pogonom
- **automatizacija** – kako bi se izbjegla ljudska pogreška potrebno je proces što je moguće više automatizirati.

2.3.2. Dijelovi dijagnostičkog sustava

Dijagnostika stanja nekog procesa predstavlja složen zadatak, ali čitav proces može se ilustrirati prema slici 5.



Slika 5. Faze dijagnostičkog sustava [7]

Karakteristične faze takvog procesa su: [7]

- mjerenje i pretvorba karakterističnih veličina
- prikupljanje podataka
- obrada podataka
- dijagnostika

2.3.2.1. Mjerenje i pretvorba veličina

Prilikom odabira mjerenih veličina potrebno je odabrati one veličine pomoću kojih se mogu dobiti najpouzdanije informacije. Iz tog razloga u stroj se ugrađuju različite vrste senzora koji konstantno mjere promjene i prikupljaju podatke. Neke od najvažnijih mjerenih veličina prilikom dijagnostike kvara su:

- **mjerenje temperature** – stalnom i relativno velikom promjenom temperature dolazi do deformacija u materijalu koje su golim oko nevidljive. Senzorom temperature dobivaju se podaci koji se pomoću matematičkog modela obrađuju i na temelju prethodnih promjena daju rezultate o stanju materijala.
- **mjerenje vibracija** – u slučaju mnogih nepravilnosti na stroju vibracije mogu biti najpouzdaniji pokazatelji promjene. Mjerenje vibracija najčešće se koristi kod rotacionih dijelova kod kojih se pojava kvara manifestira zajedno sa pojavom vibracija.
- **termografija** – pregled stroja vrši se pomoću specijalnih kamera koje omogućuju prikaz promjene u temperaturi površine. Prilikom upotrebe termografije valja imati na umu da je temperatura površine često niža od temperature jezgre.

2.3.2.2. Prikupljanje podataka

Prikupljanje podataka predstavlja zapisivanje izmjerenih podataka. Prikupljanjem podataka kroz duži vremenski period omogućeno je kvalitetno praćenje stanja stroja. Automatizirani sustav prikupljanja podataka u pravilu je bolji nego kada to radi čovjek. Automatiziranim načinom prikupljanja podatka moguće je prikupljati podatke bez prekida i uvijek s istom točnošću. Kako bi se spriječila pojava kvara na senzorima za prikupljanje podataka moguće je postaviti više senzora koji prikupljaju podatke za istu stvar.

Kako se podaci koriste u daljnjoj obradi zahtjevi koje podaci moraju ispuniti su:

- **cjelovitost** – ne smiju imati vrijednosti koje nedostaju,
- **točnost** – podaci moraju biti prikupljeni odgovarajućom opremom koja ne daje pogrešne rezultate,
- **kvaliteta** – što je moguće manje šumova i smetnji.

2.3.2.3. Obrada podataka

Temeljem prikupljenih podataka izvodi se obrada podataka. U slučaju automatizacije procesa, obrada podataka može se izvršavati za vrijeme prikupljanja podataka (on-line) ili nakon što su podaci prikupljeni i spremljeni (off-line). [7]

Najjednostavnija metoda obrade podataka sastoji se od usporedbe mjerene veličine kroz duži vremenski period. Temeljem promjena koje se događaju u vremenu može se locirati koja je to vrijednost pri kojoj dolazi do kvara.

Neke od ostalih metoda obrade podataka su: [7]

- **spektralna analiza,**
- **funkcija korelacije,**
- **usrednjavanje signala.**

2.3.2.4. Dijagnostika

Dijagnostika je zadnja i ključna faza dijagnostičkog procesa. Ona na temelju prikupljenih i obrađenih podataka otkriva vrste i veličine kvara, odnosno mogući uzrok pojave kvara. To je faza dijagnostičkog procesa koja je još uvijek najmanje automatizirana i u mnogim slučajevima

do izražaja dolazi iskustvo održavatelja. [7] Područje dijagnostike ubrzano se razvija sve većim brojem senzora koji prikupljaju razne podatke o stanju opreme. Razvojem umjetnih neuronskih mreža i strojnog učenja razvijaju se matematički modeli i algoritmi koji su u stanju razlučiti kvar od šuma signala i tako pomoći održavateljima u donošenju kvalitetnijih odluka.

2.4. Prediktivno održavanje 4.0.

Analizom velike količine podataka u održavanju predstavljena je četvrta razina zrelosti u prediktivnom održavanju. Cilj takvog načina održavanja je, temeljem napredne analitike, pronaći najučinkovitije preventivne mjere. Podaci koji se prikupljaju su podaci o korištenju, povijesti održavanju, načinu upravljanja, okolišu i svi ostali podaci koji mogu opisati način korištenja opreme. Preventivno održavanje, prikazano slikom 6. može se podijeliti u četiri razine. Prvu razinu predstavlja vizualni pregled koji se provodi periodično, a zaključci o stanju doneseni su temeljem iskustva održavatelja. Druga razina obuhvaća pregled pomoću instrumenata. Također se provodi periodički, pomoću održavatelja, a rezultati su kombinacija iskustva održavatelja i očitavanja prikupljenih instrumentima. Praćenje stanja u stvarnom vremenu karakteristično je za treću razinu prediktivnog održavanja. Na taj način stanje se provjerava kontinuirano, a obavijesti i upozorenja dobivena su ranije ustanovljenim granicama. Preventivno održavanje 4.0. predstavlja četvrtu razinu i ono se temelji na kontinuiranoj provjeri stanja, ali i prikupljenim podacima, kao što su vremenski uvjeti i način korištenja. Upozorenja koja se dobivaju temeljena su na prediktivnim tehnikama, koje relativno uspješno predviđaju dio kvarova koji se na ostalim razinama ne mogu predvidjeti. [9]



Slika 6. Razine prediktivnog održavanja [9]

Uvođenje prediktivnog održavanja baziranog na tehnologiji interneta stvari provodi se prema dijagramu sa slike 7. [10]



Slika 7. Koraci uvođenja prediktivnog održavanja temeljenog na internetu stvari [10]

2.4.1. Identificiranje opreme

Identificiranje opreme je bitno provesti kako bi se utvrdilo početno stanje smisao razvoja programa preventivnog održavanja za konkretnu opremu.

Neki od kriterija koji se mogu koristiti prilikom odabira opreme pogodne za uvođenje prediktivnog održavanja su:

- ključna operativna funkcija opreme,
- postojanje značajnih troškova popravka opreme nakon kvara,
- postojanje problema vezanih uz sigurnost uslijed nastanka kvara,
- postojanje značajnih troškova vezanih uz provođenje preventivnog održavanja,
- mogućnost pronalaska cjenovno prihvatljivog modela prediktivnog održavanja.

Sljedeće što je potrebno učiniti je analizirati svaku od izabrane opreme pomoću održavanja temeljenog na pouzdanosti (engl. *Reliability Centered Maintenance* – dalje u tekstu RCM). RCM daje procjenu mogućeg uzroka nastanka kvara pojedine opreme i pomaže razviti procedure kako bi se spriječili budući kvarovi. Počeci RCM-a datiraju iz šezdesetih godina 20. stoljeća i RCM je razvijen za potrebe zrakoplovne industrije prilikom utvrđivanja inženjera da svaki dio zrakoplova ima karakterističan životni vijek. Temeljem informacije o vremenu uporabe s većom pouzdanošću se moglo ocijeniti u kakvom je stanju pojedini dio zrakoplova. Tako se moglo predvidjeti kada će biti potrebna promjena određenog dijela.

Analiza RCM metodom za svaki kvar koji se može pojaviti na opremi može biti vrlo dugotrajna i jako skupa. Iz tog razloga analiza se provodi postavljajući sedam ključnih pitanja: [10]

1. Koje su funkcije i željeni parametri uspješnosti svake opreme?
2. Koji dio opreme ne može izvršavati svoje funkcije?
3. Koji su načini kvara prilikom svake pojave kvar?
4. Što uzrokuje određeni način kvara?
5. Koje su posljedice nastanka kvara?
6. Što se može i treba učiniti kako bi se predvidio ili spriječio kvar?
7. Što je potrebno učiniti ako se ne može odrediti odgovarajući način rješavanja kvara?

Način kvara može se podijeliti u tri grupe.

1. Kvarovi koji su se već prije pojavili na istoj ili sličnoj opremi.
2. Kvarovi koji su uspješno spriječeni postojećim održavanjem.
3. Bilo kakvi neuspjesi izvan prve dvije kategorije koje imaju mogućnost pojave u budućnosti.

2.4.2. Prikupljanje podataka

Prikupljanje podataka provodi se u realnom vremenu pomoću senzora ili kamera na promatranom dijelu opreme koji moraju biti povezani s informacijskim sustavom zaduženim za prikupljanjem podataka. Odabir metode kojom će se pratiti promjene provodi se nakon provedene RCM analize. Pritom je potrebno dati odgovor na pitanja koji će se podaci mjeriti i što želimo učiniti s podacima. Neki od primjera podataka koji se mjere su: tlak, vibracije, temperatura, napon. Podatke možemo spremiti unutar informacijskog sustava, u računalni oblak ili na neki drugi način u neku vanjsku bazu podataka.

Nakon prikupljanja podataka potrebno je prikupljene podatke pripremiti na odgovarajući način. Problemi koji se mogu pojaviti prilikom prikupljanja podataka su:

- necjelovitost – neke od vrijednosti nedostaju,
- šumovi – prikupljeni podaci sadrže pogreške u spremanju ili iskaču izvan normalnih granica,
- nekonzistentnost – podaci sadrže proturječne zapise ili odstupanja.

Uobičajeni načini pripreme podataka sadrže aktivnosti kao što su:

- brisanje podataka – brišu se podaci kojima nedostaju vrijednosti, koji sadrže šumove i koji iskaču izvan normalnih granica,
- transformacija podataka – sažimanje podataka kako bi se smanjio utjecaj nejednolike raspodjele podataka i šuma,
- smanjenje podataka – kako bi se olakšalo rukovanje podacima,
- diskretizacija podataka – konvertiranjem atributa radi lakšeg korištenja,
- čišćenje teksta – uklanjanjem znakova koji mogu uzrokovati neusklađenost podataka.

Nakon obrade podataka potrebno je odabrati one attribute temeljem kojih se provodi daljnja analiza. Smanjenje atributa posebno je važno na velikim skupovima podataka gdje određeni atributi bolje opisuju zadani skup podataka od ostalih atributa. Kako ne bi došlo do pojave smetnji, određeni atributi se uklanjaju iz daljnje analize.

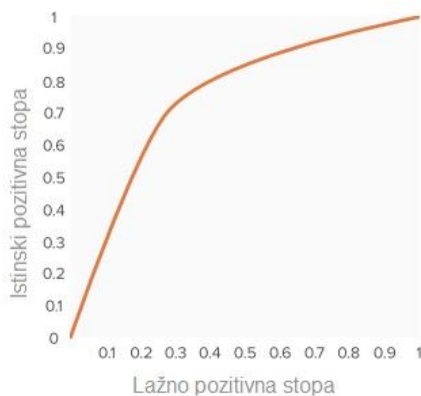
2.4.3. Dizajniranje prediktivnog modela

Dizajniranje prediktivnog modela započinje stvaranjem matematičkog modela koji je najpogodniji za konkretan slučaj. Na temelju raznih modela, kao što su: logistička regresija, stabla odlučivanja ili neuronske mreže, provode se metode strojnog učenja koje omogućuje predviđanje. Kod prediktivnog održavanja veliku ulogu ima analiza preživljavanja koja se koristi kako bi predviđela vrijeme do pojave kvara.

U sljedećem koraku dizajniranja modela potrebno je podijeliti set podatka na dva skupa – trening skup, koji u pravilu sadrži između 60 i 80% ukupnog skupa podataka i test skup, koji sadrži ostale podatke skupa. Prilikom evaluacije modela uspoređuju se modeli. Kod odabira između modela potrebno je koristiti isti trening i test skup podataka kako bi se rezultati mogli usporediti. Na kraju je potrebno odabrati onaj model koji daje najtočnije rezultate predviđanja.

Neke od metoda pomoću kojih se mjeri učinkovitost modela nabrojane su u nastavku: [10]

- **Karakteristična krivulja pouzdanosti** – vizualno predstavlja model binarne klasifikacije. Ona prikazuje stope prema kojima je pojava kvara predviđanjem okarakterizirana kao točna, a zatim se pokazalo je li točna, to jest pogrešna. slikom 8. prikazana je karakteristična krivulja pouzdanosti.



Slika 8. Karakteristična krivulja pouzdanosti [10]

- **Preciznost** – dana je kao omjer pozitivnih predviđanja. Računa se izrazom: (1)

$$\text{preciznost} = \frac{\text{istinski pozitivno}}{(\text{istinski pozitivno} + \text{lažno pozitivno})} \quad (1)$$

- **Ponovno pozivanje** – daje omjer pozitivno predviđenih pozitivnih događaja. Računa se izrazom: (2)

$$\text{ponovno pozivanje} = \frac{\text{istinski pozitivno}}{(\text{istinski pozitivno} + \text{lažno negativno})} \quad (2)$$

- **F1 Score** – daje ponderirani prosjek preciznosti i ponovnog pozivanja: (3)

$$F1 \text{ Score} = \frac{(\text{ponovno pozivanje} * \text{preciznost})}{(\text{ponovno pozivanje} + \text{preciznost})} \quad (3)$$

U slučaju prediktivnog održavanja, gdje je cijena lažno negativnih predviđanja (kvar se dogodio, ali nije predviđen) veća od lažno pozitivnih predviđanja (popravljen je nešto što neće dovesti do pojave kvara), potrebno je uzeti metodu ponovnog pozivanja kao metodu kojom će se mjeriti učinkovitost predviđanja modela.

Nakon razrade modela i odabira mjere kojom će se mjeriti učinkovitost modela započinje se s fazom testiranja modela u stvarnim uvjetima. Prilikom testiranja modela provodi se obuka zaposlenika i ispravljaju se primijećeni nedostaci na modelu.

2.4.4. Upravljanje zadatcima

Računala mogu samostalno otvarati radne naloge i upozoravati održavatelje o mogućnosti nastanka kvara, ali rezultati dobiveni modelom često se ne mogu smatrati potpuno točnima. Zato je potreban čovjek, koji će na temelju iskustva donijeti konačnu odluku hoće li se pristupiti procesu održavanja. Održavatelji temeljem iskustva dodjeljuje zadatke i određuje korake kako će se pristupiti kvaru, pritom osiguravajući sve što je potrebno kako bi se pristupilo otklanjanju kvara u trenutku koji najmanje utječe na rad cijelog sustava.

2.4.5. Učenje

Na temelju rezultata dobivenih prediktivnim modelom i ostalih akcija održavanja provodi se učenje. Cilj učenja je da se u budućnosti prilikom pojave sličnih problema donošenje odluka obavlja s većom točnošću. Jednom kada je izrađen informacijski sustav baziran na prediktivnom održavanju, može se pristupiti procesu strojnog učenja i prediktivnim modelima kako bi se odabralo najbolji način otklanjanja kvara za svaku situaciju. Time bi se omogućilo preskriptivno održavanje. Preskriptivno održavanje predstavlja novo područje održavanja koje se temelji na velikoj količini podataka i obradi velike količine podataka. Preskriptivnim održavanjem teži se automatizaciji procesa održavanja opisivanjem svakog pojedinog kvara, koji se može dogoditi te način njegovog rješavanja.

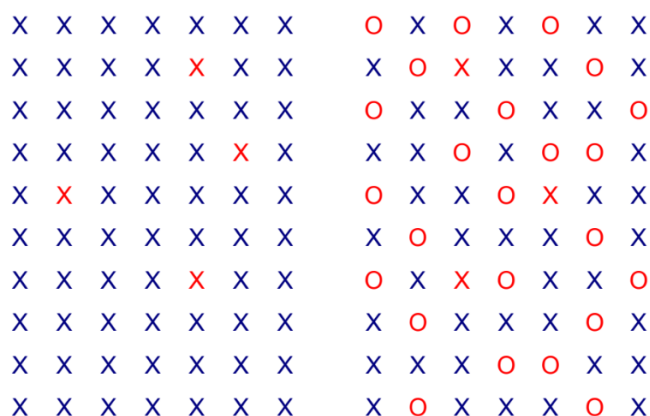
3. STROJNI I RAČUNALNI VID

Prema Automated Imaging Association, strojni vid (engl. *machine vision*) obuhvaća sve industrijske i neindustrijske primjene u kojima kombinacija hardvera i softvera pruža operativne smjernice uređajima u izrađivanju njihovih funkcija temeljene na snimanju i obradi fotografija. [11]

Računalni vid (engl. *Computer Vision*) je područje koje je doživjelo strahoviti napredak u relativno kratkoj povijesti. Napredak računalnog vida temelji se na većoj i jeftinijoj računalnoj snazi koja omogućuje kvalitetniju i bržu obradu fotografija. Primjenom umjetne inteligencije i strojnog učenja, područje računalnog vida će još više napredovati te omogućiti širu primjenu u održavanju i ostalim segmentima korištenja.

3.1. Općenito o osjetilu vida

Od pet osjetila koje čovjek posjeduje – vid, sluh, miris, okus i dodir, osjetilo vida predstavlja najvažnije osjetilo pomoću kojeg čovjek prikuplja većinu informacija iz okoline. Prema istraživanjima, pretpostavlja se da čovjek vidom prima podatke brzinama koje premašuju 10 Mbps. Većina primljenih informacija nije važna i predstavlja smetnju, ali mozak ima mogućnost apstraktno interpretirati samo mali dio važnih podataka. Pomoću vida čovjek primi najmanje dva puta više informacija u odnosu na sva druga osjetila. Interpretaciju podataka mozak obavlja unutar desetinke sekunde sa stopostotnom točnošću. Težnja znanstvenika je napraviti sustav koji će prepoznavati i interpretirati uzorke prikupljene kamerom brzinama kojima to obavlja čovjek što se danas čini nedostižnim. Za jednostavne primjere primjene računalnog vida kao što je prebrojavanje uzoraka, prikazano slikom 9., relativno je jednostavno naučiti računalno koje simbole je potrebno prepoznati prebrojati. Te je rezultate je moguće dobiti u kratkom roku. [12]



Slika 9. Primjer prebrojavanja uzoraka [13]

Za složenije zadatke primjene računalnog vida potrebno je duže učenje i kompleksniji matematički modeli koji to omogućuju. slikom 10. prikazan je primjer složenijeg zadatka računalnog vida koji iz nekoliko stotina djelomično preklapajućih fotografija rekonstruira trodimenzionalni točkasti model prostora.



Slika 10. Primjer stvaranja 3D modela prostora na temelju velikog broja 2D fotografija [13]

3.2. Sličnosti i razlike strojnog i računalnog vida

Smatra se da su strojni i računalni vid sinonimi za isti pojam, ali radi se o različitim područjima s povezanim područjem istraživanja. Ako se razmišlja o strojnom vidu kao ljudskom tijelu, računalni vid je taj koji predstavlja mrežnicu, očni živac, mozak i središnji živčani sustav. Računalni vid se prvenstveno odnosi na snimanje i automatizaciju analize fotografija s naglaskom na funkcijama vezanim uz analizu fotografija. Prema navedenom, računalni vid može se koristiti sam, bez potrebe da bude dio nekog većeg sustava. Strojni vid ne funkcionira bez računala i softvera za obradu fotografija. Pojam strojnog vida odnosi se na korištenje računalnog vida u industrijskoj i praktičnoj primjeni, to jest u procesima koji zahtijevaju informacije na temelju obrade fotografija. Potencijalna primjena strojnog vida eksponencijalno raste napretkom područja računalnog vida.

Komponente koje tvore sustave strojnog i računalnog vida u pravilu su jednake. Za snimanje fotografija koriste se kamere koje se razlikuju po vrsti senzora i objektiva. Kako bi se osigurala jasna fotografija potrebno je osvjetljenje, dok je za obradu fotografija potrebno računalo s odgovarajućim softverom.

3.3. Područja primjene računalnog vida

Računalni vid jedan je od najsloženijih procesa koji pokušava odglumiti ljudski mozak u stvaranju slike. Postavlja se pitanje kako je ljudskom mozgu tako lako stvoriti sliku objekta dok je stroju teško. Neka od područja u kojima je taj problem uspješno riješen navedena su u nastavku.

3.3.1. Obrada fotografija

Digitalna obrada fotografija koristi velik broj računalnih algoritama u procesu obrade digitalne fotografije. Obrada fotografija koristi se u raznim područjima kao što su forenzika, medicina, robotika i pametni sustavi. Digitalna fotografija sastoji se od velikog broja točkica koje se nazivaju pikseli. Svaki piksel, kod fotografije u boji, sadrži određene podatke kao što je boja, stupanj zasićenosti, nijansa i svjetlina. Neki od zadataka koji se provode prilikom obrade fotografije su čišćenje, segmentacija i geometrijska transformacija. [14]

Čišćenje fotografija provodi se zbog pojave šumova koji nastaju kao posljedica različitog stanja okoliša (na primjer razlika u osvjetljenju i temperaturi), razlike u opremi ili ljudskog utjecaja. Način rješavanja problema šumova je provođenje analize šumova kojom se dobiva model šumova temeljem kojih se lakše primjenjuje odgovarajuća strategija rješavanja problema. Neki od modela pomoću kojih se uklanjaju šumovi su: [14]

- Gaussov model – temelji se na funkciji raspodjele vjerojatnosti,
- Ujednačeni model šumova - koji pretpostavlja da će vjerojatnost pojave šuma biti ista za svaki piksel bez obzira gdje se on nalazi,
- Rayleighov šum – model koji daje deformirani oblik Gaussovog modela.

Segmentacija fotografije provodi se na temelju diferencijacije boja. To je proces grupiranja koji omogućuje da se pikseli odvoje prema intenzitetu boja. Jedan od načina segmentacije je određivanje praga koje se temelji na iskustvu korisnika. Kako bi se smanjile pogreške prilikom segmentacije razvijene su tehnike koje omogućavaju automatsko i prilagodljivo određivanje praga. Takve tehnike temelje se na principima rada neuronskih mreža, genetskih i evolucijskih algoritama. [15]

Geometrijska transformacija fotografije može se provesti na dva načina: [14]

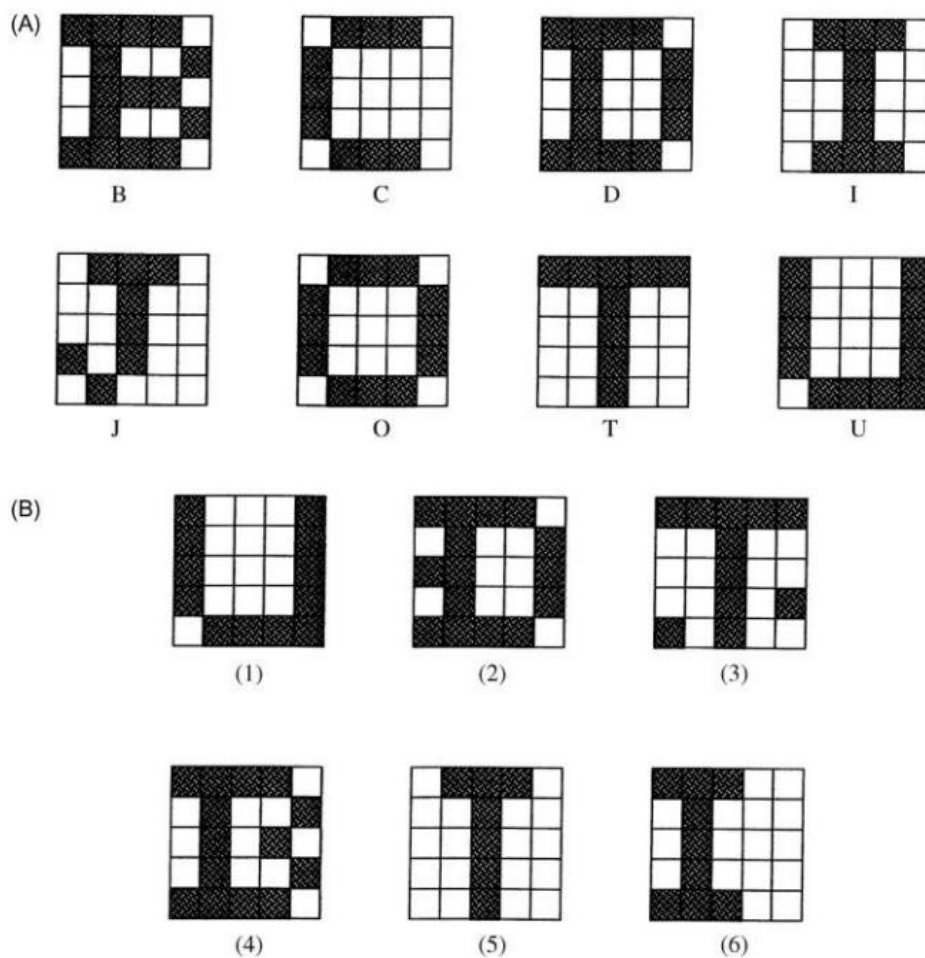
1. Prostorna transformacija fotografije - gdje se pikseli reorganiziraju,
2. Interpolacija ili ekstrapolacija fotografije – gdje se dio fotografije dodaje ili oduzima.

3.3.2. Proces prepoznavanja uzoraka

Na sljedećem primjeru objašnjena je primjena računalnog vida na naizgled vrlo jednostavnom problemu – prepoznavanja uzoraka. Strojno prepoznavanje uzoraka područje je istraživanja mnogih znanstvenih disciplina kao što su biologija, psihologija, medicina, marketing, ali i mnogih uslužnih djelatnosti. Pojam prepoznavanja uzoraka obuhvaća širok spektar aktivnosti kao što su prepoznavanje govora i pisanih znakova te otkrivanje pogrešaka u radu strojeva i njihovu dijagnostiku. [16]

Jedno od prvih prepoznavanja uzoraka korišteno je u poštanskim uredima kako bi se automatizirala podjela pisama i pritom smanjila količina ljudskog rada. Prilikom razvoja algoritma pojavili su se problemi neujednačene veličine i oblika slova što je prikazano slikom 11. Svaki uzorak prikazan slikom 11.a. može se opisati kao skup od 25 bitova informacija,

zajedno s klasom koja prikazuje tumačenje uzoraka. Računalo zatim uči uzorke i njihove klase. Na taj način računalo može prepoznati da se testni uzorak (4) prikazan na slici 11.b. pridružuje klasi B iz trening skupa podataka. [12]



Slika 11. Prepoznavanje uzoraka [12]

Međutim, ovaj pristup nije uvijek siguran. U situacijama kao što su uzorci (2), (3), (5) i (6) prikazani slikom 11.b. testni uzorci su deformirani ili dolazi do pojave šuma, što može dovesti do pogrešnog tumačenja. Problem pogrešnog tumačenja u održavanju je od velikog značaja jer čak i mali utjecaj šuma ili deformacija može utjecati na operativno stanje stroja. Način rješavanja takvog problema može se pronaći prikupljanjem velikog skupa podataka temeljem kojih je moguće provesti strojno učenje.

3.3.3. Praćenje objekata

Praćenje više objekata važan je problem u računalnom vidu koji ima široku primjenu u raznim područjima kao što je vizualni nadzor, sportska analiza, navigacija robota ili autonomna vožnja. Praćenje objekata omogućeno je primjenom strojnog učenja. Praćenje objekata u područjima navigacije robota ili prilikom autonomne vožnje mora se obavljati kontinuirano i u stvarnom vremenu, što predstavlja značajan izazov za računala koja moraju obraditi veliku količinu podataka u kratkom vremenskom periodu. Glavni problem obrade podataka predstavljaju podaci koji nisu relevantni i koji predstavljaju šum. Kako bi se riješili takvi problemi, koriste se algoritmi koji donose odluke na temelju kombinacije različitih parametara kao što je izgled, brzina kretanja i lokacija. [17]

3.4. Prednosti upotrebe strojnog vida

Strojni vid se ne može mjeriti s ljudskim vidom u kvalitativnom tumačenju složenih i nestrukturiranih podataka. Kada je riječ o kvantitativnom razlučivanju podataka, strojni vid nadmašuje mogućnosti ljudskog vida zbog svoje brzine, točnosti i ponovljivosti. Primjer upotrebe strojnog vida je pregled proizvodne linije gdje se pomoću strojnog vida može pregledati tisuće dijelova u minuti. Strojnim vidom moguće je pregledati detalje predmeta premalih da bi ih ljudsko oko uopće vidjelo. Strojnom vidu nije potreban kontakt s predmetom promatranja te se na taj način sprječava oštećenje dijelova te se smanjuju troškovi održavanja mehaničkih dijelova. Upotrebom strojnog vida povećava se sigurnost radnog okruženja jer se smanjuje potreba ljudskog rada u uvjetima štetnim za ljudsko zdravlje.

Tablicom 2. dani su primjeri kako se upotrebom strojnog vida pomaže u ispunjavanju strateških ciljeva poduzeća. [11]

Tablica 2. Upotreba strojnog vida prilikom ostvarivanja strateških ciljeva

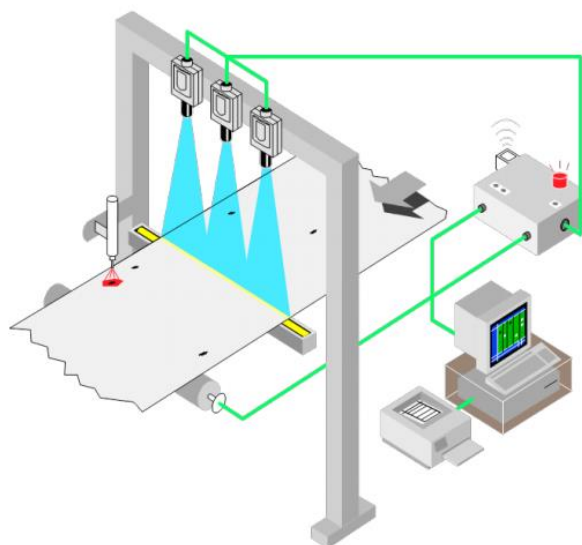
Strateški cilj	Načini postizanja strateškog cilja primjenom strojnog vida
Povećanje kvalitete	Kontrola, mjerenje, ocjenjivanje i provjera točnosti montaže.
Povećanje produktivnosti	Ponavljajući zadaci koji su se ranije obavljali ručno sada se obavljaju pomoću strojnog vida.
Fleksibilnost proizvodnje	Mjerenjem i ocjenjivanjem; vođenjem robota; provjeravanjem rada.
Smanjenje zastoja i kraće vrijeme prilagodbe	Promjene se unaprijed programiraju.
Potpunije informacije i stroža kontrola procesa	Zadaci koji se obavljaju ručno sada mogu imati podršku preko računala.
Smanjenje početnih troškova opreme	Dodavanjem strojnog vida stroju poboljšava se učinkovitost i izbjegava zastarijevanje.
Smanjenje troškova proizvodnje	Sustav strojnog vida zamjenjuje zaposlenike i otkriva nedostatke ranije u procesu.
Smanjenje otpada	Pregledom i mjerenjem.
Upravljanje zalihama	Prepoznavanje i identifikacija optičkih znakova – na primjer barcode.
Smanjenje potrebne površine	Smanjenjem broja zaposlenika.

3.5. Vrste strojnog vida

Ovisno o području primjene strojni vid se može podijeliti u tri kategorije koje se razlikuju prema broju dimenzija. Prema tome sustavi strojnog vida dijele se na jednodimenzionalne, dvodimenzionalne i trodimenzionalne.

3.5.1. Jednodimenzionalni sustavi strojnog vida

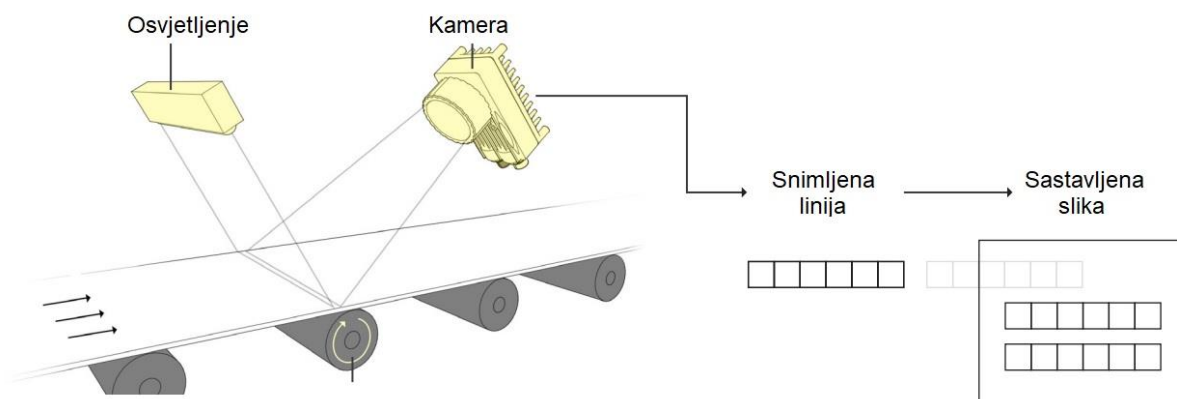
Jednodimenzionalni sustavi predstavljaju najjednostavniji oblik strojnog vida. Takvi sustavi stvaraju linijsku sliku. Uobičajeno se koriste na linijskim konvejerima za otkrivanje i klasificiranje pogreške na materijalima koji se proizvode u kontinuitetu kao što je proizvodnja papira, metala, plastične ambalaže i proizvoda u rolama. Slikom 12. prikazan je jednodimenzionalni sustav strojnog vida. Sustav se sastoji od kamera smještenih okomito na linijski konvejer. Snimke se u stvarnom vremenu prenose na računalo koje vrši obradu. Ukoliko je pronađena pogreška u proizvodnji, obavještava se osoba koja nadzire proizvodnju kako bi poduzela daljnje korake.



Slika 12. Prikaz linijskog konvejera s jednodimenzionalnim sustavom strojnog vida [11]

3.5.2. Dvodimenzionalni sustavi strojnog vida

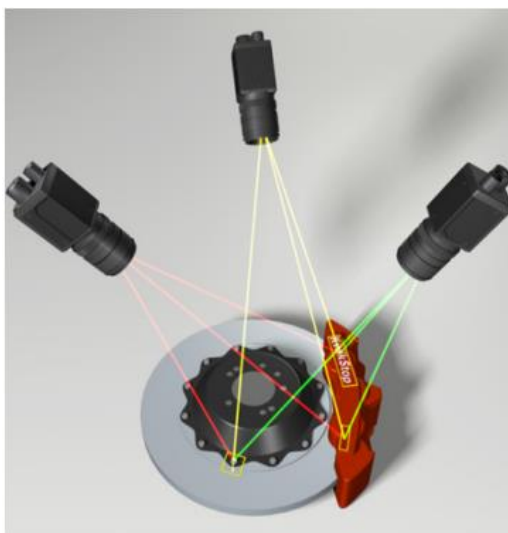
Dvodimenzionalni sustavi strojnog vida predstavljaju najčešće korišten sustav u industriji. Oni omogućavaju provjeru objekata koji se ne pomiču na konvejeru u različitim rezolucijama, dok kod objekata na konvejeru dvodimenzionalni sustav stvara fotografiju liniju po liniju. Kod implementacije dvodimenzionalnog strojnog vida potrebno je donijeti odluku o vrsti kamere koja će se koristiti. Ukoliko se koristi kamera u boji, čovjeku koji promatra fotografiju, olakšat će se raspoznavanje objekta. Međutim, za omogućavanje brže obrade snimljenih fotografija preporučuje se upotreba fotografija snimljenih u sivim tonovima. Slikom 13. prikazana je upotreba dvodimenzionalnog sustava strojnog vida na primjeru linijskog konvejera.



Slika 13. Prikaz linijskog konvejera s dvodimenzionalnim sustavom strojnog vida [11]

3.5.3. Trodimenzionalni sustavi strojnog vida

Trodimenzijski sustavi strojnog vida se za dobivanje treće dimenzije služe različitim tehnikama kao što su: laserska triangulacija, sjenčanje ili različit razmještaj osvjetljenja. Snimanje objekata u tri dimenzije sadrži veliku količinu podataka. Prije uvođenja treće dimenzije potrebno je provjeriti isplativost i mogućnosti ostalih komponenti sustava zaduženih za spremanje i obradu takvih podataka. Kod uvođenja strojnog vida u tri dimenzije potrebno je odabrati između sustava koji omogućavaju točnost mjerenja objekta ili onih koji omogućavaju snimanje veće površine uz smanjenu točnost. Slikom 14. prikazan je trodimenzionalni sustav strojnog vida koji se sastoji od tri kamere koje pomoću triangulacije izvode snimanje u tri dimenzije.



Slika 14. Trodimenzionalno snimanje površine [11]

4. ALATI ZA OBRADU SLIKE I RAČUNALNI VID

MATLAB je programska platforma posebno dizajnirana za inženjere i znanstvenike kako bi im omogućila brze i jednostavne izračune. MATLAB koristi posebno razvijen programski jezik koji u odnosu na ostale tradicionalne programske jezike, kao što su C/C++ ili Java, omogućuje najprirodnije korištenje računske matematike. Posebnost MATLAB-a je u tome što sadrži velik broj ugrađenih funkcija i alata (engl. *toolbox*) iz različitih znanstvenih područja. Alati koji će biti pobliže objašnjeni u ovom radu su Alat za obradu slike (engl. *Image Processing Toolbox*) i Alat za računalni vid (engl. *Computer Vision Toolbox*).

Radi lakšeg razumijevanja potrebno je razjasniti nekoliko osnovnih pojmova koji će se koristiti dalje kroz rad:

- **naredbeni prozor** (engl. *Command Window*) omogućuje unos izraza u naredbeni redak (engl. *command line*) kako bi se dobio traženi rezultat.
- **radni prostor** (engl. *workspace*) sadrži varijable koje su stvorene ili uvezene u MATLAB iz drugih programa. Sadržaj spremljen u radnom prostoru može se uređivati odabirom željene varijable ili kroz naredbeni prozor.
- najjednostavnija vrsta MATLAB programa naziva se **skripta** (engl. *script*). Skripta je datoteka koja sadrži više redova naredbi i poziva funkcija. Skripta se može pokrenuti upisivanjem imena skripte u naredbeni redak.
- **naredba** (engl. *command*) predstavlja preddefiniranu MATLAB funkciju.
- kako bi MATLAB bio u mogućnosti izvršiti tražene zadatke koriste se **funkcije** koje mogu samostalno izraditi ili se koriste već postojeće dostupne u MATLAB-ovoj bazi.
- **varijable** služe za spremanje pozvanih funkcija. Ime varijable mora početi sa slovom, a zatim se mogu koristiti brojevi ili donja crtica. MATLAB razlikuje malo i veliko slovo. Na taj način isto slovo predstavlja dvije različite varijable, a u slučaju ponavljanja varijable MATLAB pamti samo zadnju unesenu.
- **polja** (engl. *array*) predstavljaju elemente koji opisuju varijable.

4.1. Alati za obradu slike

Alat za obradu slike (engl. *Image Processing Toolbox*) pruža sveobuhvatni skup standardnih algoritama i aplikacije koje omogućuju obradu fotografija, analizu, vizualizaciju i razvoj algoritama. Uz pomoć ovog alata moguće je obrađivati kako dvodimenzionalne, tako i trodimenzionalne fotografije i na njima izvoditi akcije segmentacije i poboljšanja fotografije,

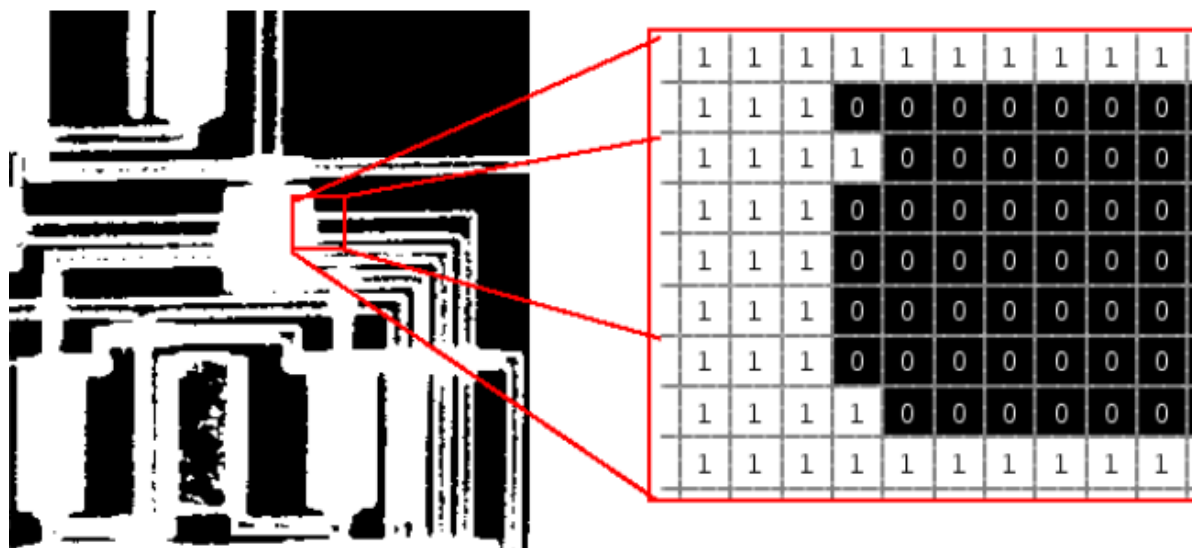
smanjenje šuma, geometrijske transformacije i usklađivanje fotografija (engl. *image registration*) koje su prikupljene iz dva različita izvora. Usklađivanje fotografija može se izvoditi tradicionalnim tehnikama obrade fotografija, ali sve je češća upotreba strojnog učenja. Pomoću MATLAB-a i alata za obradu slike moguće je automatizirati tradicionalne radne procese koji se koriste prilikom obrade fotografija. Kao izvor literature za opis alata i funkcija korišten je korisnički MATLAB-ov korisnički priručnik. [18]

4.1.1. Vrste slika u alatu za obradu slike

Alat za obradu slike definira četiri osnovna tipa fotografija koje se mogu obrađivati s prikazanim alatom. Glavna razlika između definiranih tipova slika je u načinu na koji MATLAB interpretira vrijednosti intenziteta elementa polja.

4.1.1.1. Binarna slika

Binarna slika (engl. *binary image*) svaki piksel prikazuje kao jednu od dvije moguće vrijednosti, a to su: 1 ili 0. Binarna slika spremljena je u logičkom polju (engl. *logical array*). Prilikom opisa korištenog koda varijabla `BW` odnosi se na binarnu sliku. Slika 15. prikazuje vrijednosti piksela na binarnoj slici.



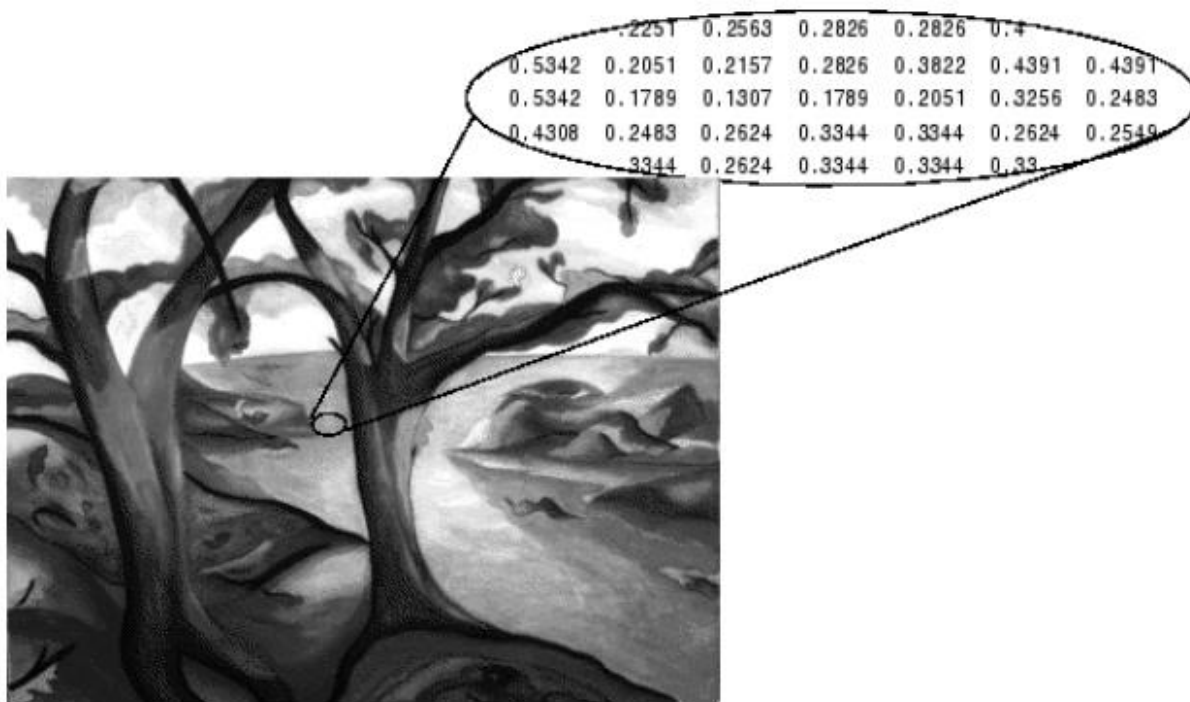
Slika 15. Vrijednosti piksela na binarnoj slici [18]

4.1.1.2. Indeksirana slika

Indeksirana slika (engl. *indexed image*) sastoji se od polja i matrice boja. Na taj način slici se mijenja intenzitet boja slike i moguće je prilagoditi određene vrijednosti prema potrebama.

4.1.1.3. Slika sivih tonova

Slika sivih tonova (engl. *grayscale image*) nastaje na način da se iz slike u boji uklone vrijednosti nijanse i zasićenosti, ali se zadržavaju vrijednosti koje utječu na svjetlinu slike. Vrijednost svjetline 0 označava crnu boju, a vrijednost svjetline 1 označava bijelu boju. Slikom 16. prikazana je slika koja ima vrijednost pola double te se na taj način prikazuju i ostali tonovi između crne i bijele boje.



Slika 16. Vrijednost piksela na slici sivih tonova [18]

4.1.1.3. Slika u boji

Slika u boji (engl. *truecolor image*) svaki piksel slike određuje s tri vrijednosti – po jedna za crvenu, plavu i zelenu komponentu boje svakog piksela. MATLAB sliku u boji sprema u tri podatkovna polja (engl. *data array*) veličine $m \times n$. Svako polje definira po jednu komponentu boje. Slikom 17. prikazana je vrijednost svakog polja za određeni piksel na slici.



Slika 17. Vrijednosti piksela slike u boji [18]

U nastavku će biti prikazane i opisane osnovne funkcije programskog paketa MATLAB koje će biti korištene prilikom razvijanja zadanog primjera.

4.1.2. Usklađivanje fotografija

Usklađivanje fotografija (engl. *image registration*) je postupak poravnanja dvije ili više fotografija koje prikazuju isti predmet, ali uz određeni odmak. Ovaj postupak provodi se na način da se odabere jedna fotografija kao referentna, dok se na drugoj slici provode geometrijske transformacije i lokalni pomaci u svrhu poravnanja s referentnom fotografijom. Razlozi zbog kojih fotografije nisu usklađene mogu biti promjene uvjeta pod kojima su fotografije snimljene, promjena perspektive kamere, izobličenje senzora ili razlike između uređaja za snimanje. Usklađivanje fotografija predstavlja prvi korak obrade fotografija, dok naknadnim obradama iz fotografija izvlači znanje. Usklađivanje fotografija koristi se u medicini ili prilikom poravnanja satelitskih snimaka.

3.2. Alat za računalni vid

Alat za računalni vid (engl. *Computer Vision System Toolbox*) nudi algoritme, funkcije i aplikacije za projektiranje i simuliranje računalnog vida i sustava za obradu videa. Pomoću ovog alata moguće je izvoditi obradu videozapisa, otkrivati pouzdanosti između objekata kao i praćenje objekata. Alat za računalni vid sadrži osnove strojnog učenja pomoću kojeg je moguće trenirati detekciju pokreta kao i prepoznavanje te pronalaženje objekata. [19]

3.3. Korištene funkcije MATLAB-ovih alata

3.3.1. Osnove učitavanja i prikaza fotografija

Učitavanje fotografije u radni prostor MATLAB-a vrši se pomoću naredbe `imread`.

```
A = imread(filename)
```

gdje `A` predstavlja varijablu u koju se sprema fotografija.

Naredba `imread` učitava fotografiju određenu varijablom `filename` koja predstavlja putanju do željene fotografije spremljene na lokalnom disku. Ukoliko je fotografija spremljena u istoj mapi gdje je spremljena i `.m` skripta nije potrebno unositi cijelu putanju već samo naziv datoteke i njezinu ekstenziju. Ako je varijablom `filename` učitana datoteka koja sadrži više fotografija učitat će se samo prva datoteka u nizu.

Pretvaranje fotografija u sive tonove vrši se pozivanjem funkcije `rgb2gray` koja kao varijablu prima fotografiju u boji koju zatim pretvara u sliku intenzivnih sivih tonova uklanjanjem informacija o nijansi i zasićenosti, ali zadržavajući svjetlinu.

```
I = rgb2gray(RGB)
```

MATLAB nije pogodan za grafički prikaz rezultata. Da bi se omogućio grafički prikaz potrebno je otvoriti novi prozor što se postiže pozivanjem naredbe `figure` i funkcije `imshow` koja omogućuje prikaz fotografija u prozoru koji se pritom prilagođava veličini slike.

```
figure  
imshow(RGB)
```

Usporedni prikaz fotografija ostvaruje se pozivanjem funkcije `imshowpair` koja za varijable ima fotografije i jednu od metoda s kojima prikazuje kombinaciju slika A i B. Metode su pokazane u tablici 3.

Tablica 3. Vrste metoda za prikaz slika

Vrijednost	Opis
'falsecolor'	Stvara sliku koja prikazuje slike A i B u različitim tonovima boje. Sive regije na prikazanoj slici prikazuju područja na kojima obje slike imaju isti intenzitet. Područja koja su označena ljubičasto predstavljaju sliku A, dok područje označeno zeleno predstavlja sliku B. To su područja gdje su intenziteti boje različiti.
'blend'	Stvara preklap slike A i B.
'checkboard'	Stvara sliku s naizmjeničnim pravokutnim područjima od slike A i B.
'diff'	Stvara razliku slike između slika A i B.
'montage'	Stvara sliku tako da postavi slike A i B jednu pokraj druge.

```
imshowpair(A,B,method)
```

Sintaksom `hold on` omogućuje se pozivanje novog grafikona (`plot`) koji kao ulazne parametre ima matrice vrijednosti koje pokazuju željenu točku te joj uz pomoć varijable `LineSpec` dodjeljuju simbol i boju prikaza. Kako bi se vrijednosti grafikona vratile na zadane postavke potrebno je pozvati sintaksu `hold off`.

```
hold on
plot (X,Y,LineSpec)
hold off
```

Dodavanje naslova na prikazanu sliku ostvaruje se pozivanjem funkcije `title`. Vrijednost koja se unese unutar zagrada ispisuje se iznad slike ili grafikona koji se pokazuje.

```
title(txt)
```


3.3.2. Funkcije za usklađivanje fotografija

Usklađivanje fotografija provodi se stvaranjem optimizatora i metrike, podešavanjem svojstva operatora i zaključno usklađivanjem fotografija.

Stvaranje optimizatora i metrike vrši se funkcijom `imregconfig` koja stvara optimizator (`optimizer`) i metriku (`metric`) na način da uspoređuje dane fotografije temeljem intenziteta, kako bi vratio zadane postavke konfiguracije.

Modalitet (`modality`) opisuje način na koji je fotografija snimljena i tako razlikuje 'monomodal' (slične vrijednosti osvjetljenja i kontrasta na fotografijama) ili 'multimodal' (različite vrijednosti osvjetljenja i kontrasta na fotografijama).

```
[optimizer, metric] = imregconfig(modality)
```

Podešavanje svojstva optimizatora provodi se kako bi se, što je moguće više, uskladile fotografije. Odabiranjem optimalnih vrijednosti smanjuje se vrijeme potrebno za obradu i dobivanje parametara. U slučaju odabira nedostatnih vrijednosti rezultat obrade ne mora pokazati najbolju vrijednost.

Parametri s kojima se utječe na rezultat obrade su:

```
optimizer.InitialRadius
```

Njime se odabire veličina radijusa pretraživanja i vrijednost s kojom je određena je pozitivan broj. Ako je parametar `InitialRadius` definiran s velikom vrijednosti, vrijeme izračunavanja se smanjuje, ali prevelike vrijednosti parametra `InitialRadius`-a mogu rezultirati optimizacijom koja ne uspijeva konvergirati ka globalnom maksimumu, već „zapne“ u lokalnom maksimumu.

```
optimizer.Epsilon
```

Epsilon je parametar s kojim se određuje minimalna veličina radijusa pretraživanja. Epsilon je također određen pozitivnim brojem i on kontrolira točnost konvergiranja podešavanjem minimalne veličine radijusa pretraživanja. Podešavanjem Epsilon-a na manju vrijednost, izračun optimizacije metrike traje dulje, ali daje točnije rezultate. Sukladno tome podešavanjem Epsilon-a na veću vrijednost vrijeme računanja se smanjuje na račun točnosti.

```
optimizer.GrowthFactor
```

`GrowthFactor` predstavlja faktor rasta radijusa pretraživanja. Vrijednost parametra izražena je kao pozitivan broj. Optimizator koristi `GrowthFactor` za kontrolu brzine kojom raste radijus pretraživanja u prostoru parametara. Isto kao i s ostalim parametrima, ako se vrijednost parametra `GrowthFactor` postavi na malu vrijednost, optimizacija je sporija, ali vjerojatnost približavanja boljem rješenju. Ako se vrijednost parametra postavi na veliku vrijednost, optimizacija je brža, ali može rezultirati pronalaženjem samo lokalnih ekstrema funkcije.

```
optimizer.MaximumIterations
```

Maksimalan broj iteracija optimizatora određen je parametrom `MaximumIterations` koji se definira pozitivnim cijelim brojem. On određuje maksimalni broj iteracija koje optimizator može obaviti. U slučaju da se konačno rješenje postigne prije postizanja maksimalnog broja iteracija proces se završava i prije dostizanja zadanog broja iteracija.

Usklađivanje fotografija, dobivenim vrijednostima optimizatora i metrike, provodi se funkcijom `imregister` koja transformira sliku s pomakom (`moving`) kako bi bila poprimila vrijednosti referentne slike (`fixed`). Kako bi transformacija slike bila uspješna obje slike moraju biti istih dimenzija. Varijabla optimizator (`optimizer`) predstavlja objekt koji opisuje metodu optimizacije metrike, dok metrika (`metric`) predstavlja objekt koji definira kvantitativnu mjeru sličnosti između slika za optimizaciju. Varijablom `transformType` određuje se vrsta geometrijske transformacije koja se izvodi na slici s pomakom (`moving`). Vrste geometrijske transformacije koje se mogu izvoditi nad slikama prikazane su tablicom 4.

Tablica 4. Vrste geometrijskih transformacija slika

Vrijednost	Opis
'translation'	U slučaju 2-D slike izvodi se translacija po (x,y) osi, dok se u slučaju 3-D transformacije izvodi translacija po (x,y, z) osi.
'rigid'	Rigid je vrsta transformacije koja se sastoji od translacije i rotacije.
'similarity'	Transformacija se provodi na temelju sličnosti. Sastoji se od translacije, rotacije i promjene mjerila.
'affine'	Affine je najkompleksniji oblik transformacije koji se sastoji od translacije, rotacije, promjene mjerila i smicanja.

```
moving_reg = imregister(moving, fixed, transformType,
                        optimizer, metric)
```

3.3.3. Funkcije za pronalaženje razlika između fotografija

Nakon što su fotografije usklađene potrebno je pronaći promjene između fotografija. Funkcija pomoću koje se pronalaze razlike naziva se `imabsdiff`. Ona, kao rezultat, vraća apsolutnu razliku između svakog elementa iz polja (engl. *array*) `Y` koji odgovara elementu u polju `X`.

```
Z = imabsdiff(X, Y)
```

Za pronalaženje najveće vrijednosti promjene između slika potrebno je iz varijable `Z` pronaći najveću vrijednost unutar polja vrijednosti. Za pronalaženje najveće vrijednosti potrebno je pozvati funkciju `max`.

```
M = max(A)
```

Funkcijom `find` pronalaze se koordinate u obliku matrice retka (`row`) i stupca (`col`) gdje promjena poprima najveću vrijednost izračunatu prethodnim korakom.

```
[row, col] = find(____)
```

U nastavku je potrebno odrediti prag osjetljivosti. Za određivanje praga osjetljivosti potrebno je usporediti varijablu u kojoj je sačuvana vrijednost promjene između slika (`Z`) i usporediti ju s brojčanom vrijednošću koja predstavlja prag osjetljivosti. Kao rezultat dobivena je slika na

kojoj će sve vrijednosti razlike manje od zadanog broja imati vrijednost 0, dok će vrijednosti razlike veće od zadanog broja ostati nepromijenjene.

Čišćenje šumova s fotografije praga provodi se funkcijom `bwareaopen` koja s dane binarne slike (BW) uklanja sve objekte manje od `P` piksela i kao rezultat daje novu binarnu sliku BW2.

```
BW2 = bwareaopen (BW,P)
```

Prethodno opisane promjene mogu se prikazati na način da se na novu sliku „prilijepi“ maka s prethodno pronađenim promjenama. Da bi to bilo moguće potrebno je pozvati funkciju `imoverlay` koja uz pomoć varijable `color` popunjava prazna mjesta na binarnoj slici i prikazuje ju preko „originalne“ fotografije.

```
B = imoverlay(A,B,color)
```

3.3.4. Ostale funkcije

Uvjetna naredba `if` sastoji se od *izraza* (engl.*expression*) koji, ako je zadovoljen, to jest nije jednak nuli, izvodi naredbu (engl.*statements*) koja slijedi iza uvjeta. Ukoliko naredba `if` nije zadovoljena provjerava se uvjet *izraza* naredbe `else`. Isto tako, ako je uvjet zadovoljen slijedi izvršavanje naredbe. `If` naredba završava ključnom riječi `end` nakon koje se nastavlja izvršenje programa.

```
if expression
    statements
else expression
    statements
end
```

Sintaksom `isempty` provjerava se da li je vrijednost polja varijable `A` prazno polje i u tom slučaju vraća logičku vrijednost 0 (`false`) ili u slučaju kada nije prazno polje vraća logičku vrijednost 1 (`true`).

```
TF = isempty(A)
```

Sintaksa `height` vraća broj redova u tablici `T`.

```
H = height(T)
```

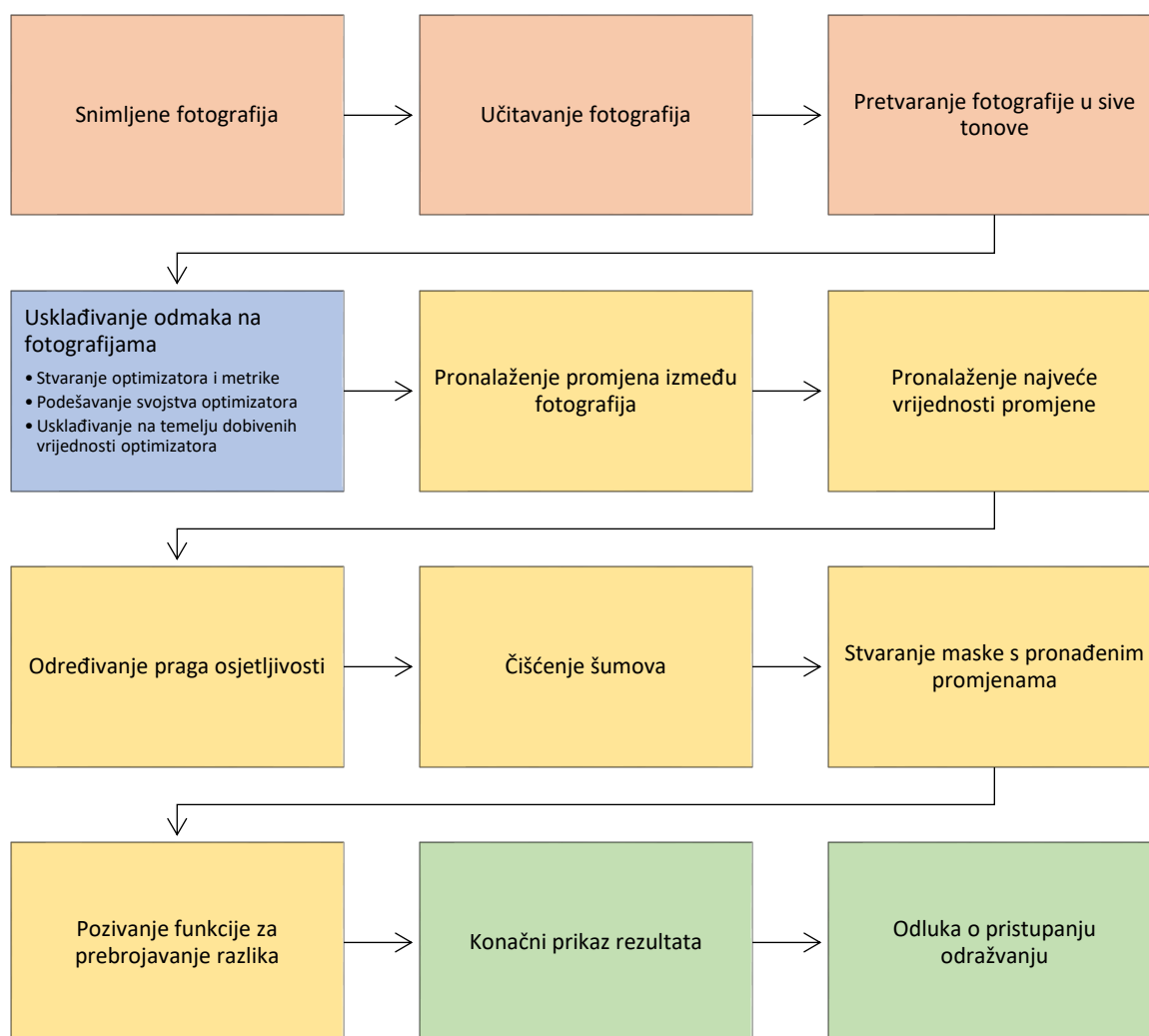
Funkcija `insertShape` vraća sliku u boji s označenim mjestima koja su dobivena pomoću varijable `position`.

```
RGB = insertShape(I, shape, position)
```

5. PRIMJENA RAČUNALNOG VIDA NA PRIMJERU ODRŽAVANJA ZRAKOPLOVA

Računalni vid korišten je za otkrivanje nedostataka na trupu zrakoplova Croatia Airlinesa. Fotografije korištenu u radu dobivene su snimanjem pomoću bespilotne letjelice, a za obradu podataka korištene su funkcije programskog paketa MATLAB R2018a.

5.1. Prikaz dijagrama toka programa za otkrivanje promjena



Slika 18. Dijagram toka programa za otkrivanje promjena na trupu zrakoplova

Dijagram toka programa koji je prikazan slikom 18. može se podijeliti u četiri faze.

Prva faza vezana je uz prikupljanje, učitavanje i pripremu fotografija iz čega proizlazi da je prvi korak dijagrama toka snimanje fotografija. Fotografije se snimaju pomoću bespilotne letjelice u vrijeme kada je zrakoplov prizemljen. Snimljene fotografije učitavaju se u aplikaciju. Prilikom učitavanja važno je odvojiti fotografiju koja prikazuje početno stanje od one koja prikazuje novo stanje. Kako bi se mogla provesti daljnja analiza fotografije se pretvaraju u sive tonove kako bi se uklonile informacije o zasićenosti. Vrijednosti piksela se ograničavaju kako bi se omogućila točnija analiza. U drugoj fazi dijagrama toka odvija se jedan od najbitnijih dijelova aplikacije za otkrivanje promjena - usklađivanje odnosa na fotografijama. Usklađivanje odnosa provodi se u tri koraka. Prvi korak je stvaranje optimizatora i metrike. U drugom koraku podešavaju se svojstva optimizatora. Podešavanjem svojstava utječe se na točnost usklađivanja, a porastom točnosti produžuje se vrijeme obrade. Stoga je potrebno optimalno prilagoditi parametre kako bi se dobio najbolji rezultat u najkraćem mogućem vremenu. Zaključno, u trećem koraku, provodi se usklađivanje fotografija temeljem dobivenih vrijednosti optimizatora. U trećoj fazi dijagrama toka vrši se usporedba početne fotografije i nove fotografije na kojoj je provedeno usklađivanje u drugoj fazi. Pronalaze se sve promjene među fotografijama te se određuje prag osjetljivosti kojim se uklanjaju manje promjene kako bi se dobio veći kontrast na slici. Čišćenje šumova provodi se kako bi se manje površine isključile iz obrade. Temeljem očišćene slike stvara se maska koja sadrži pronađene promjene. Zaključno, poziva se funkcija koja prebrojava promjene, označava ih i ispisuje njihov broj. U četvrtoj fazi, prikazuje se konačni rezultat obrade, koji je moguće pobliže pregledati. Temeljem pregleda donosi se odluka o pristupanju održavanju.

5.2. Učitavanje fotografija

Korištenjem naredbe `imread` fotografije se učitavaju u radni prostor i spremaju se u varijable naziva `pocetna` koja označava fotografiju na kojoj nema vidljivih oštećenja i ona predstavlja početno stanje koje je mjerodavno za sve buduće promjene. Varijablom `nova` označena je fotografija na kojoj su umjetno napravljene greške kako bi se ispitala funkcionalnost korištenog alata. Fotografije koje su korištene za ispitivanje rada i oznake fotografija dane su tablicom 5.

Tablica 5. Prikaz korištenih fotografija

Naziv fotografije	Orijentacija u prostoru	Opis simbola
DJI_0200.jpg		Originalno snimljena fotografija.
DJI_0200_n.jpg		Nova fotografija koja je izmijenjena uz pomoć računala.
DJI_0200_g.jpg		Nova izmijenjena fotografija koja je translatirana prema gore.
DJI_0200_d.jpg		Nova izmijenjena fotografija koja je translatirana prema dolje.
DJI_0200_l.jpg		Nova izmijenjena fotografija koja je translatirana prema lijevo.
DJI_0200_r.jpg		Nova izmijenjena fotografija koja je translatirana prema desno.
DJI_0200_gl.jpg		Nova izmijenjena fotografija koja je translatirana prema gore-lijevo.
DJI_0200_gr.jpg		Nova izmijenjena fotografija koja je translatirana prema gore-desno.
DJI_0200_dl.jpg		Nova izmijenjena fotografija koja je translatirana prema dolje-lijevo.
DJI_0200_dr.jpg		Nova izmijenjena fotografija koja je translatirana prema dolje-desno.

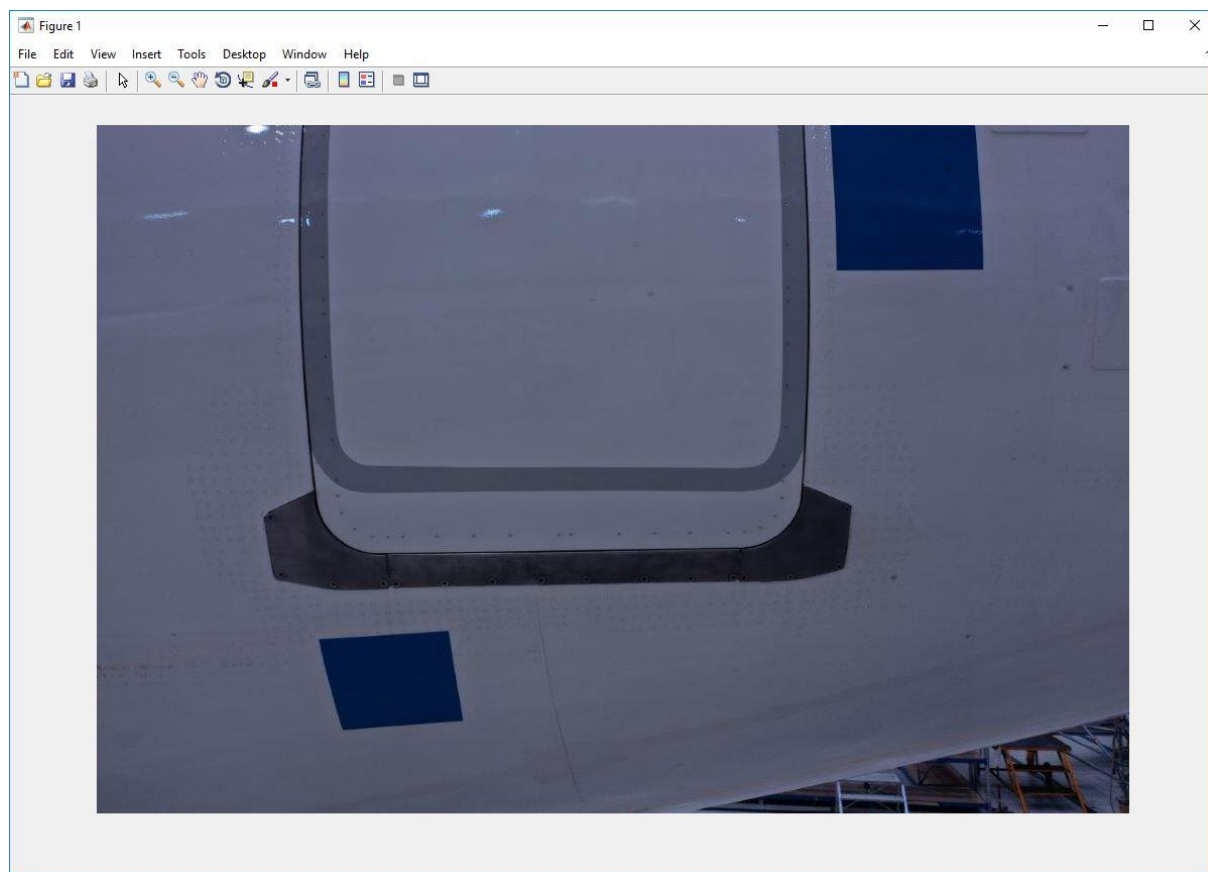
```
pocetna = imread('DJI_0200.jpg');
nova = imread('DJI_0200_n.jpg');
```

5.3. Prikaz rezultata

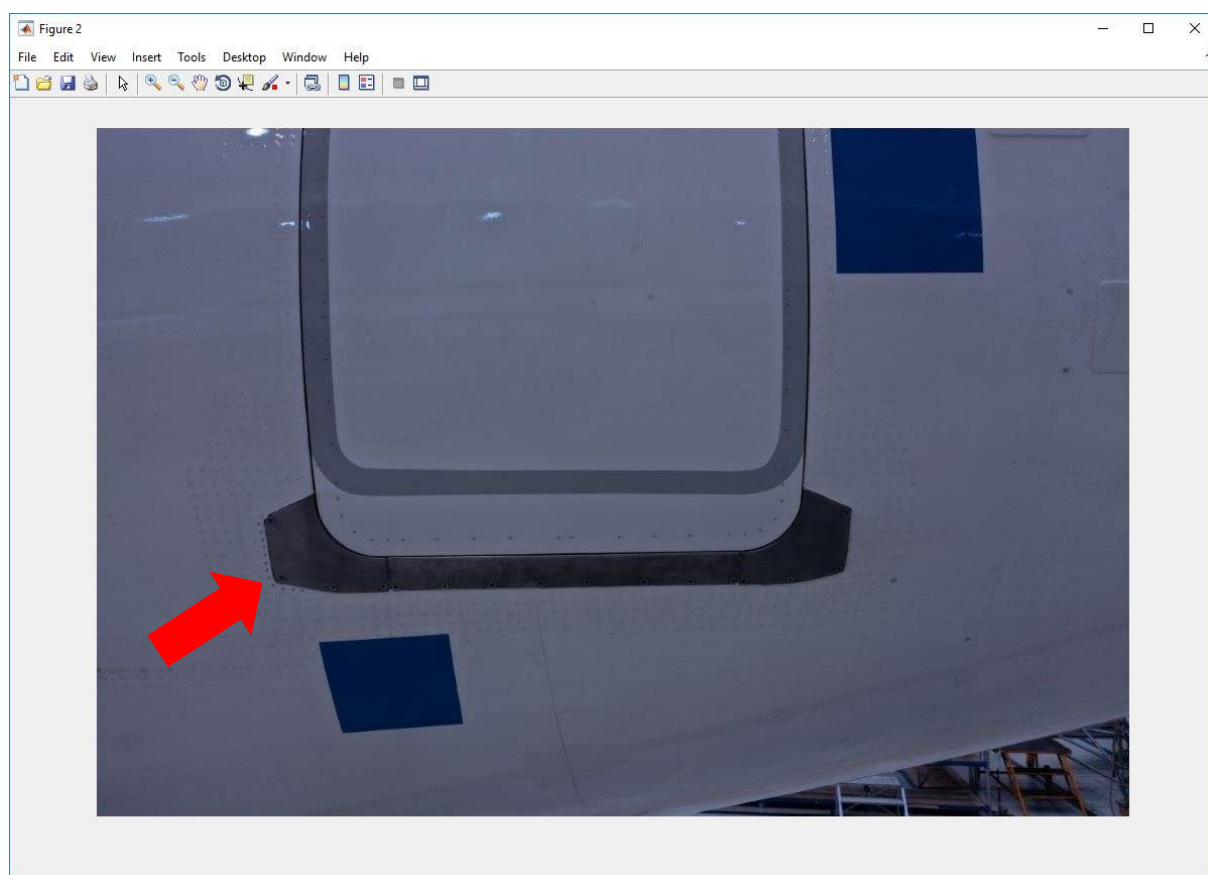
Prikaz učitanih fotografija ostvaruje se upisivanjem sljedeće sintakse:

```
figure
imshow(pocetna)
figure
imshow(nova)
```

Početna fotografija prikazana je na slici 19., a nova fotografija na slici 20.



Slika 19. Početna fotografija



Slika 20. Nova fotografija

Na slici 20. crvenom strelicom prikazano je područje na kojem su umjetno napravljene razlike, dok je na slici 21. uvećan prikaz tog područja.



Slika 21. Uvećano područje razlika

5.4. Pretvaranje fotografije u boji u sive tonove

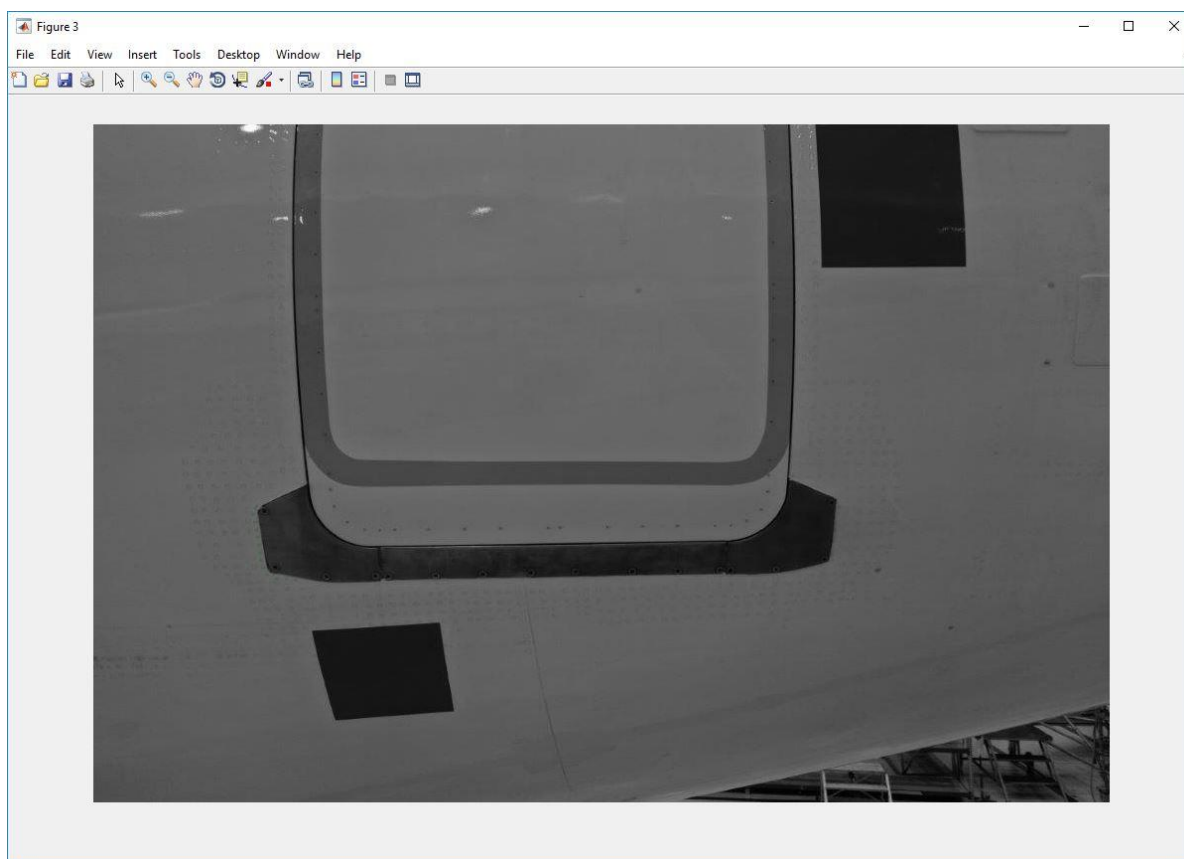
Kako bi se pretvorene fotografije spremile potrebno je napraviti nove varijable `pocetnaCB` koja sprema crno bijelu sliku pocetne fotografije i varijablu `novaCB` koja sprema crno bijelu sliku nove fotografije.

```
pocetnaCB = rgb2gray(pocetna);  
novaCB = rgb2gray(nova);
```

Usporedni prikaz fotografija ostvaruje se pozivanjem funkcije `imshowpair` koja za varijable fotografije i jednu od metoda s kojima prikazuje kombinaciju slika A (`pocetnaCB`) i B (`novaCB`). U prikazanom primjeru odabrana je metoda prikaza `'falsecolor'` koja omogućuje prikaz slika A i B u različitim tonovima boje. Područja na slici koja imaju isti intenzitet boje prikazana su u sivim nijansama, dok područja označeno ljubičastom bojom predstavlja početnu sliku (`pocetnaCB`), a zelena područja predstavljaju novu sliku (`novaCB`).

```
figure  
imshowpair(pocetnaCB,novaCB,'falsecolor')
```

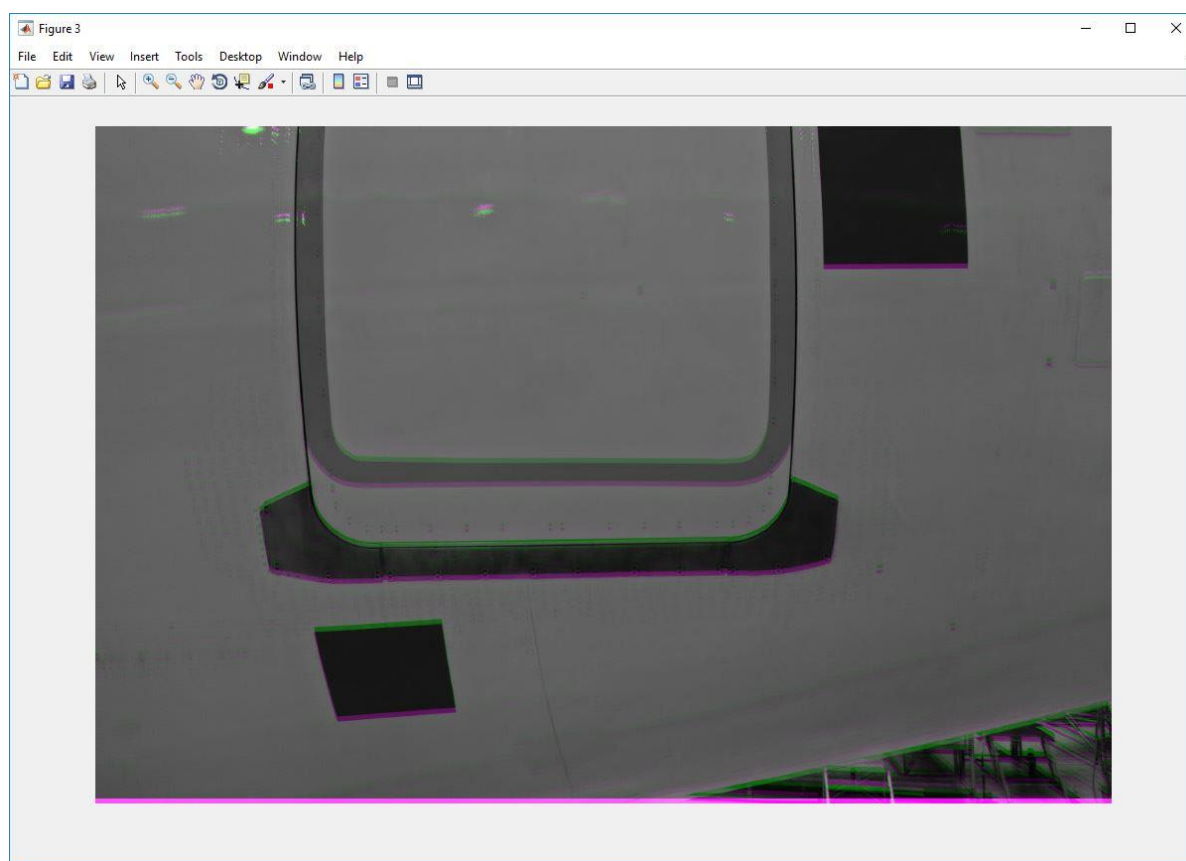
Slike u nastavku prikazuju različite kombinacije translacije fotografija. Detaljan opis pojmova dan je tablicom 5., prikazanom u ranije u tekstu.



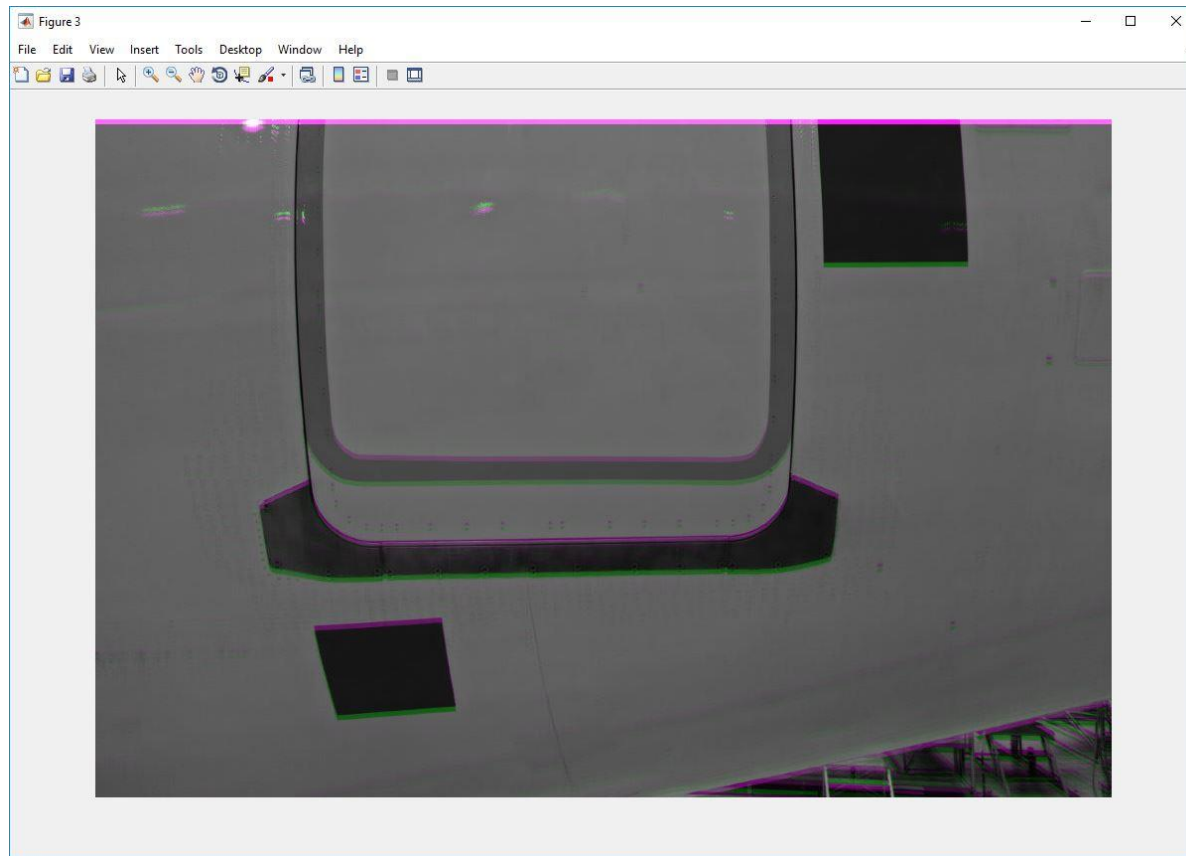
Slika 22. Usporedni prikaz početne i nove fotografije bez translacije

Prema slici 22. vidljivo je da ne postoji razlika u translaciji između početne i nove fotografije. Sva područja nove fotografije istovjetna su sa svim područjima stare fotografije jer nema promjene u boji na usporednom prikazu fotografija.

Slikom 23. prikazana je promjena kod koje je translacija nove fotografije izvedena prema gore. Na slici 23. pomak se očituje po ljubičastim i zelenim područjima. Ljubičasta područja predstavljaju pozicije specifičnih područja početne fotografije na usporednom prikazu, dok zelena područja predstavljaju pozicije specifičnih područja nove fotografije na usporednom prikazu.



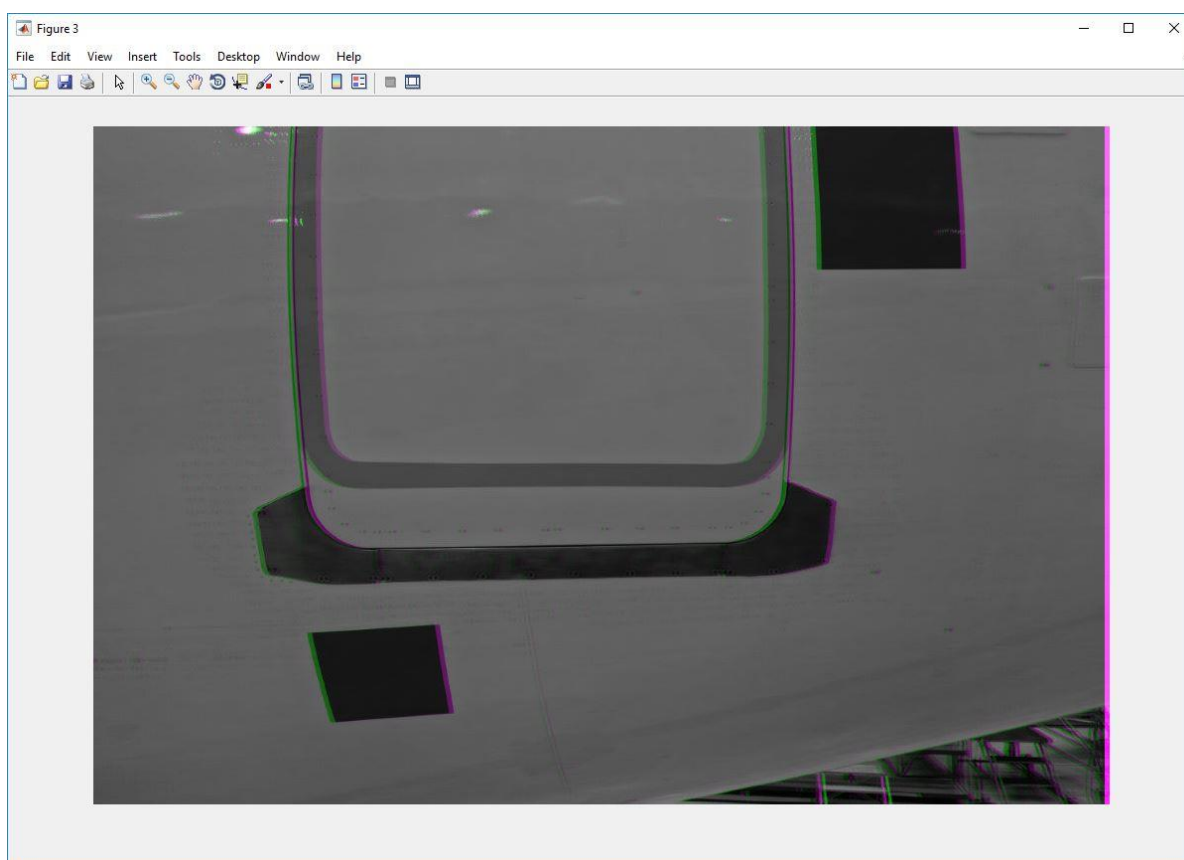
Slika 23. Usporedni prikaz početne i nove fotografije s translacijom prema gore



Slika 24. Usporedni prikaz početne i nove fotografije s translacijom prema dolje

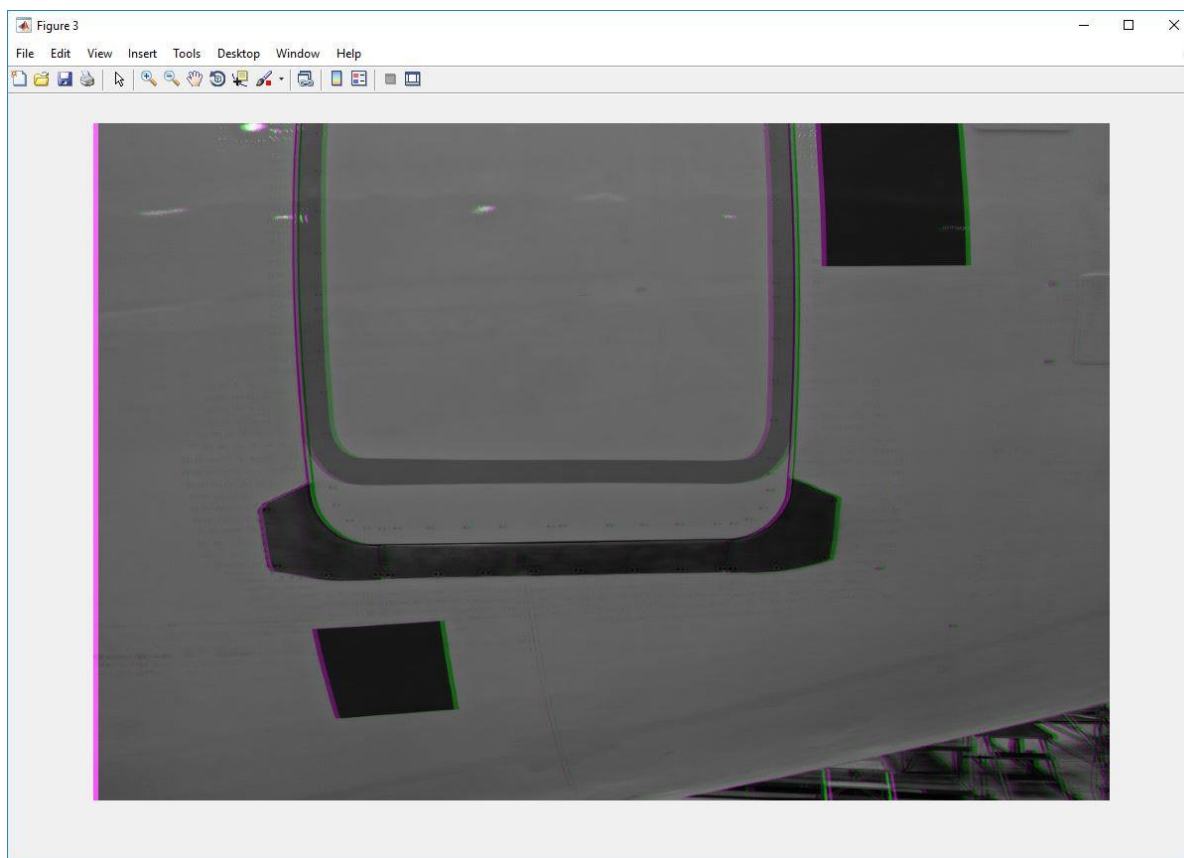
Slikom 24. prikazana je promjena uslijed translacije provedene prema dolje. Područja nove fotografije, koja se razlikuju od početne fotografije, predstavljena su zelenom bojom. Područja označena ljubičastom bojom predstavljaju ista navedena područja na početnoj fotografiji.

Prema slici 25. translacija je izvedena u lijevo što se očituje po ljubičastom području na desnoj strani usporednog prikaza fotografija. To područje govori da se tamo nalazi dio početne fotografije kojeg nema na novoj fotografiji.

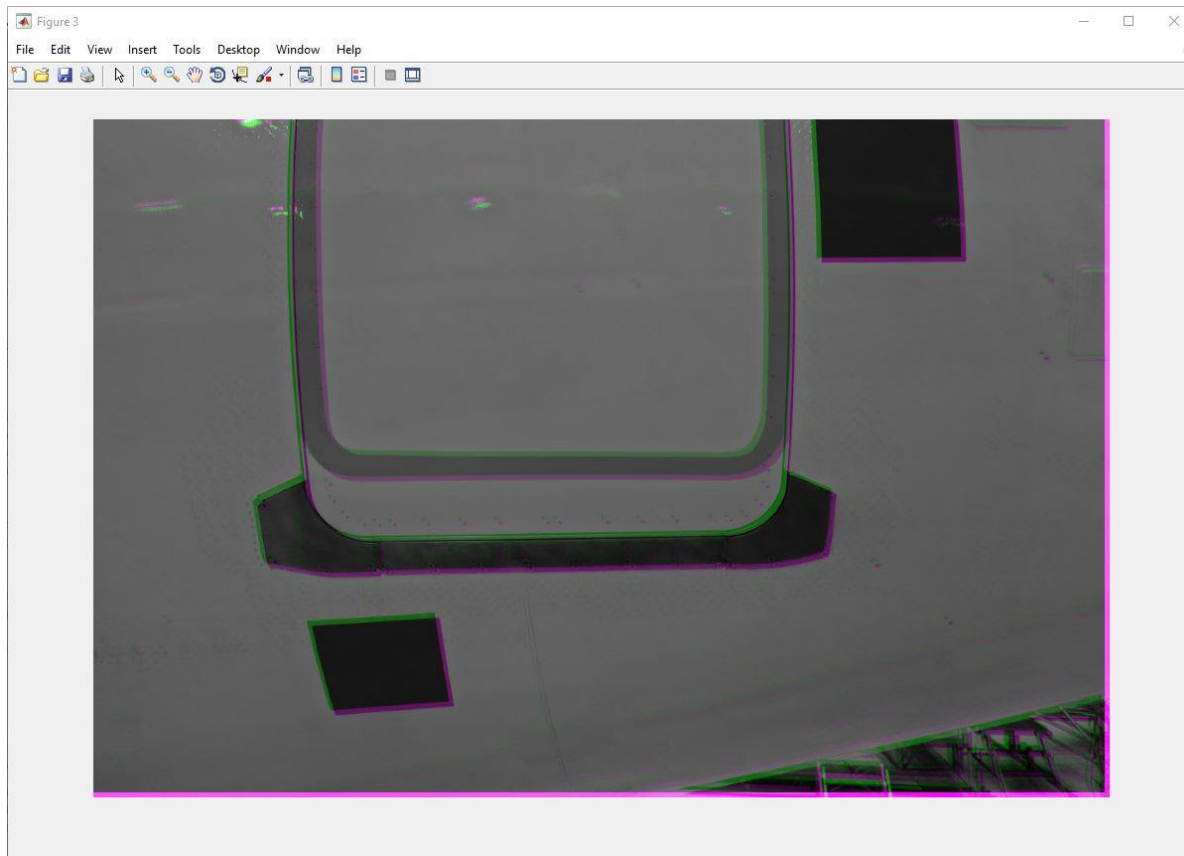


Slika 25. Usporedni prikaz početne i nove fotografije s translacijom prema lijevo

Kako je prikazano slikom 25. kada se translacija izvršila u lijevo, tako je slikom 26. prikazan usporedni prikaz početne i nove fotografije prilikom translacije prema desno što se opet najbolje očituje po lijevom rubu usporednog prikaza.



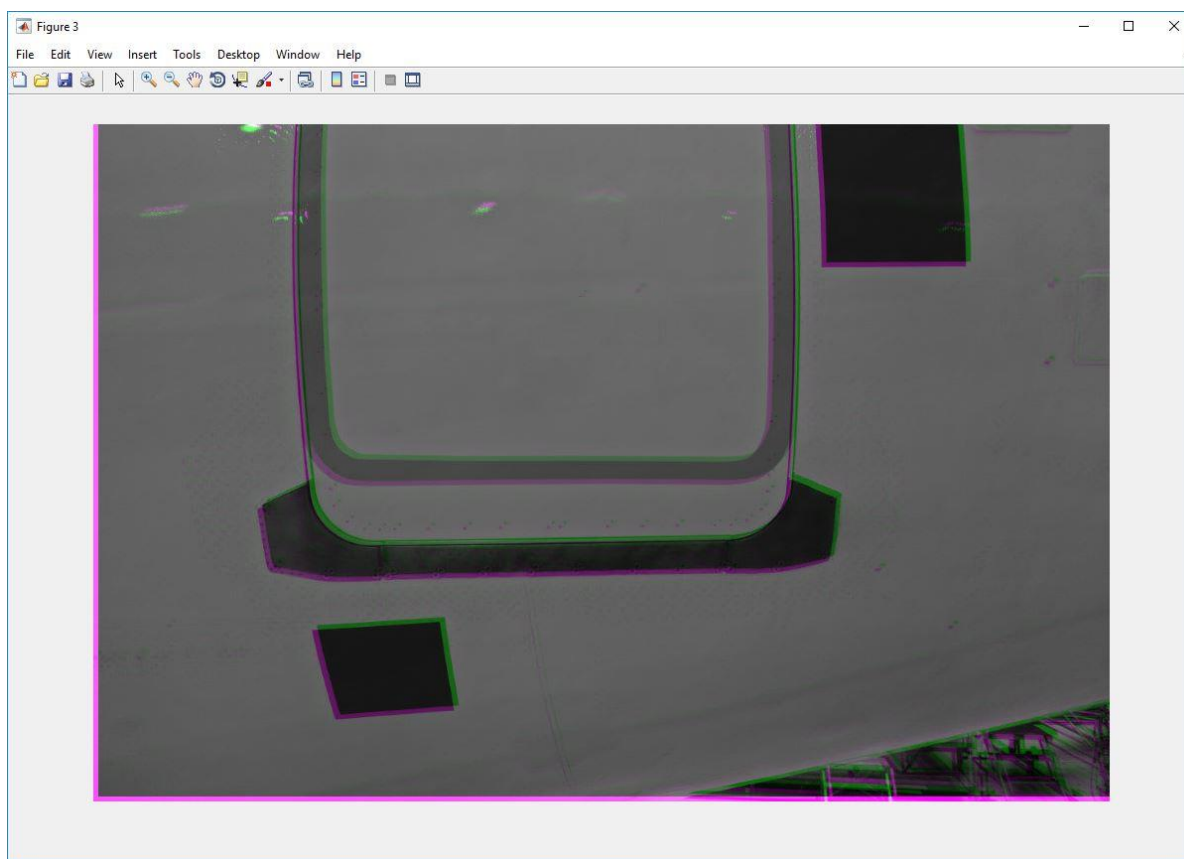
Slika 26. Usporedni prikaz početne i nove fotografije s translacijom prema desno



Slika 27. Usporedni prikaz početne i nove fotografije s translacijom prema gore- lijevo

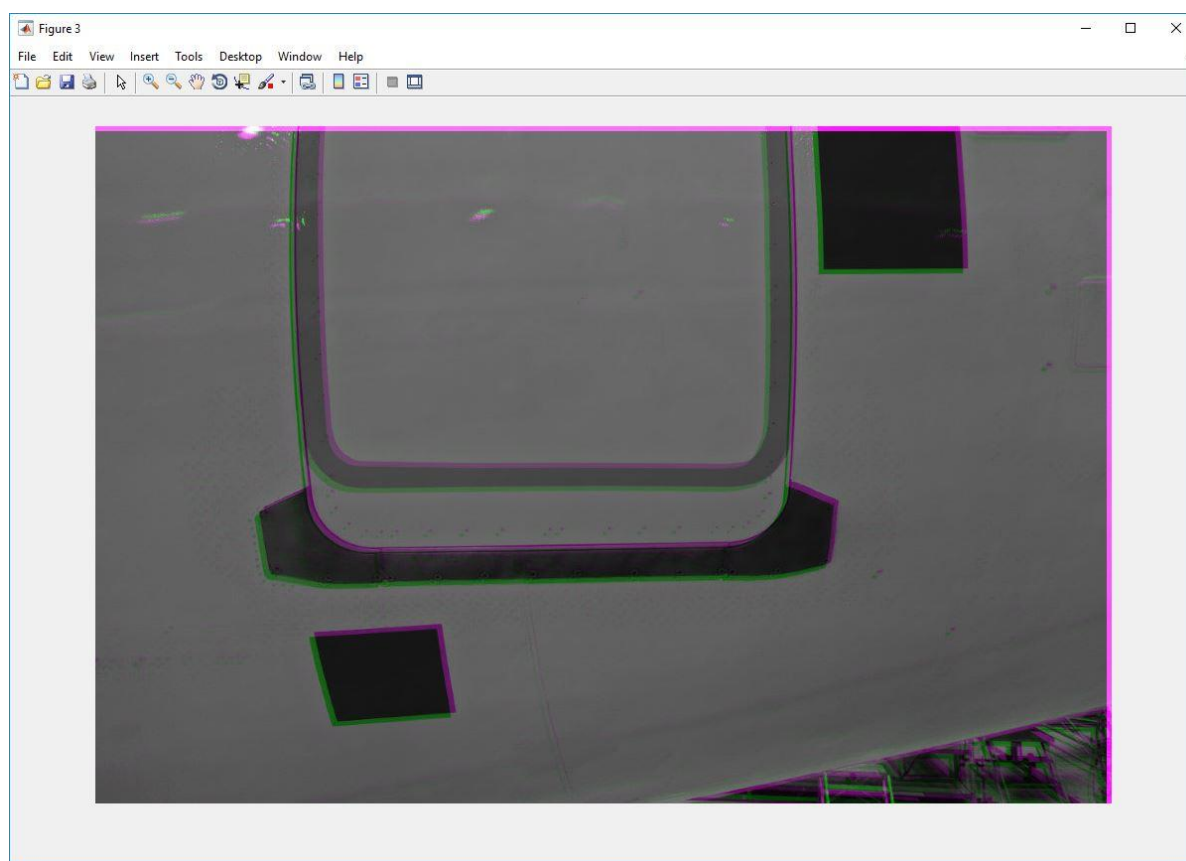
Pomak translacijom u dva smjera, gore i lijevo, prikazan je slikom 27. Na prikazanom usporednom prikazu, u odnosu na sliku 23., kada je translacija ostvarena samo prema gore i sliku 25., kada je translacija ostvarena samo u lijevo, vidljivo je kako je veći dio područja označen ljubičastom i zelenom bojom.

Slikom 28. prikazan je usporedni prikaz početne i nove fotografije s translacijom prema gore i u desnu stranu. Tako provedena translacija očituje se po ljubičastom rubu fotografije na lijevoj i donjoj stranici. Taj rub predstavlja poziciju početne fotografije različitu od nove fotografije.

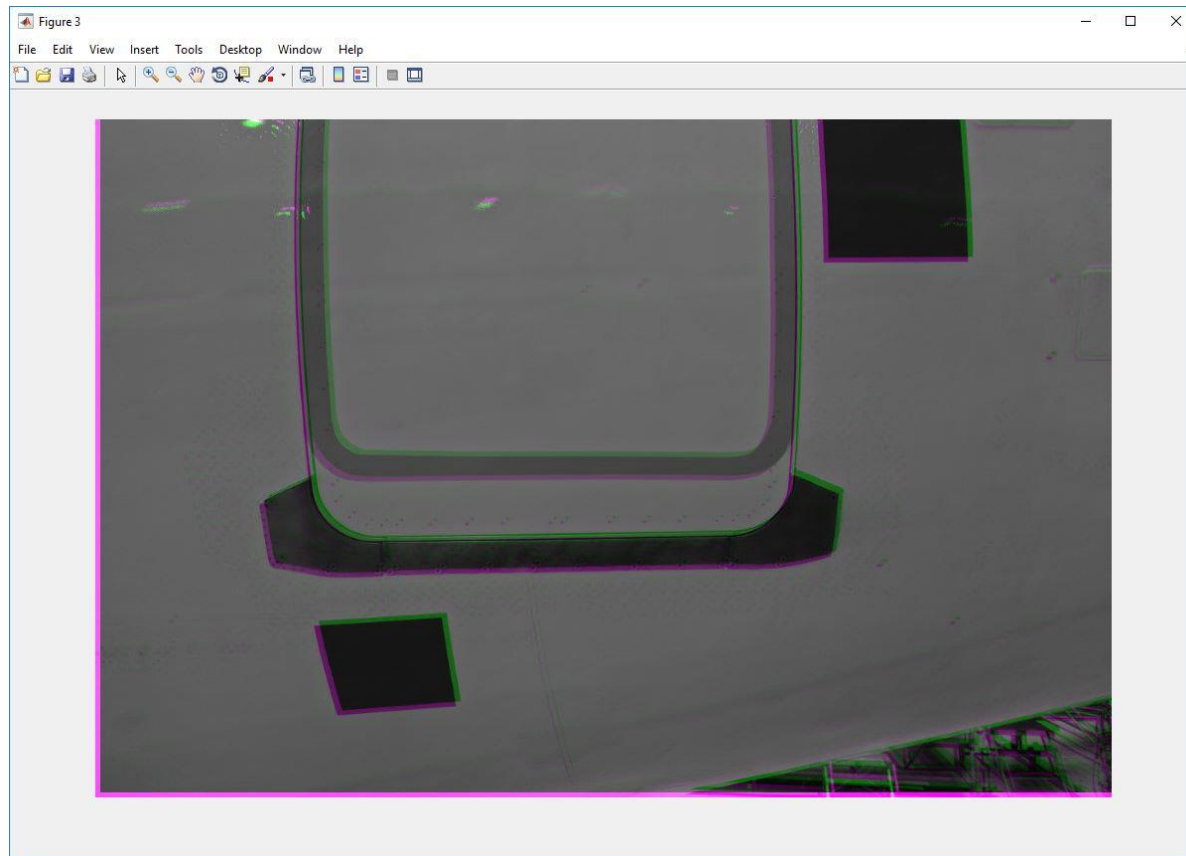


Slika 28. Usporedni prikaz početne i nove fotografije s translacijom prema gore-desno

Usporedni prikaz početne i nove fotografije izvedene s translacijom dolje-lijevo prikazan je slikom 29. Kao i na prethodnim fotografijama, područja označena ljubičastom bojom predstavljaju područja početne fotografije, dok područja označena zelenom bojom predstavljaju područja nove fotografije na usporednom prikazu.



Slika 29. Usporedni prikaz početne i nove fotografije s translacijom prema dolje-lijevo



Slika 30. Usporedni prikaz početne i nove fotografije s translacijom prema dolje-desno

Slikom 30. prikazana je zadnja moguća kombinacija translacija - u smjeru dolje-desno. Na taj način provjereni su rezultati pomaka za svaki od mogućih smjerova translacija.

5.5. Usklađivanje odmaka na fotografijama

Obzirom na način snimanja fotografija (bespilotna letjelica neće uvijek biti u istoj točki u prostoru), potrebno je provesti usklađivanje fotografija. Usklađivanje fotografija provodi se stvaranjem optimizatora i metričkih postavki modaliteta, podešavanjem svojstva operatora i zaključno usklađivanjem fotografija.

5.5.1. Stvaranje optimizatora i metrike

Varijable optimizator (`optimizer`) i metrika (`metric`) služe kako bi se pronašle sličnosti između početne i nove fotografije. Kao varijabla modaliteta (`modality`) zbog pretpostavke različite vrijednosti osvjetljenja i kontrasta na budućim fotografijama, odabrana je vrijednost `'multimodal'`.

```
[optimizer, metric] = imregconfig('multimodal');
```

5.5.2. Podešavanje svojstva optimizatora

Do vrijednosti prema kojima se kreće u podešavanje svojstva parametara došlo se eksperimentalno. Vrijednosti parametara koje daju optimalne rezultate korištene u ovom radu su:

- Odabir veličine radijusa pretraživanja

```
optimizer.InitialRadius = 0.0001;
```

- Vrijednost parametra `epsilon` kojim je određena minimalna veličina radijusa pretraživanja iznosi:

```
optimizer.Epsilon = 1.5e-5;
```

- Vrijednost parametra `growthFactor` koji predstavlja faktor rasta radijusa pretraživanja iznosi:

```
optimizer.GrowthFactor = 1.01;
```

- Maksimalan broj iteracija koji je dozvoljen u ovom primjeru iznosi:

```
optimizer.MaximumIterations = 300;
```

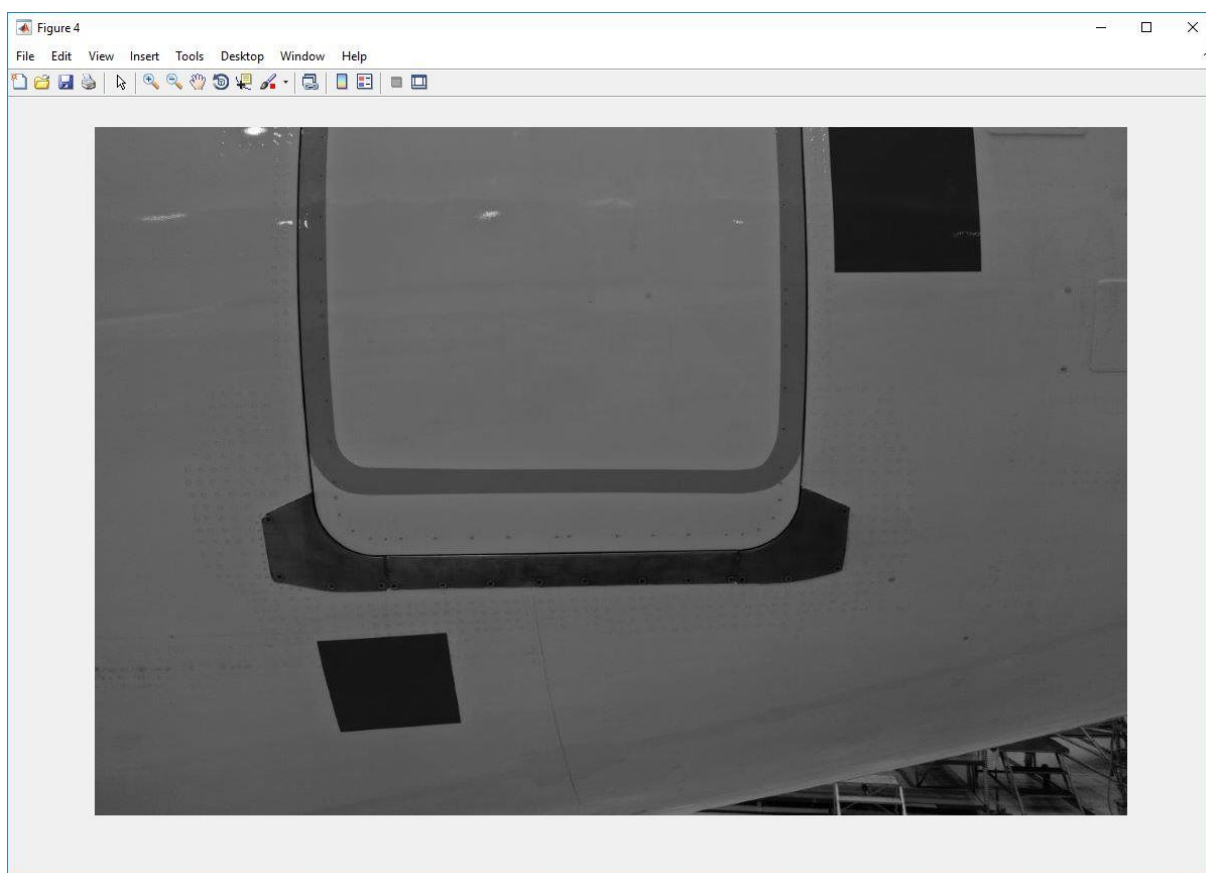
5.5.3. Korištenje dobivenih vrijednosti optimizatora i metrike radi usklađivanja fotografija

Dobivene vrijednosti optimizatora (`optimizer`) i metrike (`metric`) koriste se kako bi se nova slika (`novaCB`) uskladila s početnom (`pocetnaCB`). Korišteni tip geometrijske transformacije (`transformType`) je `'affine'` koji omogućuje ne samo promjenu u translaciji, nego i u rotaciji, promjeni mjerila i smicanju.

```
novaUskladena =  
imregister(novaCB,pocetnaCB,'affine',optimizer,metric);
```

Rezultat usklađivanja nove i početne fotografije može se vidjeti na slici 31. u nastavku koja se dobiva pozivanjem sljedeće naredbe:

```
figure  
imshowpair(pocetnaCB, novaUskladena,'blend')
```



Slika 31. Rezultat nakon provedenog usklađivanja

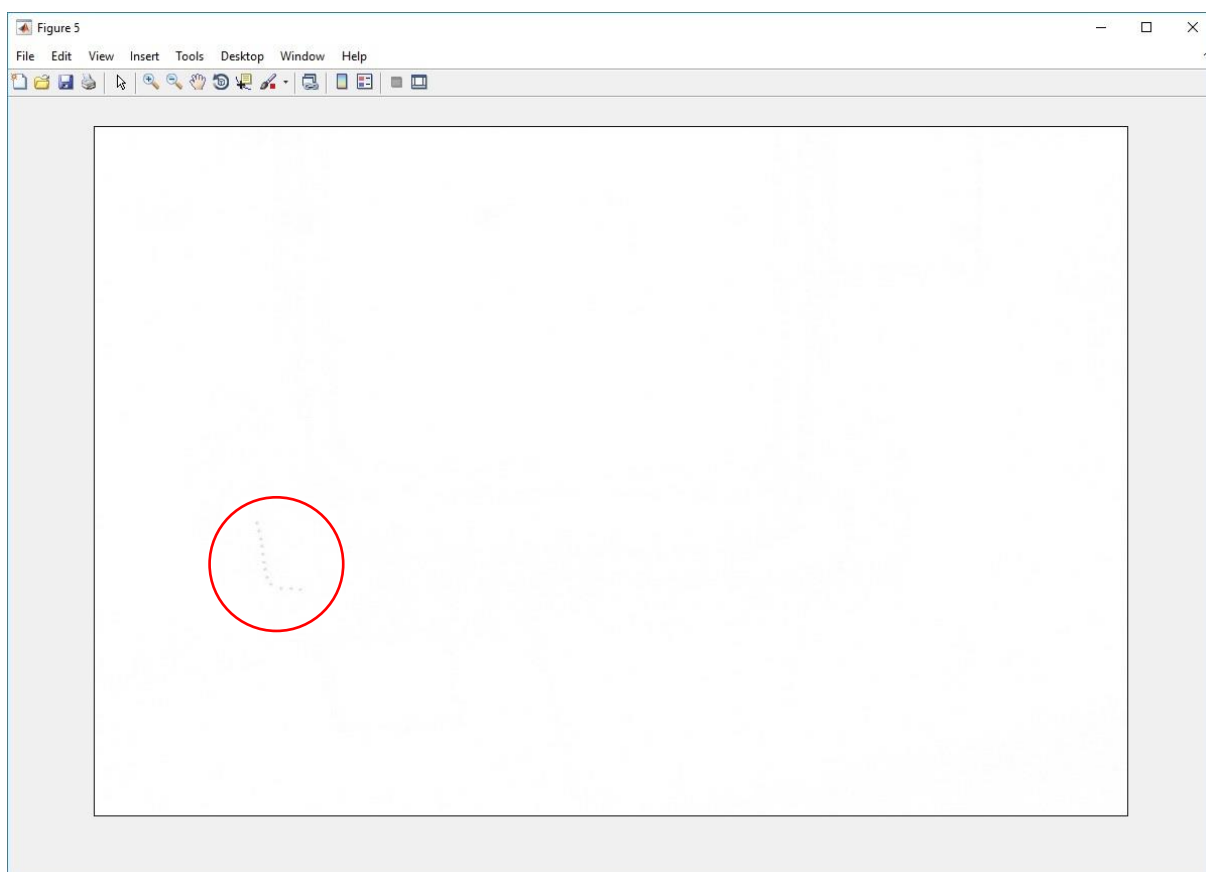
5.6. Pronalazak promjena između fotografija

Kako bi se našle promjene između fotografija potrebno je provesti usporedbu između svakog elementa polja nove (usklađene) fotografije (*novaUskladena*) i početne fotografije (*pocetnaCB*).

```
promjena = imabsdiff(pocetnaCB,novaUskladena);
```

Kako bi rezultat *promjena* bio vidljiviji, na slici 32. napravljen je inverz dobivenog rezultata. Prikaz rezultata pronalaska promjena dobiva se pozivanjem sljedeće sintakse:

```
figure  
imshow(promjena)
```



Slika 32. Promjene nastale kao rezultat usporedbe fotografija

5.7. Pronalaženje najveće vrijednosti promjene između fotografija

Najveća vrijednost promjene može se iščitati iz matrice koja je nastala kao rezultat usporedbe fotografija (promjena). Kako bi se ta vrijednost dohvatila potrebno je pozvati funkciju `max`.

```
maxPromjena = max(max(promjena));
```

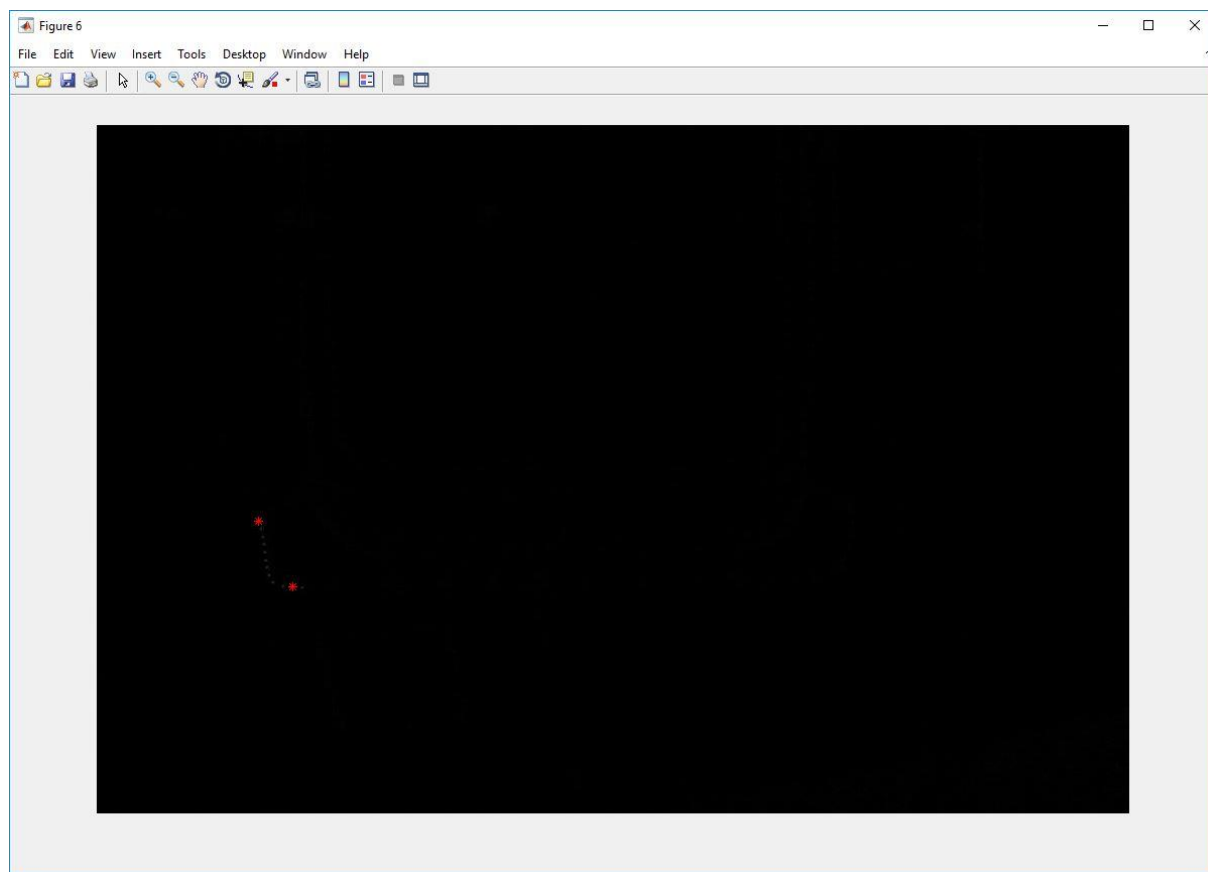
Funkcijom `find` izvlače se koordinate točke koja je poprimila najveću vrijednost promjene.

```
[iRow, iCol] = find(promjena == maxPromjena);
```

Kako bi se omogućio prikaz slike s pronađenim promjenama i označavanjem točaka s najvećim promjenama koristi se sljedeća sintagma.

```
figure
imshow(promjena)
hold on
plot(iCol, iRow, 'r*')
hold off
```

Slika 33. prikazuje točke s najvećom vrijednošću promjene.

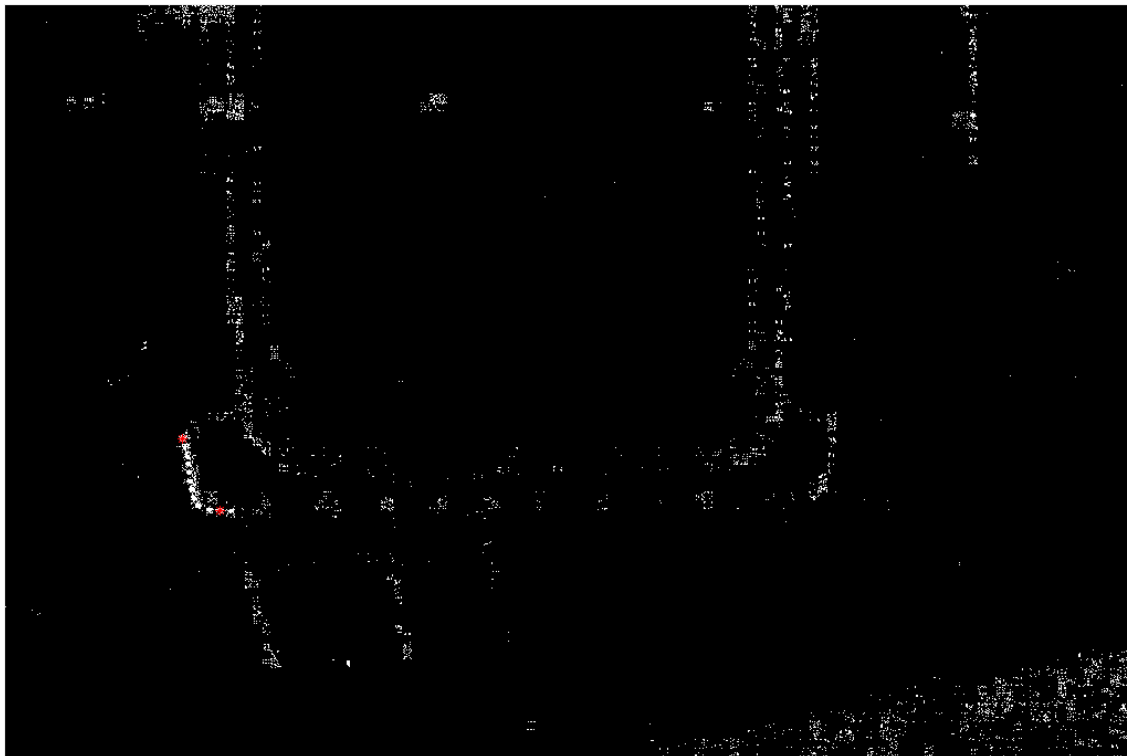


Slika 33. Točke s najvećom vrijednošću promjene

5.8. Određivanje praga osjetljivosti

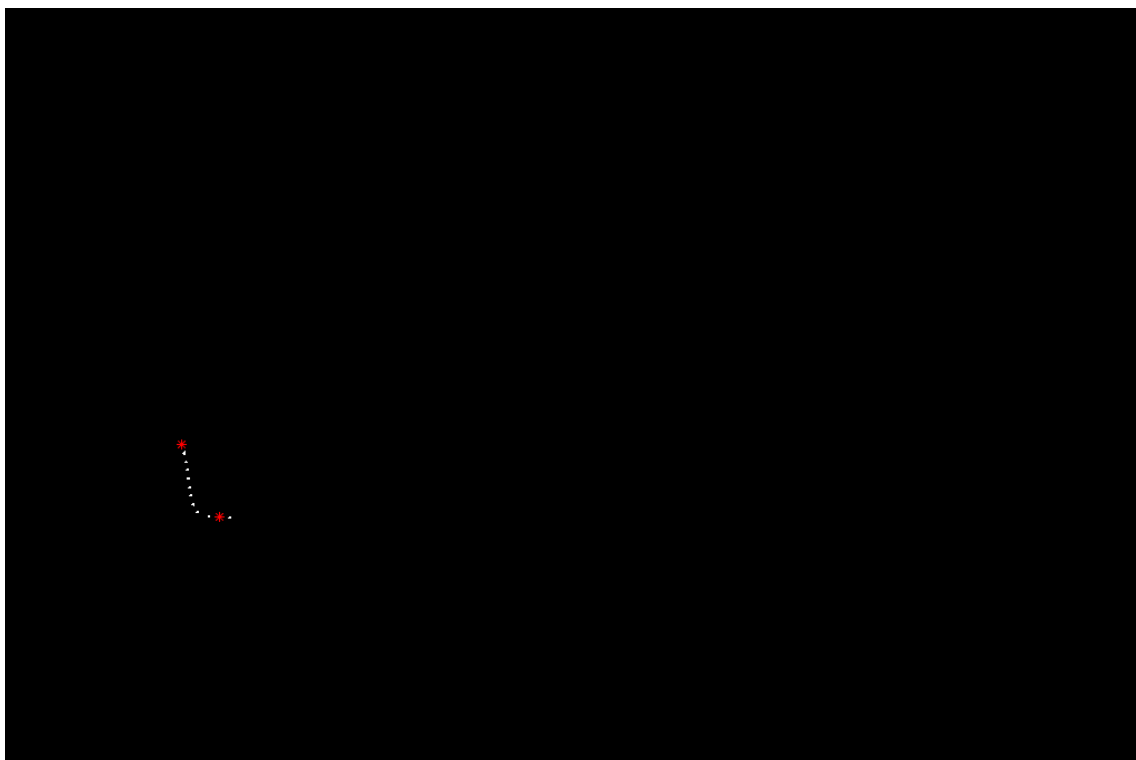
Prag osjetljivosti određuje se temeljem vrijednosti promjene koje su dobivene usporedbom nove (usklađene) fotografije i početne fotografije. Potreba za određivanjem praga osjetljivosti nastala je kako bi se iz matrice promjena uklonile vrijednosti koje su manje od vrijednosti određene pragom osjetljivosti. Određivanje praga osjetljivosti važno je provesti radi postojanja šuma na fotografijama. Kako bi se odredila optimalna razina praga osjetljivosti, koja daje najbolje rezultate za daljnju analizu, provedena je analiza prikazana slikama u nastavku.

Slikom 34. prikazana je vrijednost praga osjetljivosti za slučaj kada se uzimaju sve promjene na fotografijama veće od 2. Kako je prikazano slikom 34., broj takvih područja je jako velik te se smatra da je šum, koji pritom nastaje, nepovoljan za daljnju obradu i usporedbu fotografija.



Slika 34. Prag osjetljivosti za područja veća od 2

Prema slici 35. vrijednost praga iznosi 20. U tom slučaju napravljene pogreške su otkrivene, ali veličina područja je jako mala. Daljnjom analizom, navedena područja se prikazuju kao šum na fotografiji, a rezultat je algoritam koji prikazuje da nije došlo do promjena.



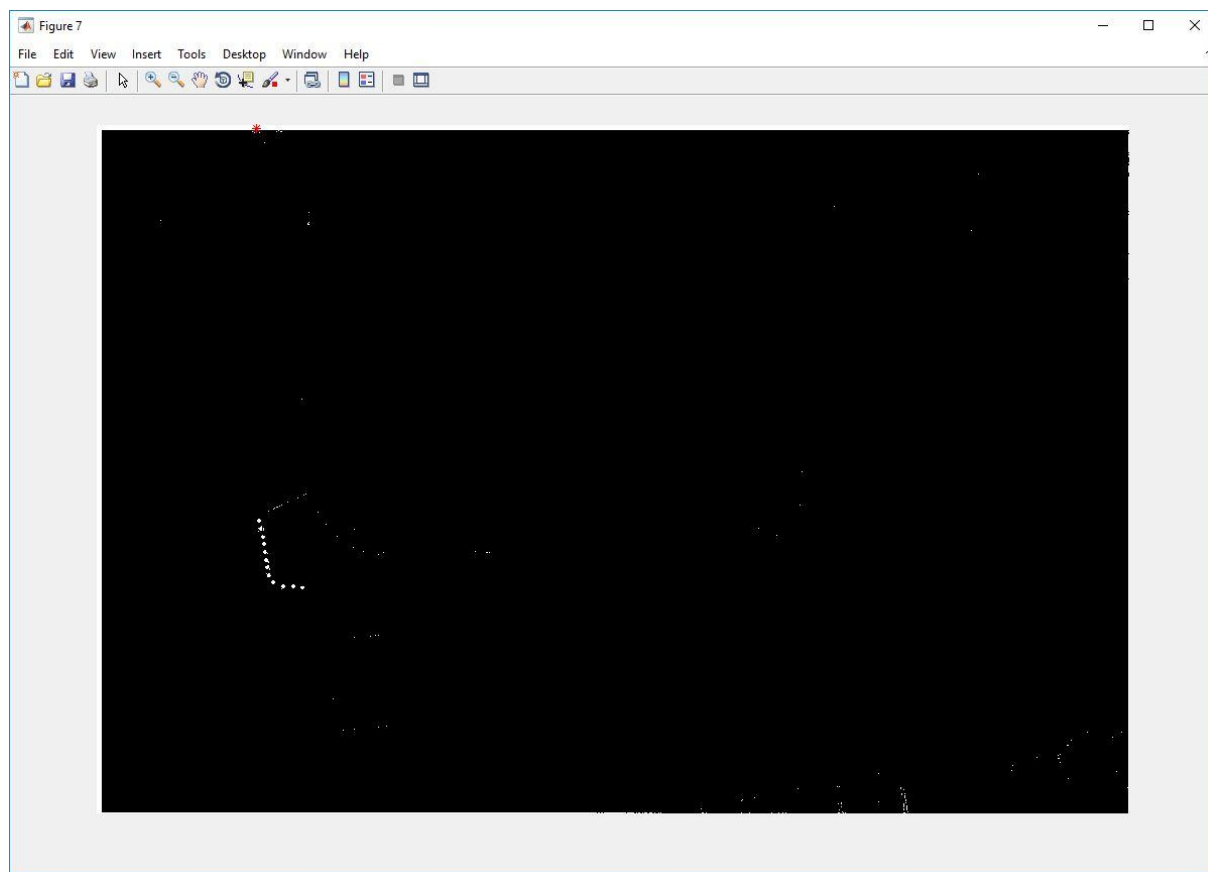
Slika 35. Prag osjetljivosti za područja veća od 20

Kako bi došlo do pojačavanja kontrasta područja, u navedenom primjeru vrijednost praga osjetljivosti, koja daje najbolje rezultate iznosi 8 što je prikazano sintaksom u nastavku:

```
slikaPraga = promjena > 8;
```

Prikaz fotografije prikazane slikom 36., nakon određivanja praga osjetljivosti ostvaruje se sintaksom:

```
figure  
imshow(slikaPraga)  
hold on  
plot(iCol,iRow,'r*')  
hold off
```

Slika 36. Prikaz fotografije nakon određivanja praga osjetljivosti

Na slici 36. korišten je prikaz između nove pomaknute gore-lijevo kako bi se bolje prikazao utjecaj šuma na sliku.

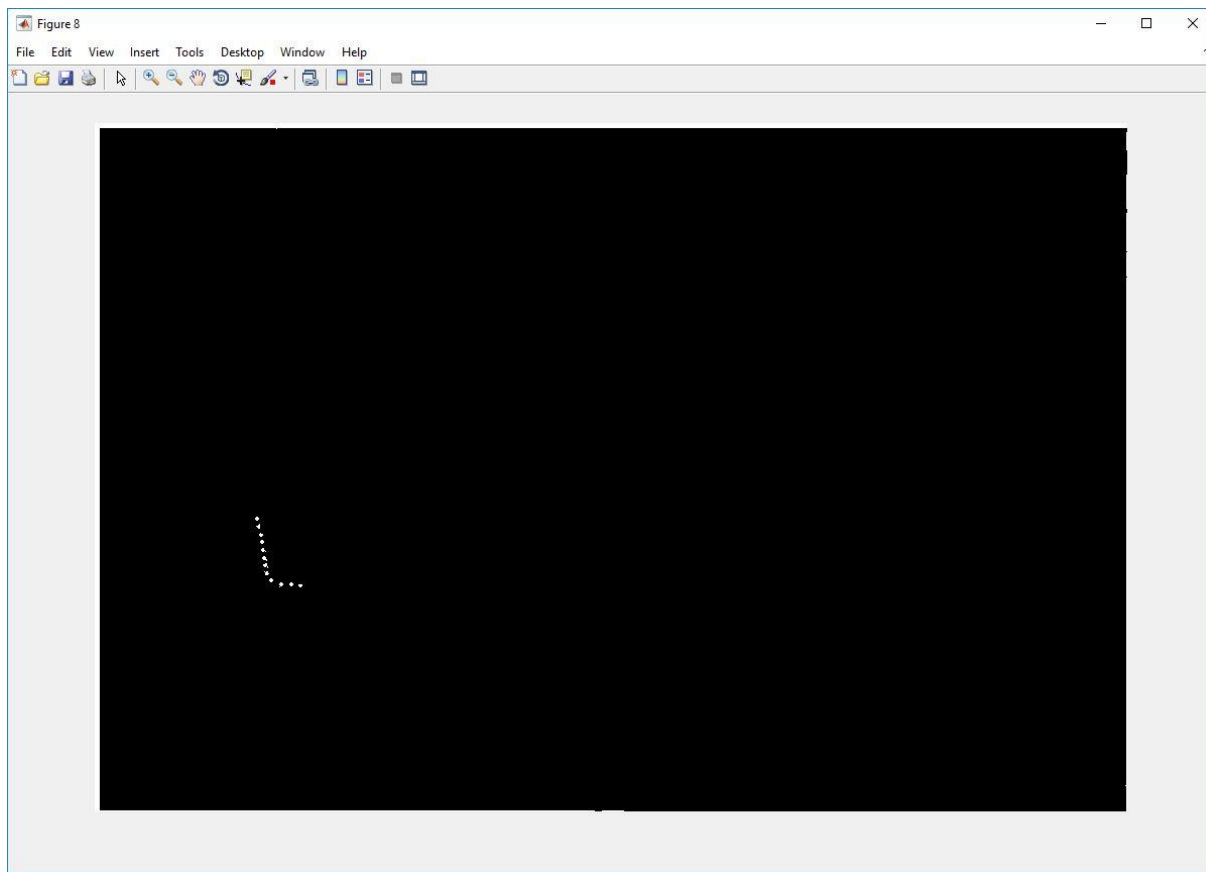
5.9. Čišćenje šumova slike

Prilikom određivanja slike praga neke vrijednosti matrice (veće od praga), rade šumove koji su prikazani na slici 35. Kako bi se ti šumovi uklonili, koristi se funkcija `bwareaopen` kojom dr uklanjaju svi objekti površinom manji od 8 piksela.

```
cistaSlika = bwareaopen(slikaPraga, 8);
```

Kao i u prethodnom odlomku, za prikaz očišćene slike, prikazane slikom 37, korištena je nova slika koja je pomaknuta gore-lijevo. Sintaksa za dobivanje traženog izlaza dana je u nastavku:

```
figure  
imshow(cistaSlika)
```



Slika 37. Prikaz slike bez utjecaja šuma

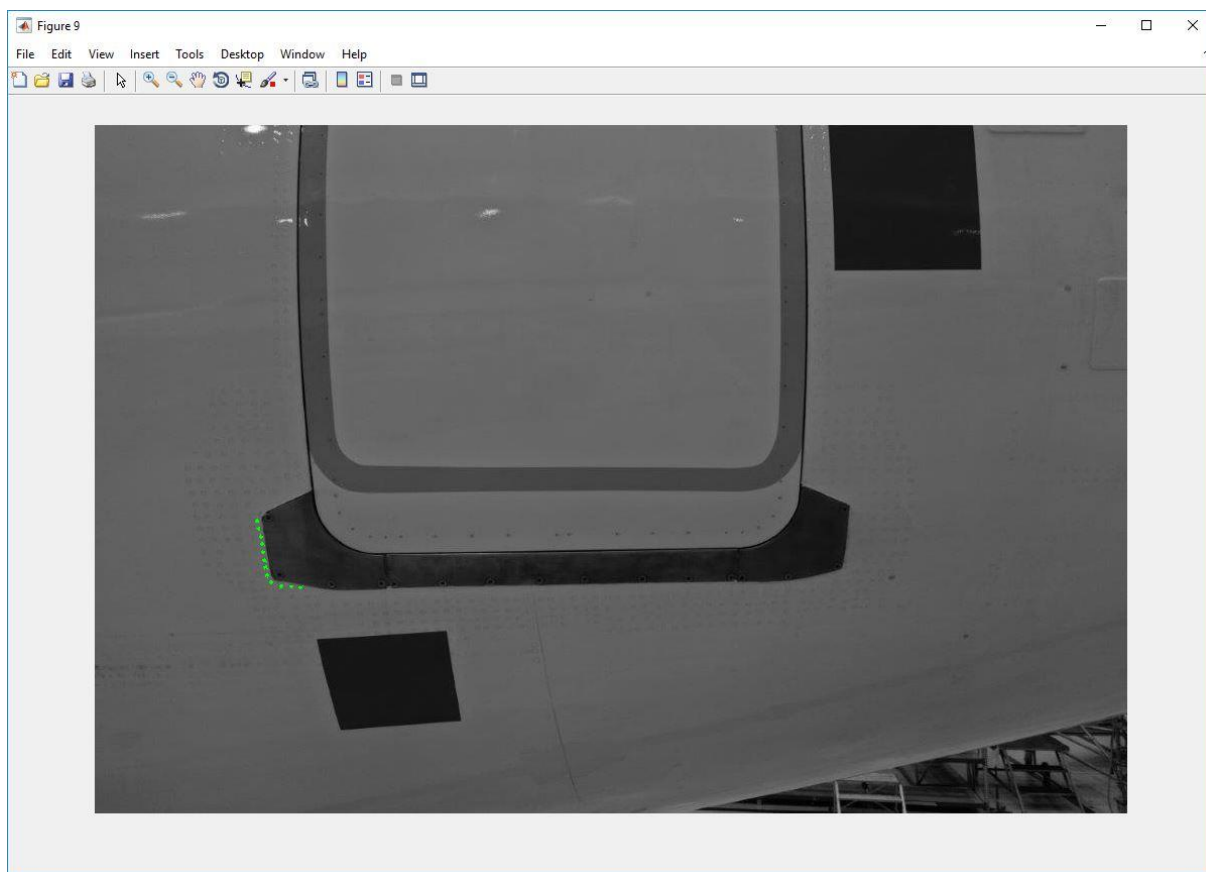
5.10. Stvaranje maske s pronadenim promjenama

Prikaz uočenih promjena na slici može se ostvariti na način da se preko „originalne“ fotografije prelijepi maska s promjenama. Funkcijom `imoverlay` se uspoređuje nova, usklađena slika (`novaUskladena`) sa slikom koja sadrži podatke o promjenama (`cistaSlika`) na kojoj se zelenom bojom označavaju promjene.

```
preklop = imoverlay(novaUskladena,cistaSlika,'g')
```

Slika 38. prikazuje područja na novoj slici na kojima je uočena promjena u odnosu na početnu fotografiju. Prikaz slike ostvaren je sintagmom u nastavku.:

```
Figure  
imshow(preklop)
```



Slika 38. Prikaz područja različita (zelena boja) od početne fotografije

5.11. Pozivanje funkcije za prebrojavanje pronađenih razlika

Funkcija za prebrojavanje pronađenih razlika poziva se sintaksom u nastavku:

```
properties = filterRegions(cistaSlika);
```

Kao ulaznu varijablu funkcija `filterRegions` prima sliku koja je obrađena funkcijom `praga` i očišćena od šumova (`cistaSlika`).

5.11.1. Opis funkcije `filterRegions`

Funkcija `filterRegions` napisana je u novoj skripti `filterRegions.m`. Funkcija je deklarirana sljedećom sintaksom:

```
function properties = filterRegions(cistaSlika_in);
```

`properties` je naziv varijable funkcije, dok `cistaSlika_in` predstavlja ulaznu varijablu funkcije.

Deklariranje naziva izlazne varijable funkcije obavlja se sintaksom kojom varijabla `cistaSlika_out` poprima sve vrijednosti varijable `cistaSlika_in`.

```
cistaSlika_out = cistaSlika_in;
```

Nakon usklađivanja fotografija na način opisan u poglavlju 5.5. javljaju se promjene na rubovima koje utječu na prikaz konačnog rezultata promjena. Kako bi se uklonile takve promjene potrebno je isključiti rubove slike iz daljnjeg promatranja, što je moguće učiniti sljedećom sintaksom:

```
cistaSlika_out = imclearborder(cistaSlika_out);
```

Varijabli `cistaSlika_out` pridružuju se vrijednosti promjena površinom veće od 9 piksela, kojima gornja granica površine teži u beskonačnost.

```
cistaSlika_out = bwpropfilt(cistaSlika_out, 'Area', [9 +  
eps(100), Inf]);
```

Varijablom `properties` dohvaćaju se sva svojstva vezana uz sliku koja se obrađuje funkcijom `filterRegions`.

```
properties = regionprops(cistaSlika_out, {'Area',  
'BoundingBox', 'Eccentricity', 'EquivDiameter', 'EulerNumber',  
'MajorAxisLength', 'MinorAxisLength', 'Orientation',  
'Perimeter'});
```

Zatim se prepisuje varijabla `properties` kako bi poprimila samo svojstva površine potrebne za daljnju analizu.

```
properties = sortProperties(properties, 'Area');
```

Stvara se `if` petlja koja provjerava promjene na novoj fotografiji u odnosu na postojeću. Ukoliko promjena nema, uvjet je zadovoljen i nastavlja se izvršavanje. U slučaju kada ima promjena stvara se strukturni niz sa svojstvima promjena.

```
if isempty(properties)
else
    properties = struct2table(properties);
end
```

5.11.2. Opis funkcije `sortProperties`

Druga funkcija koju je potrebno napraviti na temelju stvorenog strukturnog niza vraća koordinate područja na kojima je došlo do promjene i područja na fotografiji koja zadovoljavaju tražena svojstva veličine površine. Pozivanje funkcije vrši se sljedećom sintaksom:

```
function properties = sortProperties(properties, sortField);
```

Sortiranje promjena izvodi se u silaznom nizu.

```
[~,idx] = sort([property.(sortField)], 'descend');
```

Za kraj funkcije potrebno je promijeniti redoslijed cijele strukture, što se izvodi sljedećom sintaksom:

```
properties = properties (idx)
```

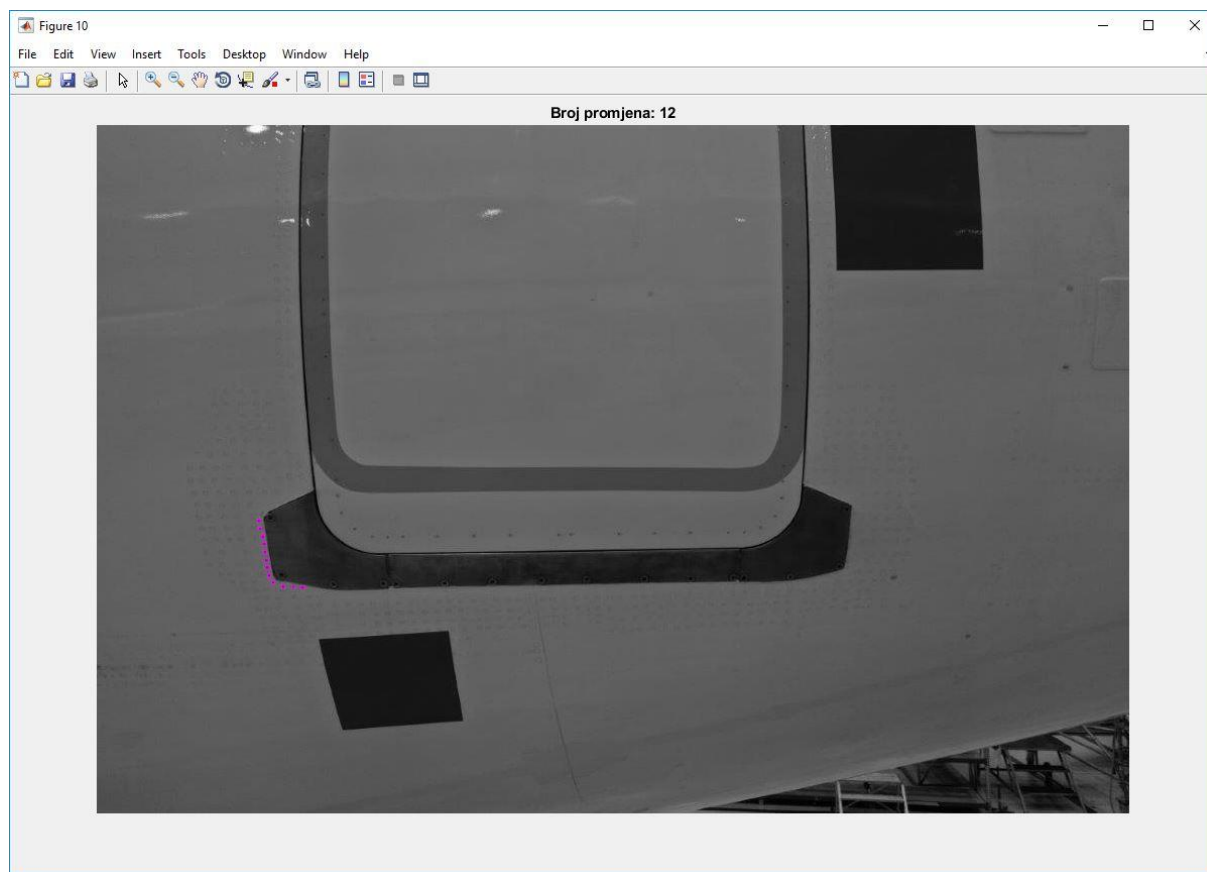
5.12. Konačni prikaz rezultata

Nakon pozivanja funkcija, stvara se `if` petlja koja provjerava postoje li promjene između početne i nove fotografije. Ukoliko nema promjena izvršava se dio koda pod `if` dijelom, a ukoliko ima promjena u odnosu na početnu fotografiju izvršava se `else` dio. Varijabla `total_count` sadrži informaciju o broju redaka matrice u varijabli `properties`. Prikaz `if` dijela koda, gdje na novoj slici nema vidljivih promjena u odnosu na postojeću, prikazan je slikom 39. Prikaz `else` dijela koda prikazan je Slikom 40. na kojoj su vidljive promjene i broj promjena koji odgovara traženim parametrima.

```
if isempty(properties)
    figure
    imshow(novaUskladena);
    title("Nema vidljivih promjena");
else
    total_count = height(properties);
    rgb = insertShape(novaUskladena, 'rectangle',
        [properties.BoundingBox], 'color', 'magenta');
    figure
    imshow(rgb);
    title("Broj promjena: " + total_count);
end
```

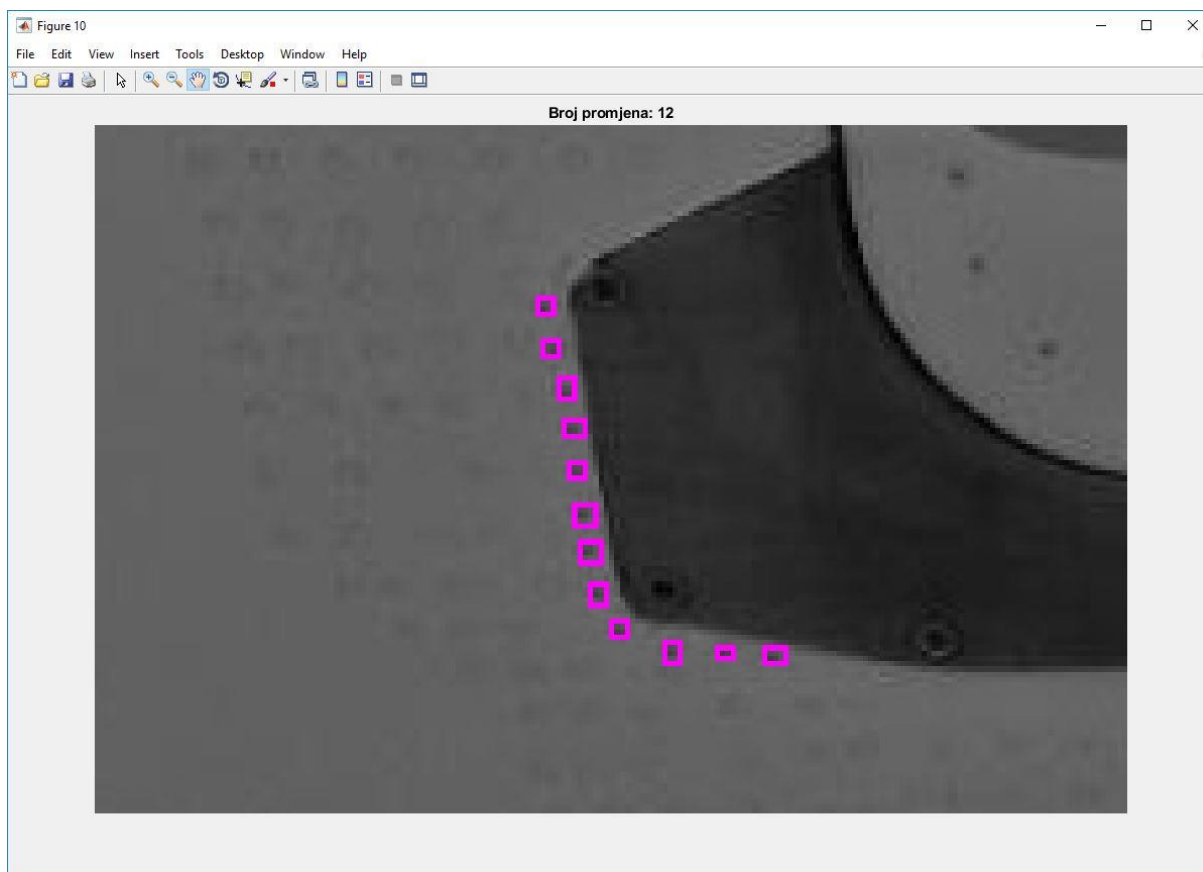


Slika 39. Prikaz nove fotografije koja nema vidljivih promjena u odnosu na postojeću



Slika 40. Prikaz nove fotografije s označenim promjenama i brojem promjena u odnosu na početnu fotografiju

Prilikom prikaza fotografije, istu je moguće povećati kako bi se dobio detaljniji uvid u promjene između postojeće i nove fotografije. Slikom 41. prikazano je uvećano područje pronađenih razlika nakon što je provedeno usklađivanje fotografija pomoću alata koji su sastavni dio MATLAB-ovog alata za obradu slika (engl. *Image Processing Toolbox*). Usporedbom slike 21. i slike 40. i 41. može se primijetiti, da je danim načinom rješavanja, program prepoznao svih 12 pogrešaka.



Slika 41. Uvećan prikaz pronađenih promjena na novoj slici

Nakon dobivenih rezultata prikazanih slikama 40. i 41. osoba zadužena za održavanje na temelju pregleda fotografija s pronađenim razlikama može donijeti početnu odluku o tome treba li se pristupiti procesu održavanja ili je registrirana pogreška nastala kao rezultat šuma.

Ovim alatom ne osigurava se stopostotna sigurnost u otkrivanju pogrešaka te je potrebno provesti vizualni pregled kako bi se uvjerilo da pogreška ne postoji. Korištenjem ovog alata održavatelji mogu dobiti brzi uvid u područja na trupu zrakoplova na kojima bi mogla nastati promjena u odnosu na prethodni pregled.

5.12.1. Točnost prepoznavanja anomalija površine

Točnost prepoznavanja anomalija površine ovisi o sljedećim koracima:

1. Kvaliteta fotografije koja je snimljena
2. Usklađivanje odnaka na fotografijama
3. Određivanje praga osjetljivosti
4. Čišćenje šumova

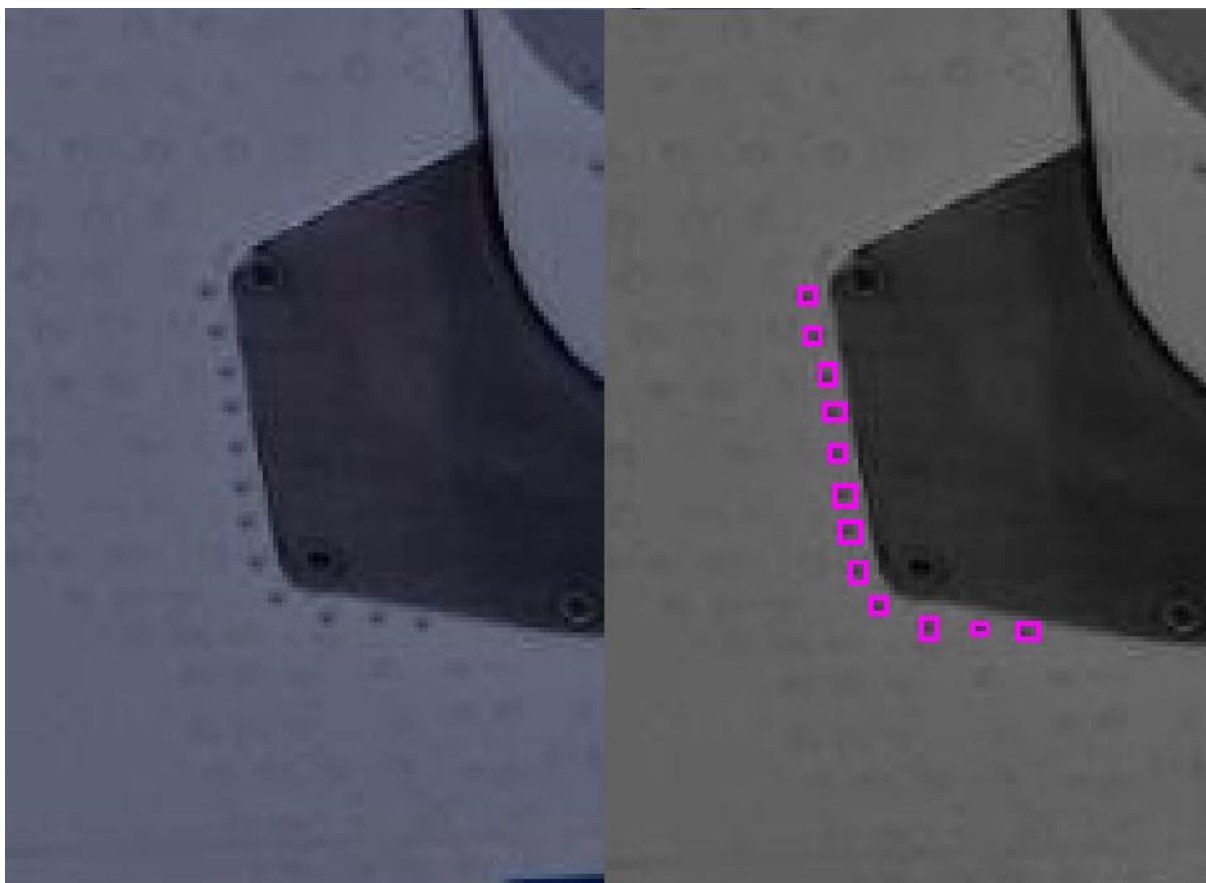
Na kvalitetu fotografije mogu utjecati parametri kao što su promjena razine osvjetljenja u hangaru ili prljavština na trupu zrakoplova. Može se pretpostaviti da će se fotografiranje zrakoplova obavljati uvijek u istom, kontroliranom okruženju. Iz toga proizlazi da će kvaliteta fotografija uvijek biti jednaka.

Drugi korak – usklađivanje odmakata utječe na točnost prepoznavanja anomalija zbog nesavršenosti algoritma u rješavanju većih problema nastalih uslijed rotacije fotografije. Mali pomaci u rotaciji su uspješno riješeni, ali uz daljnji razvoj algoritma dolazi do povećanja točnosti. Drugi način rješavanja problema usklađivanja može biti postavljanje fiksnih markera na trupu zrakoplova što bi algoritmu olakšalo pronalaženje ključnih točaka na fotografijama. Brzina kojom se prepoznaju anomalije u direktnoj je vezi s korakom u kojem se provodi usklađivanje. U slučaju ako se uz podešavanje parametara odabere veća brzina, fotografije možda neće biti potpuno usklađene što će dovesti do pojave većeg broja šumova koje će algoritam prepoznati kao anomalije, koje zapravo to nisu.

Određivanje praga osjetljivosti provodi se kako bi se pojačao kontrast između nove površine, s anomalijom, od početne s kojom se nova fotografija uspoređuje. Postavljanjem malog praga osjetljivosti, količina šuma će biti izrazito velika, a u suprotnom, postavljanjem većeg praga osjetljivosti, određene anomalije će postati zanemarive za algoritam.

Čišćenje šumova na fotografijama u direktnoj je vezi s korakom određivanja praga osjetljivosti. U slučaju kada postoji velik broj šumova na fotografiji, šumovi se grupiraju u veće površine te se odabirom velike površine čišćenja uklanjaju sve anomalije. U suprotnom slučaju, odabirom nedovoljno velike površine, broj pronađenih anomalija bit će jako velik dok u stvarnosti sve pronađene anomalije nisu rezultat pogreške već nekog vanjskog utjecaja.

Prema svemu navedenom, točnost algoritma nije moguće provjeriti na jednostavan način, već je potrebno provesti puno podešavanja parametara i provjera kako bi se dobio rezultat koji daje zadovoljavajuće rezultate. Na slici 42., na lijevoj strani, crne točke predstavljaju anomalije, dok su te iste anomalije prebrojane i označene na desnom dijelu slike. Ranije opisanom postavkom parametara za prikazani slučaj algoritam daje sto postotne rezultate točnosti. Takav rezultat je teško očekivati u praksi.



Slika 42. Usporedni prikaz nove fotografije s prikazanim razlikama i pronađenih razlika uz pomoć algoritma

ZAKLJUČAK

U današnjem svijetu, kada se teži automatizaciji i samostalnosti nastoji se automatizirati poslove vezane uz održavanje. Prikupljanje i obrada velike količine podataka pomoću prediktivnog održavanja omogućuju predikciju kvarova s velikom pouzdanošću. Primjena prediktivnog održavanja dovodi do smanjenja troškova, povećanja pouzdanosti i sigurnosti te pravovremenog otklanjanja mogućih uzroka kvarova. Računalni vid pomaže pri dijagnostici kvara korištenjem kamera i senzora. Pomoću alata za obradu fotografije programskog paketa MATLAB dolazi do pronalaska promjena na fotografijama nastalih zbog oštećenja trupa zrakoplova.

Unutar diplomskog rada prikazane su mogućnosti MATLAB-ovog alata za obradu fotografija. Korištenjem navedenog alata, na primjeru je izvršeno usklađivanje fotografija i pronalaženje pogrešaka među fotografijama. U radu je provedeno istraživanje mogućnosti pronalaženja pogrešaka na trupu zrakoplova u svrhu provođenja prediktivnog održavanja. Na dobivenim fotografijama, prikupljenim bespilotnom letjelicom, provedena je obrada kojom su prouzrokovane pogreške te je testiran rad algoritma kao simulacija uvjeta koji se mogu dogoditi u stvarnosti. Zbog odmakata nastalog nemogućnošću točnog pozicioniranja bespilotne letjelice u prostoru, na dobivenim fotografijama potrebno je izvesti usklađivanje. Usklađivanje se provodi optimizacijom parametara što utječe na točnosti i brzinu usklađivanja. Odabirom manjih vrijednosti parametara, broj provedenih iteracija bit će manji, a s time i točnost usklađivanja. Povećavanjem broja iteracija dolazi do produljenja vremena potrebnog za usklađivanje, što u slučaju obrade velike količine fotografija može trajati i nekoliko sati. Time bi se prouzrokovali zastoji u procesu održavanja i nemogućnost korištenja zrakoplova što utječe na povećanje troškova. Nakon određivanja optimalnih parametara usklađivanja i dobivenih rezultata daljnji proces otkrivanja anomalija provodi se usporedbom fotografija na način da se pronađu ona područja na novoj fotografiji koja ne odgovaraju područjima na staroj fotografiji. Zatim je potrebno odrediti optimalnu vrijednost praga osjetljivosti. Odabirom nedovoljno velike vrijednosti praga osjetljivosti, broj šumova na fotografiji je jako velik. Odabirom manje vrijednosti praga osjetljivosti vrijednosti koje bi mogle predstavljati određene anomalije grupiraju se sa šumovima. Grupiranjem anomalija i šumova dolazi do problema određivanja područja s anomalijom jer se prevelika i premala područja uklanjaju iz daljnje analize. Nakon završne analize, koja određuje broj anomalija i označava područja anomalija prema danim parametrima, omogućen je prikaz fotografije. Temeljem dobivenog prikaza, osoba zadužena za

održavanje ima mogućnost povećavanja kako bi dobila detaljniji uvid u pronađene promjene između postojeće i nove fotografije temeljem čega se donosi odluka o daljnjim postupcima vezanim uz proces održavanja.

Nedostatak primijenjenog algoritma je što se prilikom snimanja fotografija može pojaviti rotacija. S manjim odklonom u rotaciji, primijenjeni algoritam se relativno uspješno nosi, dok s većim odklonom u rotaciji, dolazi do većih odstupanja. Dugi nedostatak primijenjenog algoritma je što nije u mogućnosti samostalno odrediti optimalnu vrijednosti parametra praga osjetljivosti za svaki novi set fotografija. Navedeno ograničenje može predstavljati veliki problem prilikom optimizacije jer velika količina šuma uzrokuje pronalaženje anomalija koje se ne bi trebale pronaći. Pronalazak velikog broja anomalija rješivo je kasnijim vizualnim pregledom, što će se odraziti na potrebno vrijeme zahvata održavanja, ali ne i na kvar. Lošiji slučaj od prevelike količine šuma je kada šum uopće ne postoji. Tada anomalije, koje su trebale biti otkrivene nisu uopće zabilježene što može dovesti do problema.

Smjer daljnjeg istraživanja može uključivati dodatno istraživanje mogućnosti MATLAB-ovog alata za obradu fotografije kako bi se uklonili šumovi nastali zbog vanjskog utjecaja, na primjer svjetla u hangaru. Druga mogućnost razvoja istraživanja je razvoj u smjeru računalnog vida koji bi u stvarnom vremenu omogućio prikaz razlika u odnosu na prethodno stanje. Takva funkcionalnost može se ostvariti implementacijom neuronskih mreža i sustava za strojno učenje. Primjenom takvih algoritama, prediktivnom sustavu, bilo bi omogućeno automatsko prepoznavanje razlika, bez fizičke provjere održavatelja koji određuje operativno stanje. Time bi se poduzeli daljnji koraci održavanja.

Literatura

- [1] The British Standard Institution. 2018. *Maintenance – Maintenance terminology BS EN 13306:2017*. BSI Standards Limited 2018.
- [2] Fredriksson, Gustav; Larsson Hanna. 2012. *An analysis of maintenance strategies and development of a model for strategy formulation – A case study*. Chalmers University of Technology, Göteborg.
- [3] Fluke. 2005. *The basics of predictive / preventive maintenance*. Fluke Corporation. Everett.
- [4] Kodnić, Veljko; Horvat, Marko; Maroević, Franjo. 2013. *Primjena dijagnostike kao osnove održavanja po stanju motora osobnog automobila*. Tehnički glasnik 7, 1(2013), 35-41.
- [5] Mobley, R. Keith. 2002. *An Introduction to Predictive Maintenance*. Elsevier Science. Woburn.
- [6] Sullivan, G. P.; Pugh, R.; Melendez, A. P.; Hunt, W. D. 2010. *Operations & Maintenance Best Practices – A Guide to Achieving Operational Efficiency*. U.S. Department of Energy.
- [7] Miletić, Alen. 2002. *Dijagnostičke metode i kriterij za ocjenu elektromehaničkog stanja asinkronog stroja*. Sveučilište u Zagrebu – Fakultet elektrotehnike i računarstva, Zagreb.
- [8] Fišer, Rastko; Ferkolj Stanislav. 1998. *Modeliranje in dijagnostika napak v pogonih z asinkronskim motorji: doktorska disertacija*. Ljubljana.
- [9] Mulder, Michel; Harman, Mark. 2018. *PdM 4.0. Beyond the hype 2018*. PwC Netherlands.
- [10] 2016. *The 5 Step Roadmap to IoT-Based Predictive Maintenance*. XMPro. Sjedinjene Američke države
- [11] Cognex. 2016. *Introduction to machine vision – A guide to automating process & quality improvements*. Cognex. Sjedinjene Američke države.
- [12] Davies, E. R. 2018. *Computer Vision – Principles, Algorithms, Applications, Learning – Fifth Edition*. Mara Conner. London.
- [13] Szeliski, Richard. 2010. *Computer Vision: Algorithms and Applications*. Springer.
- [14] Zhou, Huiyu; Wu, Jiahua; Zhang, Jianguo. 2010. *Digital Image Processing – Part I*. Ventus Publishing. Telluride.
- [15] Zhou, Huiyu; Wu, Jiahua; Zhang, Jianguo. 2010. *Digital Image Processing – Part II*. Ventus Publishing. Telluride.
- [16] Dabas, Poonam; Kumar Umesh. 2014. Pattern Recognition using Artificial Neural Network. *International Journal of Computer Applications Technology and Research*. Volume 3 – Issue 6, 358 – 360.

- [17] Xiang, Yu; Alahi, Alexandre; Savarese, Silvio. 2015. *Learning to Track: Online Multi-Object Tracking by Decision Making*. Santiago.
- [18] MATLAB. 2019. *Image Processing Toolbox – User Guide*. MathWorks.
- [19] MATLAB. 2016. *Computer Vision System Toolbox – Getting Started Guide*. MathWorks.

PRILOZI

I. CD-R

II. SKRIPTA ZA USKLAĐIVANJE I USPOREDBU SLIKA: *Obrada_slike.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Pocetni dio koda vezan je uz učitavanje fotografija.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Učitavanje fotografija
pocetna = imread('DJI_0200.jpg');
nova = imread('DJI_0200_n.jpg');

% Prikazivanje učitanih fotografija
figure
imshow(pocetna)
figure
imshow(nova)

% Pretvaranje učitanih fotografija u sive tonove
pocetnaCB = rgb2gray(pocetna);
novaCB = rgb2gray(nova);

% Usporedni prikaz fotografija u sivim tonovima
figure
imshowpair(pocetnaCB, novaCB, 'falsecolor')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Kod u nastavku vezan je za poravnanje fotografija zbog neusklađenosti
% nastalih nemogućnošću identičnog pozicioniranja bespilotne letjelice
% prilikom ponovnog fotografiranja.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Stvaranje optimizatora i metrike
[optimizer, metric] = imregconfig('multimodal');

% Podesavanje svojstva optimizatora
%
% potrebno je podesiti svojstva kako bi fotografije bile sto je moguće više
% usklađene kako bi se uz pomoć ostatka koda pronašle razlike između
% početne i nove fotografije.
optimizer.InitialRadius = 0.0001;
optimizer.Epsilon = 1.5e-5;
optimizer.GrowthFactor = 1.01;
optimizer.MaximumIterations = 300;

% Usklađivanje fotografija dobivenim vrijednosima optimizatora i metrike
novaUsklađena = imregister(novaCB, pocetnaCB, 'affine', optimizer, metric);

% Preklap usklađene i početne fotografije
figure
imshowpair(pocetnaCB, novaUsklađena, 'blend')

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Dio koda u nastavku vezan je za pronalazenje promjena izmedu pocetne i
% nove (uskladene) fotografije
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Pronalazak promjena izmedu fotografija
promjena = imabsdiff(pocetnaCB, novaUskladena);

% Prikaz razlike izmedu fotografija
figure
imshow(promjena)

% Pronalazak najvece vrijednosti promjene izmedu slika
maxPromjena = max(max(promjena));
[iRow, iCol] = find(promjena == maxPromjena);

% Prikaz tocke s najvecom vrijednosti promjene izmedu slika
figure
imshow(promjena)
hold on
plot(iCol,iRow,'r*')
hold off

% Odredivanje praga osjetljivosti
%
% Varijabla slikaPraga sadrzi samo one vrijednosti slike gdje je promjena
% dvije fotografije veca od 8

slikaPraga = promjena > 8;

% Prikaz fotografije sa određenim pragom osjetljivosti
figure
imshow(slikaPraga)
hold on
plot (iCol,iRow,'r*')
hold off

% Ciscenje sumova sa slike praga
%
% uklanjaju se svi objekti koji imaju manje od 8 piksela i stvara se nova
% slika
cistaSlika = bwareaopen(slikaPraga, 8);

% Prikaz ociscene slike
figure
imshow(cistaSlika);

% Stvaranje fotografije koja na novoj slici ima masku pomocu koje su
% vidljive promjene u odnosu na pocetnu sliku
preklop = imoverlay(novaUskladena,cistaSlika,'g');

% Prikaz fotografije sa vidljivim promjenama
figure
imshow(preklop)
```



```

% Pozivanje funkcije koja daje konacni prikaz slike i broj gresaka u odnosu
% na pocetnu fotografiju
%
% if petlja služi da se u slučaju kada na novoj fotografiji nema promjena
% pojavi poruka da nema vidljivih promjena, a u slučaju kada ima promjena
% da se ispiše broj promjena koje se uz pomoć pozvane funkcije i oznace
properties = filterRegions(cistaSlika);

if isempty(properties)
    figure
    imshow(novaUskladena);
    title("Nema vidljivih promjena");
else
    total_count = height(properties);
    rgb = insertShape(novaUskladena, 'rectangle', [properties.BoundingBox],
        'color', 'magenta');
    figure
    imshow(rgb);
    title("Broj promjena: " + total_count);
end

```

III. SKRIPTA ZA BROJANJE I OZNACAVANJE PROMJENA: *filterRegions.m*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Funkcija u nastavku služi za pronalazak promjena
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function properties = filterRegions(cistaSlika_in)

% Deklariranje naziva varijable
cistaSlika_out = cistaSlika_in;

% Uklanjanje promjena koje se nalaze uz rub fotografije
cistaSlika_out = imclearborder(cistaSlika_out);

% Filtriranje promjena na slici koje imaju površinu čija je donja granica
% veća od 9 piksela, a gornja granica teži u beskonačnost
cistaSlika_out = bwpropfilt(cistaSlika_out, 'Area', [9 + eps(100), Inf]);

% Dohvaćanje svojstava vezanih uz sliku
properties = regionprops(cistaSlika_out, {'Area',
    'BoundingBox', 'Eccentricity', 'EquivDiameter', 'EulerNumber',
    'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'Perimeter'});

% Odvajanje svojstava na temelju kojih se radi daljnja analiza
properties = sortProperties(properties, 'Area');

% Stvaranje petlje koja provjera ima li promjena na novoj fotografiji u
% odnosu na postojecu. Ukoliko nema promjena kod se zaustavlja, a ako ima
% promjena stvara se strukturni niz sa svojstvima promjena
if isempty(properties)

else
    properties = struct2table(properties);
end

```

```
% Stvaranje nove funkcije koja na temelju svojstva i strukturnog niza vraca  
% koordinate promjena koje zadovoljavaju trazena svojstva  
function properties = sortProperties(properties, sortField)  
  
% Sortiranje promjena silazno  
[~,idx] = sort([properties.(sortField)], 'descend');  
  
% Promjena redosljeda cijele strukture  
properties = properties(idx);
```