

# ZAZNAVA POLOŽAJA KROGLICE NA RULETI IZ VIDEO POSNETKA

Tadej Zupančič, Aleš Jaklič

Laboratorij za računalniški vid  
Fakulteta za računalništvo in informatiko  
E-pošta: [tadej.zupancic@fri.uni-lj.si](mailto:tadej.zupancic@fri.uni-lj.si)

---

**POVZETEK:** *Prispevek obravnava algoritem za zaznavo položaja kroglice in pripadajoče številke na cilindru rulete. Kot vhod je uporabljen video posnetek iz datoteke z možnostjo enostavne razširitve na »živi« video iz video kamere.*

---

## 1. UVOD

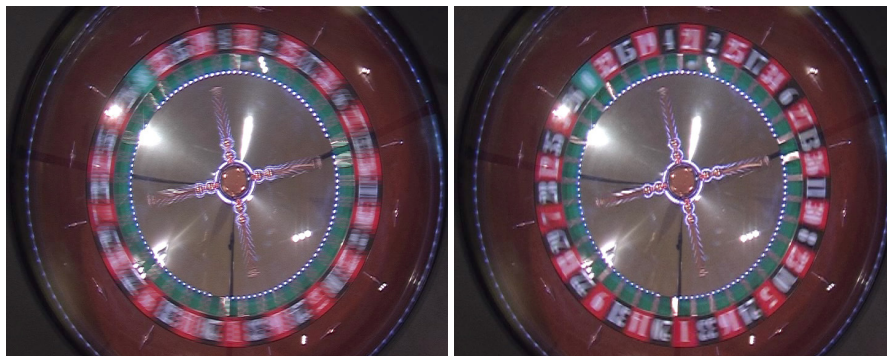
Ruleta je ena izmed najstarejših igralnih naprav za izvajanje iger na srečo. V preteklem desetletju je ruleto v obliki igralnega avtomata pričelo razvijati, izdelovati in prodajati na svetovnem tržišču kar nekaj podjetij v Sloveniji. Običajno se za zaznavanje položaja kroglice uporablja eden ali do trije optični senzorji, ki so nameščeni v enakomernih kotnih razmikih po obodu cilindra in v povezavi z enkoderjem kota zasuka cilindra omogočajo določitev položaja kroglice. V nadaljnjem besedilu je opisan postopek, s katerim lahko z uporabo informacij v video posnetku dokaj natančno določimo položaj kroglice in temu položaju pripadajočo številko na ruleti. Takšen avtomatski postopek omogoča kontrolo pravilnosti delovanja drugih senzorjev, kot tudi vključitev predlaganega sistema v že obstoječe naprave brez mehanskih ali električnih posegov.

## 2. VHODNI PODATKI

Video posnetek, ki ga posname video kamera, še ni primeren za takojšnjo analizo in obdelavo, saj je po PAL standardu sestavljen iz okvirjev velikosti 720×576 pik, ki si sledijo s hitrostjo 25 okvirjev/sekundo. Vsak okvir je prepleten iz dveh zamaknjenih polj velikosti 720×288, od katerih eno vsebuje le lihe, drugo pa le sode vrstice.

Pri predvajanju posnetka na televiziji tega ponavadi ne opazimo, pri pregledovanju posameznih okvirjev pa je učinek takega združevanja lepo viden predvsem pri dinamičnih posnetkih. Pred obdelavo je zato potrebno odpraviti prepletenost, saj si s tem precej izboljšamo vhodne podatke.

Za odpravljanje prepletenosti obstaja več pristopov. Nekateri spuščajo lihe oziroma sode vrstice, nekateri pa okvir razstavijo na polji in ju predvajajo zaporedno, pri čemer imamo iz tega razloga sedaj 50 polovično visokih okvirjev/sekundo. Pri tej aplikaciji smo uporabili drugi pristop skupaj s filtrom, ki nam višino videa interpolira nazaj na 576 pik in zgladi prehode med posameznimi polji. Razlika med prepletenim in neprepletenim okvirjem je vidna na sliki 1.



Slika 1: Razlika med prepletenim (levo) in neprepletenim okvirjem (desno)

### 3. RAZLAGA ALGORITMA

#### 3.1 Uporabnost video posnetka

Video posnetek vsebuje dosti več informacij kot statična slika, saj si pri analizi določenega okvirja lahko pomagamo s prejšnjimi okvirji, za katere približno vemo, za koliko se razlikujejo od opazovanega. V tem primeru se številke in kroglica na ruleti gibljejo po določeni krožnici (elipsi), zato lahko predvidimo, kje se v določenem trenutku nahaja objekt, ki ga iščemo.

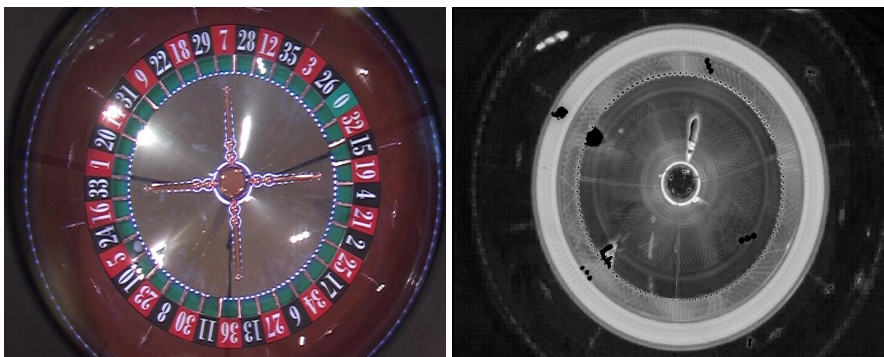
#### 3.2 Tehnika zajema

Za uspešen video posnetek potrebujemo dovolj svetlobe, saj le tako lahko zagotovimo dovolj kratek čas zaklopa in se izognemo t.i. »motion blur-u«. Na sliki 1 je bil čas zaklopa predolg, zato slika ni tako ostra kot slika 2, kjer je bil čas zaklopa krajši.

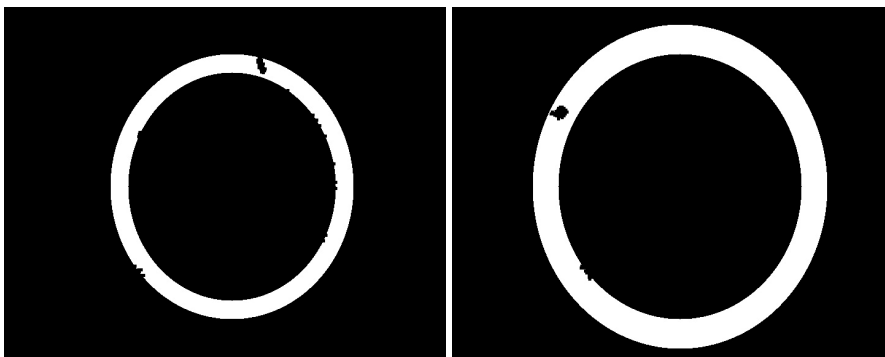
#### 3.3 Problem odsevov in odbleskov

V prostorih, kjer stojijo rulete, je zaradi boljšega izgleda prostorov ponavadi veliko luči, ki lahko negativno vplivajo na samo sliko. Pojavljajo se razni odsevi in odbleski, ki se jim težko izognemo. Ti odbleski se na srečo zaradi mirovanja kamere pojavljajo vedno na istem delu slike in jih zato lahko zaznamo.

Na sliki 2 je prikazana slika, na kateri ima vsaka točka vrednost enako razliki največje in najmanjše vrednosti te točke v celotnem posnetku. Vrednosti točk, na katerih vidimo odsev, so preko celega posnetka enake, zato so na tej sliki obarvane črno. Iz te slike lahko sestavimo dve maski odsevov (slika 3), ki jih potem uporabljamo pri obdelavi slike. Eno masko uporabljamo pri zaznavi kroglice, drugo pa pri zaznavi številke.



Slika 2: Uporabljen je bil krajši čas zaklopa (levo), slika spreminjanja vrednosti točk preko posnetka (desno)



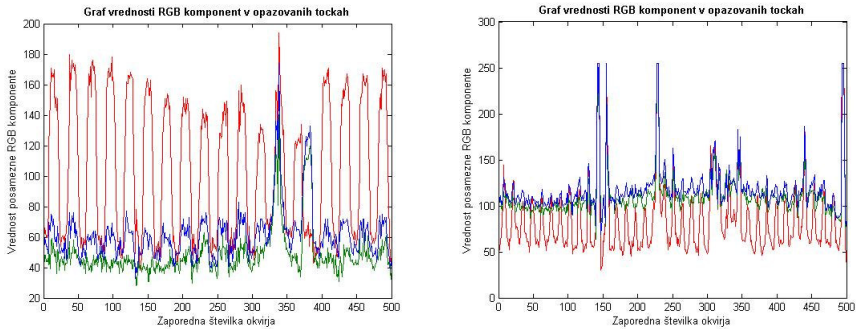
Slika 3: Maski, na katerih so z belo označene točke, ki so »uporabne«

### 3.4 Izbira opazovanih točk

Po določitvi središča in radija krožnice (elipse) lahko natančno določimo področje na ruleti, kjer se lahko nahaja kroglica in področje na katerem so številke. Na vsakem izmed teh dveh področij izberemo krožnico in opazujemo samo točke na krožnici. Razširjeni sliki zaporedja opazovanih točk in njima pripadajoča grafa sta na slikah 4 in 5.



Slika 4: Zaporedje opazovanih točk pri iskanju položaja zelenega polja (levo), zaporedje opazovanih točk pri iskanju položaja kroglice (desno)



Slika 5: Graf opazovanega zaporedja točk pri iskanju položaja zelenega polja (levo), graf opazovanega zaporedja točk pri iskanju položaja kroglice (desno)

### 3.5 Iskanje kroglice

Pri iskanju kroglice najprej izračunamo lokalne ekstreme na zaporedju opazovanih točk (slika 5). Točke številčimo od 1 do  $n$ , kjer  $n$  predstavlja število točk na opazovani krožnici. Točka  $i$  je lokalni maksimum, če zanjo velja:

$$y(i-1) < y(i) \text{ in } y(i) > y(i+1)$$

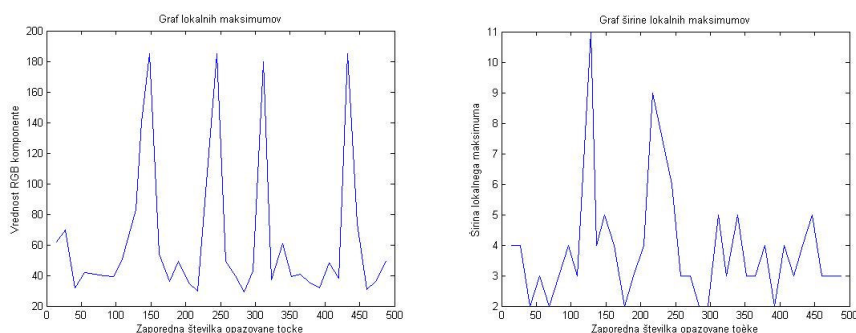
ter lokalni minimum, če velja:

$$y(i-1) > y(i) \text{ in } y(i) < y(i+1)$$

V izogib prevelikemu številu lokalnih ekstremov uvedemo še dva pogoja. Razlika vrednosti dveh sosednjih nasprotnih ekstremov mora biti višja od nekega vnaprej postavljenega praga. Drugi pogoj pa namesto dveh sosednjih enakih ekstremov potrди samo večjega od obeh in tako zagotovi, da imamo izmenično zaporedje minimumov in maksimumov.

Po določitvi zaporedja lokalnih ekstremov izračunamo še širino posameznih lokalnih maksimumov. Vsak lokalni maksimum v zaporedju je obdan z dvema lokalnima minimumoma. Pregledamo vse točke med tema dvema lokalnima minimumoma in preštejemo število točk, ki so večji od nekega praga, ki je odvisen od višine lokalnega maksimuma, katerega širino računamo.

Med izračunanimi lokalnimi maksimumi (Slika 6) izberemo tiste, ki imajo največjo širino. Izmed teh pa lahko na podlagi prejšnjih položajev kroglic in hitrosti premika kroglice med posameznimi okvirji določimo, kateri lokalni maksimum predstavlja kroglico.



Slika 6: Zaporedje lokalnih maksimumov (levo), graf širine lokalnih maksimumov (desno)

### 3.6 Iskanje številke 0

Za uspešno določanje številke, na katero je padla kroglica, potrebujemo referenčno točko, za katero vemo, katera številka ji pripada. S pomočjo te točke in s položajem kroglice lahko izračunamo številko na kateri je kroglica.

Ruleta je sestavljena iz 16 polj rdeče barve, 16 polj črne barve in številke 0, ki je zelene barve. Na grafu na sliki 4 lahko razberemo, da je na polju, ki predstavlja številko 0, vrednost zelene barve veliko višja kot na drugih poljih. Izjema je še vrh pri zaporedni številki okvirja 345, ki predstavlja odsev in ga glede na masko na sliki 3 ne obravnavamo. Iz vrednosti RGB komponent tako vsaki točki zaporedja določimo barvo. Lokacijo številke 0 določimo kot srednjo točko v zaporedju zelenih točk. Zaradi odbleskov se nam lahko zgodi, da bo številka 0 ravno v trenutku opazovanja pod odbleskom in je zato na ta način ne bo mogoče zaznati. V tem primeru iz prejšnjih premikov izračunamo pričakovano lokacijo te točke v tem trenutku.

### 3.7 Določanje številke, na kateri je kroglica

Iz zaporedne številke točke, v kateri je izbrani lokalni maksimum, zaporedne številke točke, ki predstavlja število 0 lahko z uporabo kota med točkama izračunamo na katerem polju se nahaja kroglica. Po ugotovitvi zaporedne številke polja pa lahko iz tabele preberemo polju pripadajočo številko.

## 4. REZULTATI IN NADALJNI RAZVOJ

Z opisanim algoritmom nismo imeli težav pri določanju polja s številko 0, saj algoritem tudi v primeru, da je polje v trenutku opazovanja pod odbleskom, najde ustrezno rešitev. Pri določanju položaja kroglice pa odbleski povzročajo večje težave. To je sicer problem pri določanju položaja kroglice v določenem okvirju, vendar lahko z opazovanjem več zaporednih okvirjev ta problem izničimo.

Pri uporabi opisanega algoritma se dostikrat zgodi, da izbrani lokalni maksimum pri iskanju položaja kroglice ni točno na kroglici, ampak na njenem robu oziroma na enem izmed dveh prekatov, ki omejujeta polje. V tem primeru nam algoritem kot rezultat lahko vrne število, ki je eno polje levo ali desno od števila, na katerem se nahaja kroglica.

Ta problem je enostavno rešljiv z implementacijo algoritma, ki v ožji okolici izbranega lokalnega maksimuma poišče kroge (elipse) in tako določi središče kroglice oziroma ugotovi, da se kroglica tam ne nahaja. Prav tako imamo namen, vkolikor se bo algoritem izkazal za prepočasnega za obdelavo v realnem času, del algoritma implementirati z uporabo procesorja grafične kartice[1] in programskega jezika Cg[2].

Rezultati na video posnetku z 260 okvirji so podani v tabeli 1.

Tabela 1: Rezultati zaznave številke, na katero je padla kroglica

<i>VRSTA NAPAKE</i>	<i>ŠTEVILO NAPAK</i>	<i>ODSTOTEK NAPAK</i>
Napaka za več kot 1 polje	4	1,54%
Napaka za 1 polje	22	8,46%
Skupno napačna številka	26	10%

## 5. ZAKLJUČEK

Igralec, ki igra ruleto, bi rad čimprej vedel, na katero številko je padla kroglica, saj je od tega odvisno, ali je kaj zadel ali ne. Glede na to, da je v igralnicah za posamezno mizo pogosto gneča, bi lahko opisano implementacijo s pomočjo dodatnega prikazovalnika uporabili za prikaz številke, na katero je padla kroglica.

Običajno se za take rešitve uporabljajo razne strojne rešitve, npr. senzorji za kroglico. Take strojne rešitve pa pogosto ne pridejo v poštev, saj zahtevajo poseg v samo ruleto. Zaradi zagotavljanja pravičnosti rulete, morajo biti rulete certificirane, za vsak poseg v ruleto pa je treba postopek certificiranja ponoviti, kar je zamudno in pogosto tudi cenovno nesprejemljivo.

Z montažo video kamere nad ruleto in montažo dodatnega prikazovalnika ne posegamo v ruleto in se tako izognemo ponovnemu certificiranju, hkrati pa dosežemo skoraj isti rezultat kot z uporabo strojnih rešitev.

## LITERATURA

1. J. Fung (2005), Computer Vision on the GPU, *GPU Gems 2*, Addison Wesley, str. 649-665
2. R. Fernando, M. J. Kilgard (2003), *The Cg Tutorial*, Boston: Addison-Wesley
3. T. Žgur (2005), *Detekcija položaja kroglice na igralniški ruleti s pomočjo računalniškega vida*, diplomsko delo, Ljubljana: Fakulteta za računalništvo in informatiko