

# DETEKCIJA VLITIH NAVOJEV S POMOČJO TERMOVIZIJE

ANŽE ŠVIGELJ IN BORUT BATAGELJ

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija.  
E-pošta: as9467@student.uni-lj.si., borut.batagelj@fri.uni-lj.si

**Povzetek** Cilj članka je opis razvoja spletne aplikacije za detekcijo vlitih navojev segrevane deske za sneg s pomočjo termografske kamere v okolju SICK AppStudio, ki omogoča razvoj v programskem jeziku Lua na računalniku SICK SIM4000. V začetku sta predstavljeni področji računalniškega vida in termografskih kamer. V naslednjem koraku je predstavljena oprema in pripadajoče tehnologije, ki smo jih uporabili. V nadaljevanju pa je prikazana implementacija dveh algoritmov za detekcijo navojev: zaznavanje področij in Houghova transformacija kroga ter njuna primerjava.

**Ključne besede:**

termovizija,  
Lua,  
SICK  
AppStudio,  
termografska  
kamera,  
Houghova  
transformacija

# MOLDED THREAD DETECTION USING THERMAL IMAGING

ANŽE ŠVIGELJ & BORUT BATAGELJ

University of Ljubljana, Faculty of computer and information science, Ljubljana,  
Slovenia.

E-mail: as9467@student.uni-lj.si, borut.batagelj@fri.uni-lj.si

**Abstract** The aim of this article is to concisely describe the development of a web application for detecting molded threads of a heated snowboard using a thermographic camera in the SICK AppStudio environment, which uses the Lua programming language on the SICK SIM4000 industrial computer. In the beginning, the areas of computer vision and thermographic cameras are described and presented. In the following chapter, we presented the equipment we worked on and the associated technologies we used. We continued with the description of the implementation of two algorithms we used for detecting threads: blob detection and Circle Hough Transform then compared the results.

**Keywords:**

thermal  
imaging,  
Lua,  
SICK  
AppStudio,  
thermal  
camera,  
Hough  
transform

## 1 Uvod

Termografija je dandanes ključnega pomena v industriji, policiji, medicini, vojski in drugje. Pri policijskem delu je lep praktičen primer tega iskanje oseb in predmetov ponoči in ob omejeni vidljivosti. Poleg termografije je danes med nami vse bolj prisoten tudi računalniški vid in se pojavlja na vse več področjih. V industriji nam na primer koristi pri ločevanju na slabe in dobre izdelke ter polizdelke. Računalniški vid je tudi vse bolj prisoten na merilcih hitrosti na različnih avtocestnih odsekih, saj je boljša in cenejša alternativa klasičnim fotoradarskim merilcem in omogoča branje tablic in njihovo shrambo v podatkovno bazo (Czajewski in Iwanowski, 2010).

V članku je predstavljeno iskanje vlitih navojev na deski za sneg (angl. snowboard) s termovizijo, ki je kombinacija zgoraj navedenih pojmov. Predstavljena bo obdelava slik in algoritma za detekcijo, z zajemom iz termografske kamere FLIR A615.

Ciljna aplikacija teče na industrijskem računalniku SICK SIM4000 in je razvita v okolju SICK AppSpace v programskem jeziku Lua (Jerusalimschy, 2016). Deska je ob izdelavi v tovarni navadno premazana s smolo oziroma posebno zmesjo in se zato lukenj z navoji, ki služijo namestitvi vezi ne vidi. Ideja je, da celotno desko enakomerno segrejemo in s termografsko kamero preverimo, kje so ti navoji, saj so luknje narejene iz kovine in se ohlajajo počasneje kot preostali del deske.

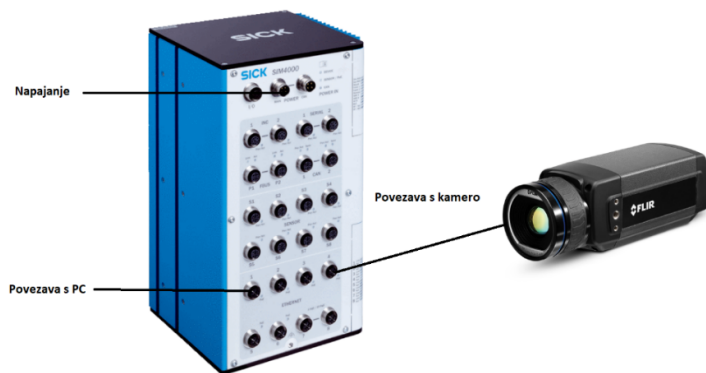
## 2 Uporabljene tehnologije

### – Industrijski računalnik SICK SIM4000

Računalnik SICK SIM4000 je namenjen izdelavi in izvajanju aplikacij povezanih z računalniškim vidom, ki ga je izdelalo nemško podjetje SICK, ki se v večini ukvarja z izdelavo kamer in senzorjev za področje računalniškega vida ter z razvojem namenskih aplikacij za uporabo njihovih izdelkov (SICK, 2021). Računalnik ima 24 priključkov (Slika 1), ki služijo povezavam s kamerami, senzorji, osebnim računalnikom za namen razvoja in režo za lokalno shranjevanje podatkov na SD kartico.

## – Termografska kamera FLIR A615

Za zajem slik smo uporabili termografsko kamero FLIR A615 (Slika 1) z vidnim kotom 25°. Omogoča zajem slik v ločljivosti 640 x 480 slikovnih elementov, ki jih predstavi v 16-bitnem formatu. Naprava ima razpon zaznavanja temperature med 20 °C in 2000 °C z natančnostjo  $\pm 2$  °C ( $\pm 2$  %) na posamezno branje temperature.



Slika 1: Prikaz povezav na industrijski računalnik SICK SIM4000 s povezano termografsko kamero FLIR A615.

Vir: lasten

## – Razvojno okolje SICK AppStudio

Razvojno okolje je namenjeno razvijanju in poganjanju aplikacij na računalniku SICK SIM4000. Razvoj poteka v skriptnem jeziku Lua.

Med drugim smo za namene shranjevanja podatkov v bazo uporabljali tudi odprtokodno večplatformsko strežniško okolje Node.js in program za podatkovne baze MongoDB.

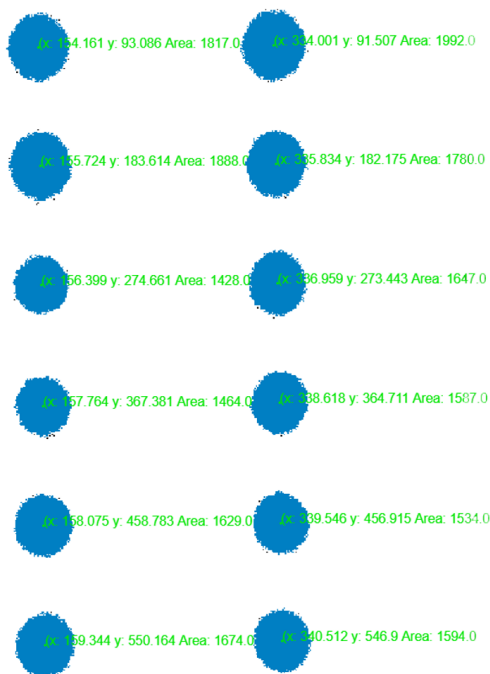
### 3 Razvoj aplikacije

Aplikacija ima naslednje funkcionalnosti: gumb za vklop in izklop kamere, nastavitvev IP naslova kamere, nastavitvev mejne vrednosti binarizacije, nastavitvev parametra velikosti področji in raztezka posameznega področja, gumb za obdelavo slike, prikaz surove slike, prikaz obdelane slike z vidnimi koordinatami, shranjevanje slike,

obdelava slike z zaznavanjem področij in Houghove transformacije kroga, shranjevane v bazo in vpogled v bazo.

### 3.2 Zaznavanje področij

Pri metodi zaznavanja področij (angl. blob detection) posneto sliko najprej pretvorimo v sivinsko in jo potem binariziramo, kar je postopek pretvorbe sivinske slike s 256 odtenki v sliko z dvema barvama, črno in belo, kar izloči nepomembne elemente nad nastavljeno mejno vrednostjo. Iz binarizirane slike nato izvlečemo vsa področja, ki imajo več kot 20 črnih slikovnih elementov skupaj in jih shranimo v tabelo področij. Nad tabelo smo nastavili še filtra površine (angl. area) in raztezka (angl. elongation), kar izloči vse neprimerne objekte iz tabele. Za določitev središča smo v okolju SICK AppSpace iz knjižnice PixelRegion uporabili funkcijo `Image.PixelRegion.getCenterOfGravity(blobs[i], image)`, ki na posameznih področjih določi težišče in vrne koordinate težišča oziroma v našem primeru središče (Slika 2).



Slika 2: Grafični izris področij in koordinat.

Vir: lasten.

### 3.3 Houghova transformacija kroga

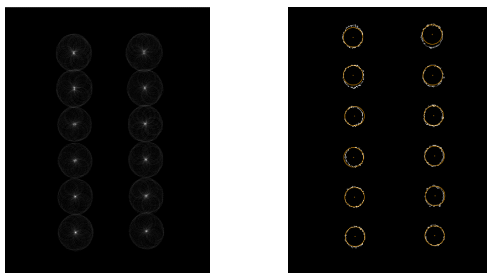
Pri Houghovi transformaciji kroga smo nad binarizirano sliko uporabili Cannyjev detektor robov, s pomočjo katerega dobimo sliko robov, ki z visoko verjetnostjo sovpadajo z robovi na binarizirani sliki. Nato se nad vsakim robnim slikovnim elementom nariše krožnica z želenim polmerom, kjer je robni slikovni element središče kroga, okoli celotnega roba, dokler ne pridemo na isto mesto. Posamezen krog lahko predstavimo v enačbi 1.

$$r^2 = (x - a)^2 + (y - b)^2 \quad (1)$$

V enačbi 1  $a$  in  $b$  predstavljata središče kroga,  $r$  predstavlja polmer,  $x$  in  $y$  predstavljata točko na krožnici.

Rezultat je akumulatorsko polje vidno na prvi sliki 3. Ker krožnice v našem primeru niso popolne krivulje smo v akumulatorskem polju določili lokalne maksimume, ki so v našem primeru točke na akumulatorju, kjer je največje število stičišč akumuliranih krožnic oziroma kjer je jakost slikovnih elementov na akumulatorju najvišja. Lokalni maksimumi predstavljajo najboljše kandidate za središča krožnic (OpenCV, 2022). Algoritem lahko zapišemo s psevdokodo v šestih korakih:

1. Najdi robove na sliki s Cannyjevim detektorjem robov
2. Za vsako področje
  3. Za vsako točko na robu
    4. Nariši krožnico s središčem v tej točki na robu s polmerom  $r$
    5. Poišči lokalni maksimum v akumulatorskem polju
6. Najdene parametre  $(r, a, b)$ , ki ustrezajo lokalnim maksimumom, preslikaj nazaj na izvirno sliko (Slika 3).



Slika 3: Akumulatorsko polje (levo) in izris najdenih krogov na vhodni sliki robov (desno).

Vir: lasten.

### 3.4 Primerjava obeh metod

Pri primerjavi obeh metod smo na sliki ročno določili središča za vsak posamezen navoj tako, da smo poljubno izbrali tri točke na posamezni krožnici in s pomočjo njih določili središče. Koordinate središč navojev po vseh treh metodah so vidne v tabeli 1. Na dnu tabele je prikazano še povprečno odstopanje posamezne metode od ročnega izračuna središč. Za izračun odstopanja smo uporabili formulo Pitagorovega izreka za računanje razdalje med dvema točkama v koordinatnem sistemu. Na podlagi podatkov in izračunov lahko vidimo, da je točnost določanja središč pri obeh metodah primerljiva, saj je razlika med njima v povprečju le desetinka slikovnega elementa in od ročno izmerjenih središč v povprečju odstopata le za en slikovni element.

Tabela 1: Primerjava razlik med ročnim izračunom središč in uporabljenima metodama.

št. navoja	ročno		zaznavanje področij	Hough
	x	y	odstopanje	odstopanje
1	115	167	1,6826	2,5838
2	206	165	0,9156	1,6466
3	296	166	2,0807	1,2236
4	388	165	1,5605	1,4040
5	481	164	0,8021	0,9960
6	571	163	0,7066	0,8884
7	113	345	1,3319	1,2606
8	205	345	0,2745	1,2964
9	298	347	2,0796	2,8351
10	387	346	2,1942	2,8317
11	480	346	0,7498	0,3417
12	572	343	1,5816	0,5450
povprečje:			1,3300	1,4877

### Literatura

- Czajewski, W., in Iwanowski, M. (2010). Vision-based vehicle speed measurement method. V International Conference on Computer Vision and Graphics (str. 308-315). Springer, Berlin, Heidelberg.
- Ierusalimsky, R. (2016). Programming in Lua, Fourth Edition. Lua.org
- OpenCV, Open Source Computer Vision. (2022). Hough Circle Transform. [https://docs.opencv.org/3.4.15/d4/d70/tutorial\\_hough\\_circle.html](https://docs.opencv.org/3.4.15/d4/d70/tutorial_hough_circle.html)
- SICK, Sensor Intelligence. (3. 8. 2021) SIM4x00, Flexible. Intelligent. Communicative. [https://cdn.sick.com/media/familyoverview/2/52/452/familyOverview\\_SIM4x00\\_g386452\\_en.pdf](https://cdn.sick.com/media/familyoverview/2/52/452/familyOverview_SIM4x00_g386452_en.pdf)

