



CHALMERS

# Hastighetsmätning genom avståndsmätning med hjälp av kamera från en smartphone

Med hjälp av openCV och Huawei P40 Pro beräkna  
hastigheten av en slagen golfboll

Examensarbete inom Data- och Informationsteknik

Carl Seeman

INSTITUTIONEN FÖR DATA OCH INFORMATIONSTEKNIK  
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021  
[www.chalmers.se](http://www.chalmers.se)

## Sammanfattning

Denna uppsats beskriver möjligheter med att använda sig av moderna datorsyns algoritmer för att kunna bestämma hastighet på objekt i rörelse. Huvudsyftet är att beräkna hastighet på en golfboll slagen av en spelare med hjälp av med de nyare smartphones med möjlighet att ta höghastighets-video. En beskrivning för metoden temporal differencing för att identifiera och analysera objekt i rörelse görs och det demonstreras varför det är en lämplig metod att tillämpa vid denna applikation.

En mängd golfslag har spelats in och analyserats samtidigt som en radar speciellt utformad för att mäta golfslag har använts som jämförelse och testat på applikationens precision. Slagen har spelats in på mobiltelefonen P40 Pro tillverkad av Huawei som klarat av att fånga bollens rörelse trots den mycket höga hastighet som bollen har färdats i.

Resultatet från den algoritm som utvecklats visar att den beräknade hastighet avviker i snitt 4,35 procent med en mycket liten varians. Det undersöks och reflekteras kring vad som kan ge upphov till denna avvikelse och att felaktigt uppmätt bollstorlek i programmet troligen begränsar annars utmärkt resultat.

Huvudpunkten i uppsatsen är att det är fullt möjligt att med hög precision mäta bollens hastighet vid ett golfslag med en smartphone och datorsyns-metoder och verktyg. Dock inte helt felfritt och för att kunna bevisa precisionen bättre krävs fler experiment och en bättre metod för att bestämma storleken på föremålet som ska analyseras. Det återstår att testa applikationen vid snabbare slag än de som gjort i de experiment som skett i detta arbete.

## **Abstract**

This thesis describes the possibilities of using modern computer vision algorithms paired with the new smartphones capable of capturing high framerate videos allow for measuring the velocity of a moving golf ball. Methods to detect moving objects is described and the case of why the method of temporal differencing is well suited for the purpose of measuring the velocity accurately is made.

A set of golf shots have been recorded on a smartphone and simultaneously analysed by a radar device specifically made for the tracking of golf shots. The purpose of the radar is to give accurate baseline velocity to be able to measure the accuracy of the application calculating the velocity. Shots have been recorded on a P40 pro smartphone made by Huawei which have successfully been able to capture images of the ball flying at very high velocities. The results of the algorithm running the analysis were able to calculate a velocity with an average deviation of 4,35% from the baseline velocity given by the radar unit. What causes this deviation and what parameters might impact the measured error is discussed and the ball size measured in pixels plays a central role in the calculation and is the most probable cause of the error.

The main conclusion of is that it is possible with high precision to measure the velocity of a golf shot with a smartphone capable with high-speed recording and modern computer vision methods and tools. It is not however easy to get the perfect results and to prove the accuracy further a lot more experiments and tweaking is required. Particularly in the region of better ability to measure the size of the ball using the camera.

# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>6</b>
1.1	Syfte . . . . .	6
1.2	Mål . . . . .	6
1.3	Avgränsningar . . . . .	6
<b>2</b>	<b>Teori &amp; Teknisk Bakgrund</b>	<b>8</b>
2.1	Golfboll . . . . .	8
2.2	Bilder . . . . .	8
2.3	Bilder per sekund . . . . .	9
2.4	Slutartid . . . . .	9
2.5	Datorsyn-Computer Vision . . . . .	9
2.5.1	OpenCV . . . . .	9
2.5.2	Filter . . . . .	9
2.5.3	Threshold . . . . .	10
2.5.4	Kantdetektering . . . . .	10
2.5.5	Morphologisk operation . . . . .	11
2.5.6	Rörelsedetektering . . . . .	11
2.6	Färgskalor . . . . .	11
2.7	Trackman . . . . .	12
<b>3</b>	<b>Relaterat Arbete</b>	<b>13</b>
3.1	ShotVision app . . . . .	13
3.2	Garmin R10 . . . . .	13
3.3	Yosuke Yamaguchi, Motoki Miura (2016) . . . . .	13
3.4	Jung et al (2010) . . . . .	14
<b>4</b>	<b>Metod</b>	<b>15</b>
4.1	Kamera . . . . .	15
4.2	Kamerans position . . . . .	15
4.3	Inspelning . . . . .	15
4.4	Utförande . . . . .	15
4.4.1	Importering av video . . . . .	15
4.4.2	Bildbehandling . . . . .	16
4.5	Hastighetsmätning . . . . .	16
4.5.1	Objekturskiljning . . . . .	16
4.5.2	Misslyckad Objekturskiljning . . . . .	18
4.5.3	Hastighetsbedömningen . . . . .	18
4.5.4	Beräkning av verklig hastighet . . . . .	18
4.6	Utförande av mätning . . . . .	19

4.7	Utrustning . . . . .	19
<b>5</b>	<b>Resultat</b>	<b>20</b>
5.1	Hastighetsmätning . . . . .	20
5.2	Felmätning . . . . .	21
5.3	Hastighetsförändring inom samma slag . . . . .	23
<b>6</b>	<b>Diskussion</b>	<b>24</b>
6.1	Begränsningar . . . . .	24
6.2	Experiment . . . . .	24
6.3	Resultatanalys . . . . .	24
6.4	Etik och miljö . . . . .	25
6.5	Framtida förbättringar . . . . .	25
	<b>Referenser</b>	<b>26</b>

# 1 Introduktion

De senaste 50 åren har forskning och utveckling inom datorsyn skapat enorma möjligheter för interaktion mellan maskin och människa. Även om ämnet fortfarande kan anses vara komplicerat har det kommit många uppfinningar och verktyg som varit till hjälp med att skapa imponerande tillämpningar i produkter. Dessa verktyg har gjort det möjligt att identifiera objekt och följa deras rörelse i en videosekvens vilket är det område som kommer beröras i denna rapport [1].

Då golfspelare som har möjlighet att slå bollen långt har en fördel blir det en viktig del att kunna träna på. Den enskilt största faktorn för att slå långt är den hastighet som spelaren kan sätta i golfbollen. För att som spelare kunna träna och förbättra sin förmåga att slå längre är det viktigt att kunna få feedback på sina slag. Feedback på om de swingteknikövningar man tränar på eller styrketräning spelaren gör utanför golfen hjälper spelare att slå längre.

Existerande lösningar för att kunna mäta hastighet på en slagen golfboll bygger på dopplerradarteknik och detta gör de komplicerade och dyra. De senaste åren har några tillverkare av mobiltelefoner levererat dessa med möjlighet att spela in video med högre bilder per sekund. Att kunna använda sig av kameran i de nyare mobiltelefonerna, som har en höghastighetskamera, kan göra hastighetsmätning mer tillgängligt för fler spelare om en applikation som körs på dessa utvecklas.

## 1.1 Syfte

Syftet med projektet är att utifrån video på ett golfslag taget från sidan inspelad på telefon kunna mäta utgångshastighet på den slagna golfbollen.

## 1.2 Mål

Huvudmålet framgår av syftet i att kunna mäta hastighet på golfbollen.

- Att kunna ställa upp sin mobil i en tripod vid sidan av golfbollen, slå golfbollen och få ut siffror på vilken hastighet bollen når efter att den är slagen.
- Jämföra precisionen mot annan utrustning för att se hur precisa mätningar man kan göra.

Projektet anses lyckat om man kan komma inom 5% av de värden som professionella mätinstrument visar.

## 1.3 Avgränsningar

Den stora begränsningen ligger i tillgången av kamerans API som begränsas av telefontillverkaren. Det gör att applikationen troligtvis inte blir helautomatisk i det att spela in slag, analysera hastighet och presentera siffror. Inspelningsmomentet är det svåra och avgränsningen blir då att denna måste triggas manuellt. Vilket hastighetsintervall som ska gå att

läsas av begränsas också. Önskvärt är att kunna mäta upp till 300 km/h eller 180 mph vilket skulle motsvara högsta slaghastighet hos amatörer med några få undantag.

## 2 Teori & Teknisk Bakgrund

För att förstå resultatet och diskussionen bättre behöver en del information och teori kring golf, datorsyn och algoritmer samt utrustning presenteras. Här beskrivs de områden som läsaren behöver ha med sig.

### 2.1 Golfboll



Figur 1: Golfboll

Golfbollens utformning är reglerad av R&A som är det organ som bestämmer reglerna för sporten golf. De tillverkas i en mängd olika färger men de måste ha en diameter på minst 42,67 millimeter och en maxvikt på 45,93 gram [14].

De bollar som har använts i testet tillverkas av Titleist och enligt en undersökning har de mycket tät tolerans gällande dess diameter på 42,7 millimeter. Detta är det mått som används i analysen [15].

### 2.2 Bilder

Bilder sparas som en samling av pixlar där varje pixel kan innehålla ett antal bitar beroende på vilken typ av bild. Bilden kan exempelvis vara i färg eller gråskala och detta ändrar hur många bitar varje pixel består av. För gråskala så har varje pixel ett värde kallat intensitet. Ett högt värde eller hög intensitet, motsvarar ljus område på bilden. Lågt värde motsvarar mörkt område. För färgbilder är ett vanligt format (RGB) eller (BGR) där varje pixel tilldelas tre värden och benämns efter den ordning som färgen representeras. RGB representeras av intensiteten för Rödfärg i pixeln som följes av intensiteten Gröna och sist Blå. Tillsammans kan dessa tre färger användas för att återskapa en mängd färger och i kombination av en



mängd pixlar kan då en bild representeras genom att ordnas i en matris. Denna lagringstyp av bilder kallas för bitmappsbild [1].

## **2.3 Bilder per sekund**

Då videos oftast består av en samling bilder ordnade i tid så visar detta hur många bilder som processas under en sekund. Video kommer i detta arbete ses som en samling bilder som ordnats sekventiellt efter tid. För att kunna fånga objekt som rör sig blir antalet bilder per sekund en viktig del. Om objektet rör sig snabbt kan det vara så att objektet flyttat sig för långt ”över bilden” så att bara några få bilder på objektet sparats. Fler bilder per sekund innebär att objektet fångas på flera bilder då den rör sig. Kommer i fortsättningen benämnas FPS för engelskans ”Frames per second”.

## **2.4 Slutartid**

När en bild ska tas så exponeras en bildsensor i kameran för ljus. Tiden som den sensorn exponeras kallas slutartid. En lång slutartid innebär att sensorn får möjlighet att samla in mycket ljus. Allt som händer under tiden som sensorn exponeras för ljus samlas in så föremål som rör sig kommer då inte vara fast på ett ställe utan kommer lämna avtryck över flera pixlar. Exempelvis framstår bollen i bilden som ett avlångt föremål då den under slutartiden befanns sig på olika platser, den lämnar ett spår efter sig. En högre FPS innebär att slutartiden måste bli kortare vilket leder till att föremål som rör sig snabbt blir mindre suddiga och då enklare att analysera. En kortare slutartid innebär dock att en mindre mängd ljus träffar sensorn vilket ställer mer krav på att objektet är väl upplyst när bilderna tas.

## **2.5 Datorsyn-Computer Vision**

Här nämns de metoder, verktyg och algoritmer som används för bildanalys

### **2.5.1 OpenCV**

OpenCV är ett stort mjukvarubibliotek av som innehåller en mängd bildbehandlingsmetoder och algoritmer. Det ges ut som BSD licensierad produkt vilket innebär att det är fritt att använda inom vissa ramar [16].

### **2.5.2 Filter**

I de flesta bilder så finns en viss nivå av störningar, för detta kan ett Gaussian filter användas. Det analyserar en  $N \times N$  matris av bilden och ger ett medelvärde av matrisen till den pixeln i mitten med en hög vikt på mittenpixeln så att den inte ändras för mycket av sin omgivning. En pixel som har en stor kontrast mot sina grannar kan ofta vara en icke önskvärd störning i bilden. Gaussian filter minskar då dessa ”utstickare” till pixlar, men som konsekvens gör den bilden något suddigare genom att minska kontrasten [1].

### 2.5.3 Threshold

En vanlig teknik i bildanalys är att använda sig av så kallad thresholding. Man sätter då ett gränsvärde och sedan analyseras intensiteten i varje pixel, är intensiteten på pixeln över detta gränsvärde sätts den pixelns nya värde till högsta/vit pixel. Är den under detta sätts den till det lägsta värdet/ svart pixel.

250	>	255
200	>	255
90	<	0
30	<	0

Figur 2: Thresholding har genomförts där det blå strecket visar gränsvärdet för intensiteten

För att använda detta som pre-processing av bilder skapas exempelvis en matris med samma dimension som bilden som analyseras. Sedan analyseras varje pixel och den nya matrisens motsvarande pixel fylls på med ettor för de ljusa områden och nollor för de mörka. Denna nya matris kan då användas som mask för det område som är intressant [1].

### 2.5.4 Kantdetektering

Kanter i bilder består vanligtvis av en skarp kontrast mellan närliggande pixlar. Exempelvis ett vitt sträck på en svart vägg har en tydlig kant. Bilder tagna på ett sådant tydligt exempel kommer ändå ha viss nivå av störning som kan komplicera analysen. Därför utgår de flesta algoritmer från en liknande uppsättning av funktioner. Först filtreras bilden för störningar, sedan förtydligas kontraster och thresholding sker följt av positionering av funna kanter. Filtrering sker med exempelvis gaussian [3].

Det nästkommande steget förstärker kontraster med olika metoder. Thresholding försöker ta bort de pixlar som anses vara störningar och positioneringen försöker samla ihop de hittade "kanterna" till sammanhängande kroppar. Positionerings metoden som appliceras i detta arbete bygger på Gaussian gradient och är en av de vanligare metoderna. Den algoritmen ger bra resultat på en stor bredd av applikationer och bakas därför ofta in i mer avancerade metoder för bildanalys. Algoritmen analyserar bilden och sparar en array över gradienter i intensitet. Resultatet analyseras och pixlar som har högt maxima (höga värden) räknas då som kanter i en bild. Efter detta så görs de kanter som har starka värden tydligare genom

thresholding medan de svagare kanter tas bort och i slutet får man en binär Array med de funna kanterna [3].

### **2.5.5 Morphologisk operation**

Morphologisk operation är ett sätt att förarbeta bilder innan de används till analys. Den använder sig av en kärna/kernel av ett visst mått tillsammans med exempelvis Erosion och Dilution. Erosion analyserar bilden och letar efter föremål som matchar den storlek på kernel som valts och mörkar ner närliggande pixlar som då inte ses som en del av det intressanta området. Detta leder till att mindre områden av ljusare pixlar som inte hänger ihop med det intressanta område ses som störningar i bilden och filtreras bort. Efter Erosion kan Dilution användas framgångsrikt då det intressanta området gjorts mindre än vad den var från början. Dilution tar och breddar ut de intressanta områdena och kan med rätt parametrar återställa den ursprungliga storleken på objektet [1].

### **2.5.6 Rörelsedetektering**

Det finns ett antal metoder för att finna objekt i rörelse inom datorsyn. Några av dessa är optical flow eller temporal differencing. Optical flow är en mycket beräkningsintensiv algoritm och ansågs inte passa lika väl för stora videofiler i en mobilapplikation. Temporal differencing blev den metoden som användes. Detta då den är snabb och kan enklare appliceras i en applikation [5, 8, 9]. Den går ut på att varje bild i en video jämförs pixel för pixel efter skillnader med sin efterkommande bild. Detta sparas som en array med samma storlek som originalet där varje pixel som inte har ändrats mellan bilderna är svart medan de pixlar som har ändrats är olika ljusa beroende på hur stor förändring som skett. Det går då att läsa ut från bilden vilka objekt som är i rörelse.

## **2.6 Färgskalor**

Som tidigare nämnt sparar kameran ner bilder som linjärkombinationen av röda, gröna och blå värden i separata kanaler. Dessa tre kanaler innehåller värden på styrkan varje färg i bilden har men dessa värden influeras starkt av hur mycket ljus som träffar objektet. Skillnaden mellan ett fotograferat objekt i ljus miljö kontra mörk är mycket stor och det blir svårt att särskilja objektet. Det är därför vanligt att konvertera en bild från RGB till HSV vid bildanalys. HSV står för Hue, Saturation och Value. Hue kanalen beskriver färgen för varje pixel. Saturation visar hur stark den färgen är och Value korrelerar med ljushet eller mängden ljus som faller på objekten [2]

Fördelen med HSV är att skillnad i ljus i en scen kan enklare hanteras då analys ska ske [4].

## **2.7 Trackman**

Den enhet som används som referens vid hastighetsmätning kallas trackman 4 och tillverkas av företaget Trackman. Till skillnad mot den analys av bilder som projektet gör så använder sig Trackman 4 av dopplerradar. Då den samlar in data med frekvens på 40 000 gånger per sekund så är dess upplösning mer precis än de 960 FPS som kameran klarar av [6].

### **3 Relaterat Arbete**

Här presenteras några liknande applikationer och forskningspapper som använder sig av olika metoder för att bestämma hastighet på föremål.

#### **3.1 ShotVision app**

ShotVision app är en applikation som finns tillgänglig för IOS. Den Använder sig av kameran för att bedöma utgångshastighet på bollen, klubbans hastighet samt utgångsvinkel på slaget. Användaren ställer upp kameran vinkelrätt med slagriktningen och applikationen upptäcker automatiskt slag och gör beräkningar som sedan visas upp på skärmen. Programmet är gratis för basfunktioner så som hastighetsmätning men har ytterligare i ett betalt läge.

Företaget påstår att deras applikation ger ”Accuracy for the everyday golfer” vilket begränsar deras löfte om precision. Egen erfarenhet vid användning gällande precision var varierad mycket på grund av att placeringen av kameran är mycket känslig. Applikationen använder sig också av en Iphone som endast klarar av att spela in i 240 FPS vilket kan bidra till lägre precision. [10]

#### **3.2 Garmin R10**

Garmins R10 är en bärbar dopplerradar enhet som specifikt har designats för att mäta olika parametrar på ett golfslag. Lik Trackman placeras den bakom bollen riktad mot det tänkta mål som bollen slås mot. Varje slag registreras automatiskt och data från slaget visas på en sammankopplad smartphone som kör en viss applikation.

Garmin själva säger att den parametern som är mest precis är bollhastigheten och visar i sitt produktblad att de beräknar precisionen till att vara plus-minus en miles per hour. Produkten släpptes under 2021 med ett pris motsvarande en bråkdel av trackman. [11]

#### **3.3 Yosuke Yamaguchi, Motoki Miura (2016)**

I “Real-time Analysis of Baseball Pitching using Image Processing on Smartphone” undersöker författarna möjlighet att mäta hastigheten på en kastad baseboll. De använder sig av olika metoder för datorsyn algoritmer för att finna en optimal metod att precist bestämma hastighet på bollen. De beskriver hur viktigt moment kastet är och hur träning med mätutrustning kan hjälpa spelare att utvecklas. Nuvarande metod utgår från en handhållen dopplerradar som måste vara placerad i linje med kastet men med deras metod kan de mäta kastet från en mycket bredare position. De använder sig av pixelsubtrahering som huvudmetod för att se skillnader mellan de inspelade bilder och med ytterligare filtrering kan de isolera bollens position för att genomföra sin analys.

De får relativt bra resultat från deras metod med de nämner att de var många kast där applikationen inte kunde identifiera bollen. Mycket på grund av att bakgrunden av videon

inte är uniform utan kan innehålla andra spelare som rör sig eller markeringar på planen med liknande färg som bollen. [12]

### **3.4 Jung et al (2010)**

I "Measurement of Initial Motion of a Flying Golf Ball with Multi-Exposure images for Screen-golf" föreslår författarna ett alternativ till den teknik som användes i simulatorgolf runt om i sydkorea. Artikel är skriven 2010 och vid denna tid använder sig simulatorer av sensorer på infrarött ljus för att analysera bollen och detta ska enligt dem ge lägre precision än kamerabaserat system. Deras metod bygger på att låta kameran ha något längre slutartid och istället ha ett snabbt blinkande LED eller stroboskop för att ta så kallat "multi-exposure image". På detta sätt kan de använda sig av en billigare kamera med lägre bilder per sekund och kan med hjälp av stroboskop lokalisera bollens position fler gånger i samma bild. Varje gång som bollen belyses av stroboskopet så reflekterar den ljuset till kamerans sensor och med en känd frekvens på stroboskopet kan författarna beräkna bollens position.

De skriver att metoden lämpar sig väl för en inomhusmiljö där flera kameror kan placeras ut för att kunna analysera bollens position i tre dimensioner och att de kan synkroniseras med ljuspulserna. [13]

## **4 Metod**

Detta kapitel beskriver den utrustning som används för att samla in data samt de moment som genomförts för att analysera denna data. Slutligen beskrivs hur denna dataanalys leder till ett resultat.

### **4.1 Kamera**

För att spela in videon på golfslaget behövs en kamera som kan ta många bilder per sekund. För detta har projektet riktat sig mot P40 pro som tillverkas av Huawei. Huvudanledningen till detta val var att den har den snabbaste inbyggda kameran vid projektets början med en möjlighet att fånga video i upp till 1920 FPS med en upplösning på 1280x720 pixlar under 0,5 sekunder. Vid högre FPS så finns mer data att analysera då bollen är med i fler bilder [7].

### **4.2 Kamerans position**

Kameran placeras på ett trebent stativ och blir då stationär. Positioneringen är sedan vinkelrätt med den tänkta slagriktningen i höjd med bollen. För enkelhetens skull placeras den på en standard driver klubblängds avstånd i från bollen, detta innebär att den placeras 45,5 tum (115,5 cm) från bollen.

### **4.3 Inspelning**

Kameran spelar in en video på antingen 1920x1080 upplösning med 960FPS eller 1280x960 upplösning i 1920FPS. Begränsning i hårdvaran tillåter bara inspelning under kortare tid och den varierar under de olika lägena. Vid 960fps klarar telefonen av att spela in en hel sekund medan vid det snabbare läget sjunker tiden till en halv sekund. Detta är mycket kort tid och kräver bra tajming hos den som manövrerar kamerautlösaren, speciellt i det snabbare läget. Båda lägena spelar in i 3 kanaler med 8 bitars upplösning där de tre kanalerna består av Röd Grön och Blå färg [7].

### **4.4 Utförande**

För att genomföra analysen av hastigheten måste en del förarbete ske för att förbereda och behandla videon.

#### **4.4.1 Importering av video**

Den inspelade videon sparas som en .mp4 videofil från smartphonen. Den laddas in ett program som placerar varje enskild bild i videofilen en array. Metoden som isolerar bygger på modulen imread som finns i openCVs bibliotek. För att göra projektet enklare så är det viktigt att videoklipppet trimmas så att så lite onödig information som möjligt kommer med.

Startpunkten på videon ska vara det ögonblick då bollen träffas och slutpunkt bör vara sista bilden som bollen är i bild.

#### 4.4.2 Bildbehandling

För att minska mängden data som ska bearbetas skärs bilden till i höjd, från vänster och underifrån. Detta då dessa områden inte används i analysen och tar endast upp beräkningstid. Då de flesta algoritmer som används i programmet utgår från bilder i gråskala konverteras videon med tre kanaler till en med en kanal som ytterligare reducerar storleken på indata och gör den enklare att analysera. Detta sker genom inbyggd metod i openCV. Sedan filtreras bilden med ett gaussian filter för att ytterligare reducera brus i bilden.

#### 4.5 Hastighetsmätning

Hastighetsmätningen sker genom analys av bilder i en video i ett antal steg.

##### 4.5.1 Objekterskiljning

Eftersom kameran hålls stationär och det som ska identifieras är i rörelse så används metoden för temporal differencing då den ansågs ge bäst resultat. Det som rör sig på bilden kommer att vara de pixlar som ändrar sig mellan de olika bilderna. Rörelsen fångas av två stegs absoluta skillnaden och sparas som en array.



(a) bild tid n



(b) bild tid n+1

Figur 3: Två bilder som jämförs





Figur 4: Absoluta skillnaden mellan bild (a) och (b)

Dessa bilder har ofta brus i sig då det lätt kan bli små skillnader mellan de olika bilderna och detta behöver filtreras och sedan genomgå thresholding. Alla förändringarna är inte intressanta att analysera, det kan vara objekt som klubba och spelaren som också fångas i jämförelsen. För detta sker en morfologisk operation på bilden som först eroderar bilden för att sedan genomför dilution på den. Detta sker med en kernel på 5x5 pixlar vars dimension har provats fram genom manuell testning. Denna process leder till att brus har reducerats och gör det möjligt att genomföra thresholding på bilden för att få en binärbild med de föremål som rör sig tydligt isolerade.



Figur 5: Thresholding har genomförts på ovanstående bild

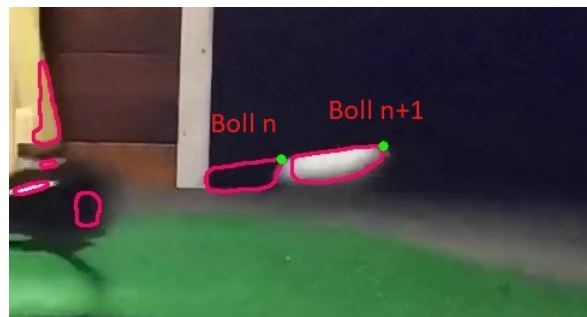
Denna binärbild processas av metoden findContours som finns i OpenCV bibliotek. Metoden använder sig av gaussian gradient för att identifiera kanter i bilden, vilka efter thresholding är mycket tydliga. Metoden samlar ihop sammanhängande kanter och returnerar en lista på de olika sammanhängande konturer den funnit. Listan består av koordinater för de olika punkter där en kant har påträffats i konturen och dessa koordinater blir grunden för hastighetsmätningen. Då bollens hastighet är hög blir dess kontur avlång, därför används bollens position längst fram i dess färdriktning som referenspunkt då den sidan har tydligast kontur i originalbilden.

#### 4.5.2 Misslyckad Objektskiljning

För att kunna särskilja bollen mot bakgrunden var en hypotes att man skulle kunna använda färgen av bollen som en unik egenskap. Genom detta skulle det då bli möjligt att isolera färgen som bollen fick på bild och sedan använda thresholding för att skapa en binärbild där det senare går att genomföra analys på. Användandet av färg eller HSV skala för att identifiera bollen misslyckades då för många objekt var i samma färg eller hade liknande HSV värden så antalet objekt att separera ut var för många.

#### 4.5.3 Hastighetsbedömningen

Den referenspunkt som togs fram för varje kontur är den som används för att bestämma hastigheten. Efter att bollen är träffad av klubban kommer bollens referenspunkt vara den punkt som ligger längst fram i färdriktningen, detta skulle vara närmast bildens högra kant om spelaren är en högerspelare. Det är på grund av att efter kontakt mellan boll och klubbhuvud kommer bollen sättas i högre hastighet än den hastighet som klubban har efter kontakt. Den referenspunkt som identifieras blir då bollens position i tid  $n+1$ . För att hitta bollens position i tid  $n$  så används den kända positionen för  $n+1$  och kunskapen om att bollen inte kan finnas framför, ovanför eller direkt under. Den måste finnas i motsatt färdriktning och på en något lägre höjd. Algoritmen använder sig av punkt  $n+1$  och den närmaste punkten som uppfyller kravet.

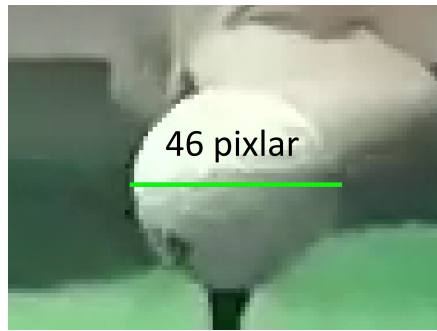


Figur 6: Identifierad kontur för bollens position i de två perioderna

Dessa punkter har en  $x,y$  koordinat och med Pythagoras sats kan avståndet mellan dem bestämmas. Med avståndet bollen färdas mellan varje bild kan en verklig hastighet senare beräknas.

#### 4.5.4 Beräkning av verklig hastighet

Då avståndet mellan bilder är känt kan den verkliga hastigheten räknas ut om det verkliga avstånd som varje pixel motsvarar tas fram. Då kameran placeras på samma avstånd från bollen vid varje slag kan detta fastställas som en konstant då vi vet bollens verkliga storlek och kan mäta hur stor den är i antal pixlar genom manuell analys av video, se figur 7.



Figur 7: Mått i pixlar genom manuell analys

Denna konstant kan också kallas mer beskrivande för pixlar/mm och tas fram genom formel.

$$\frac{\text{Objekets storlek i pixlar}}{\text{Objektets verkliga storlek i millimeter}} = \frac{\text{pixlar}}{\text{mm}} \quad (1)$$

Med avståndet per bild får vi reda på verkligt avstånd som bollen rör sig mellan varje bild.

$$\frac{\text{Beräknat avstånd}}{\text{Pixlar/mm}} = \frac{\text{mm}}{\text{bild}} \quad (2)$$

För att få fram hastigheten i meter per sekund använder vi antalet bilder per sekund eller FPS multiplicerat med 1000 för att göra om millimeter till meter.

$$\frac{\text{mm}}{\text{bild}} \times \text{FPS} \times 1000 = \frac{\text{meter}}{\text{sekund}} \quad (3)$$

För att sedan få ut rätt värde i miles per hour, som är den enhet som används i industrin, multipliceras meter/sekund med konstanten för konvertering till mph.

$$\text{m/s} \times 2,23693629 = \text{mph} \quad (4)$$

#### 4.6 Utförande av mätning

Då kamerans mycket korta inspelningstid kräver det att en person manövrerar kameran samtidigt som en annan utövar slaget. Den som manövrerar kameran väntar på att spelaren börjar sin baksving och når toppen för att vid detta tillfälle trycka på slutaren och starta inspelning. Tajmingen kräver lite tid att få till.

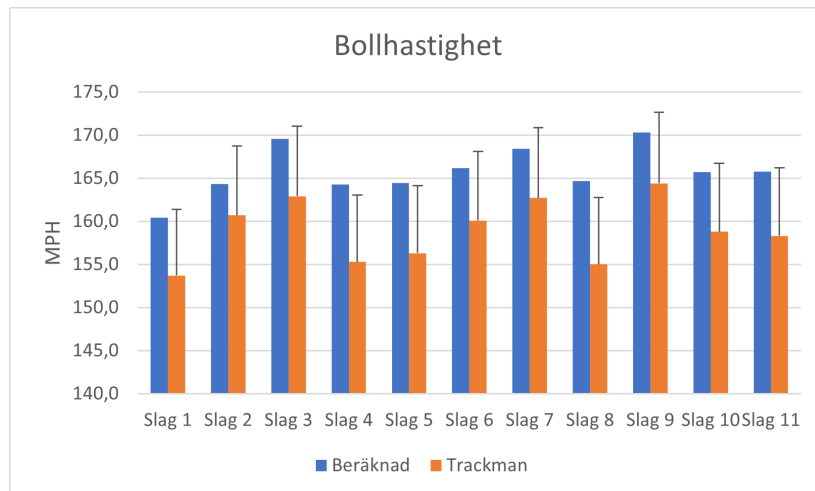
#### 4.7 Utrustning

Vid experimentet användes utöver den telefon för filmning en radar från tillverkaren Trackman, denna maskin har möjlighet att med hjälp av dopplerradar bestämma hastighet på bollen tillsammans med ett antal ytterligare parametrar. Då denna enhet kostar mer än vissa bilar var det lyckligtvis så att det gick att hyra tid med denna hos ett lokalt företag vid namn Innesvingen. Denna placeras direkt bakom bollen i rikningen som slaget sker i.

## 5 Resultat

Här sammanställs resultatet från det experiment som utförts.

### 5.1 Hastighetsmätning



Figur 8: Bollhastighet beräknad i programmet jämfört med Trackman. Stolparna visar 5% avvikelse från Trackman.

Den beräknade hastigheten jämförs med den hastighet som Trackman presenterar efter varje avslutat slag. På resultatet från Trackman placeras en pelare som visar ett femprocentig positivt fel intervall för att demonstrera avvikelse. Samtliga hastigheter som har beräknats fram av programmet har ett högre värde än det som Trackman presenterar. Programmet har en genomsnittlig felvisning på 4,35 procent. Då inte alla slag har samma utgångshastighet blir det intressant hur trenden mellan trackman och programmet skiljer sig.

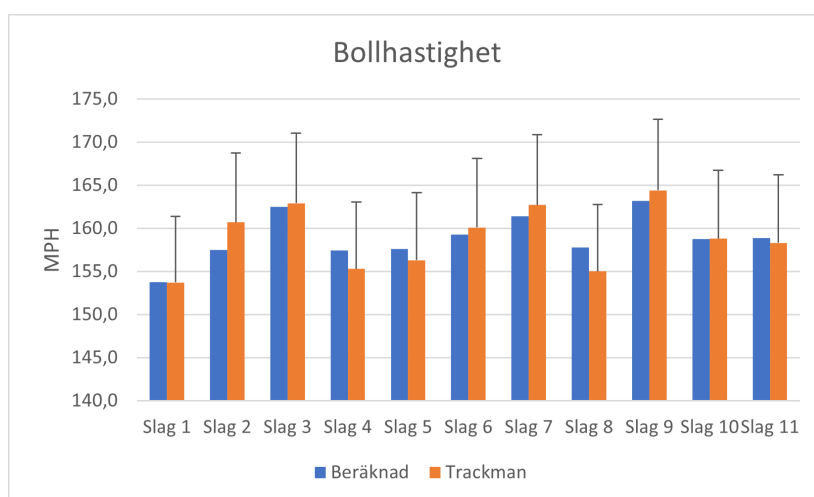
Slag	Beräknad hastighet	Trackman	Skillnad procent
Slag 1	160,7	153,7	4,55%
Slag 2	164,6	160,7	2,44%
Slag 3	169,8	162,9	4,25%
Slag 4	164,5	155,3	5,94%
Slag 5	164,7	156,3	5,39%
Slag 6	166,4	160,1	3,97%
Slag 7	168,7	162,7	3,68%
Slag 8	164,9	155	6,40%
Slag 9	170,6	164,4	3,75%
Slag 10	166,0	158,8	4,51%
Slag 11	166,1	158,3	4,90%
Avg			4,35%

Tabell 1: Resultat och fel i procent

Standardavvikelsen för felmätningen är 1,11 så även om hastigheten i sig är överskattad är skillnaden relativt konstant. Från detta resultat är det klar att det sker en konsekvent överskattning av den uppmätta bollhastigheten av programmet.

## 5.2 Felmätning

För att undersöka vad som kan ge bidrag till den överskattningen har olika värden på den manuella bollmätningen testats. Med den uppmätta storleken på bollen ändras från 46 pixlar till 48 pixlar får vi följande resultat.



Figur 9: Resultat med en större uppmätt storlek av bollen

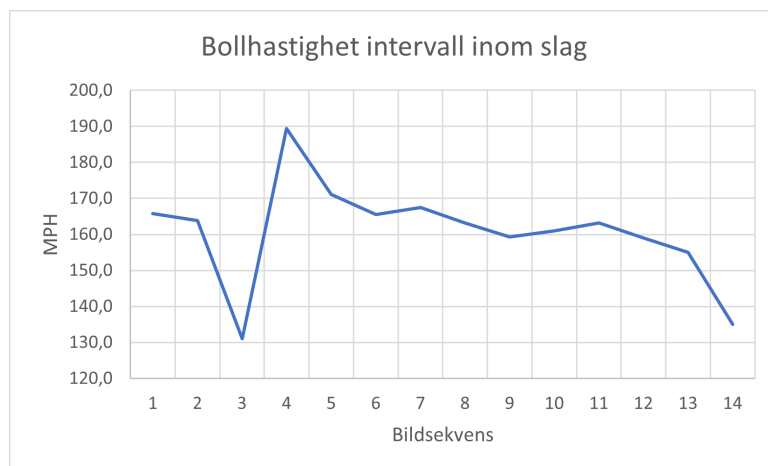
Slag	Beräknad hastighet	Trackman	Skillnad procent
Slag 1	153,7	153,7	0,03%
Slag 2	157,5	160,7	1,99%
Slag 3	162,5	162,9	0,26%
Slag 4	157,4	155,3	1,36%
Slag 5	157,6	156,3	0,83%
Slag 6	159,3	160,1	-0,53%
Slag 7	161,4	162,7	-0,80%
Slag 8	157,8	155	1,80%
Slag 9	163,2	164,4	-0,73%
Slag 10	158,8	158,8	-0,01%
Slag 11	158,9	158,3	0,37%
Avg			0,01

Tabell 2: Resultat efter justering av storleken på bollen i pixlar

Från tabell 2 ser vi att redan vid en ökning från 46 pixlar till 48 får vi ner felet till 0,01 procent och en standardavvikelse på 1,07. Eftersom storleken på bollen i pixlar används som en central del av algoritmen ger en förändring av dess värde en direkt ändring på resultatet. Ett större värde påverkar den beräknade hastigheten genom att ge ett lägre resultat.

### 5.3 Hastighetsförändring inom samma slag

Programmet mäter hastighet mellan varje sekvens av bilder genom att beräkna avstånd som bollen färdas. I figur 10 ser vi att avståndet varierar mellan bilderna i samma video. Detta beror i de flesta fall på en felaktigt identifierad position av bollen i ett fåtal av bilderna men att ett genomsnitt på de hastigheter som beräknas ger ett tillförlitligt resultat.



Figur 10: Den beräknade hastighet mellan varje sekvens i ett slag.

## 6 Diskussion

### 6.1 Begränsningar

Den största begränsningen för detta projekt är telefonen, där både hårdvara och mjukvara ligger till grund för detta påstående. I kameraläget på 1920 FPS kan kameran endast spela in i 0,5 sekunder vilket gjorde det till en för svår uppgift att konsekvent lyckas fånga slaget i bild. Därför fick 960 FPS-läget användas istället.

Den andra stora begränsningen som gjorde att projektet inte ledde till en mobil applikation som var målet var tillgången till kamerans API. I android får utvecklaren möjlighet att använda sig av olika API, Application programming interface, där tillverkare av varje telefon kan välja vilka kamerainställningar som ska vara tillgängliga. Den maximala FPS som går att få ut hos både Samsung och Huawei genom det, i Android, inbyggda camera2 API är 240FPS [17].

Det tillvägagångssätt som tillverkarna av telefoner tillåter användaren att spela in med högre hastighet är genom den inbyggda kamerapplikationen, något som de vid skrivandets stund inte har en öppen API för. I dokumentationen för nya Android 12 finner vi att de ska erbjuda fler inställningar genom camera2 API, men eftersom android 12 inte fanns tillgängligt under projektets tid kunde detta inte undersökas.

En stor begränsning med den metod att använda ett fast mått för pixlar/mm är att detta mått måste justeras om en annan smartphone används då linsen mellan de olika kamerorna inte nödvändigtvis har samma inzoomning.

### 6.2 Experiment

Eftersom de olika slag som spelats in slås av samma spelare i liknande hastighet har inte ett stort spektrum av hastigheter jämförts. Alla slag har en resulterande bollhastighet i ett relativt tätt intervall men i avgränsningarna skrevs det att hastigheter upp till 180 mph var önskvärt att undersöka. Under projektets tid hann inte en person som klara av att nå dessa hastigheter testas vilket innebär att dessa hastigheter inte har undersökts. Att felet på hastighetsmätningen blir större vid detta scenario är rimligt då en högre hastighet innebär kameran får svårare att hänga med men i vilken utsträckning är inte bevisat.

Högsta hastigheten som mättes upp i resultatet på 165mph kan jämföras mot hastigheter runt 200mph som professionella golfspelare når. Vid dessa farter är bollen suddigare och är synlig under färre bilder vilket kan göra att analysen fallerar. Vid mätning av lägre hastigheter blir hastighetsmätningen troligtvis mycket mer precis då bollens kanter blir mer tydliga och mindre estimering om dess verkliga position behöver göras.

### 6.3 Resultatanalys

Den standardavvikelsen som resultatet gav visar på en tydlig möjlighet att kunna mäta hastigheten på snabbt flygande föremål med hjälp av en mobiltelefons kamera. Hastigheten



som mättes upp av programmet är inom den felmarginal som låg som mål i projektet. Den hastighet som mäts upp är mycket lovande då den jämförs med Trackman. Variansen på resultatet är relativt liten men den har en konstant överskattning av hastighet.

Att hastigheten som mäts upp inom ett slag varierar var ett väntat resultat. Resultatet från detta är att även om det lätt blir ett visst fel vid identifiering av bollen i någon bildsekvens så blir genomsnittet av mätningarna nära det önskade resultatet. Det är därför som en högre FPS på kameran är önskvärt eftersom det, vid samma kameraplacering och slag, ger fler bilder på objektet i rörelse. Att ha fler bilder leder då till fler mätpunkter och mer tillförlitligt resultat.

En trolig felkälla är bollens uppmätta mått i pixlar som bestäms från bilden och har en inverkan i slutresultatet. Med kameran placerad på ett förutbestämt avstånd från bollen mäts golfbollens diameter till 46 pixlar men eftersom det är små detaljer som ska analyseras med begränsad upplösning kan exakthet bli lidande. Varje pixel i bilden har antingen en boll i sig eller inte vilket gör att exaktheten blir högst på en pixels nivå. Detta tillsammans med att objektet är litet gör att procentuellt påverkar denna binära representation av pixlar resultatet i högre grad än om det var ett större objekt på samma avstånd. Med det metodval som gjorts är en lösning att placera kameran närmre bollen så att den fyller större del av kamerabilden men då blir det område där kameran kan filma bollen i flykt mindre.

Att använda någon av mobilens andra linser och kameran sensorer övervägdes för uppmätning av bollens storlek då de i detta fall har bättre upplösning. Huvudkameran har en mycket högre upplösning men har inte möjlighet att ta tillräckligt många bilder per sekund så den kan bara användas för storleksbedömning på bollen. Detta skulle dock kräva mycket kalibrering då kamerans fysiska position skiljer sig mot den kamera som fångar slaget. Den kameran har också en annan lins vilket innebär en annan nivå av inzoomning som skulle försvåra bestämmandet av millimetrar per pixel.

## **6.4 Etik och miljö**

Uppsatsen berör ett ämne som inte direkt kan förknippas med några etiska konsekvenser. Miljöaspekten har inte heller en direkt inverkan.

## **6.5 Framtida förbättringar**

Då bestämning av bollens storlek mätt i pixlar är mycket viktig är det en ide att lägga tid på att förbättra det segmentet. Det skulle kunna lösas med en bättre kamera eller tänka igenom en bättre placering av kameran. Många fler slag skulle behövts spelas in för att undersöka felet vid olika hastigheter och dessutom identifiera om bollens mätta storlek har samma inverkan vid dessa.



# CHALMERS

## Referenser

- [1] Szeliski, R., 2011. Computer Vision. Texts in Computer Science. Available at: <http://dx.doi.org/10.1007/978-1-84882-935-0>.
- [2] Ramanath R., Drew M.S. (2021) Color Spaces. In: Ikeuchi K. (eds) Computer Vision. Springer, Cham. [https://doi.org/10.1007/978-3-030-63416-2\\_452](https://doi.org/10.1007/978-3-030-63416-2_452)
- [3] Elde J.H. (2021) Edge Detection. In: Ikeuchi K. (eds) Computer Vision. Springer, Cham. [https://doi.org/10.1007/978-3-030-63416-2\\_217](https://doi.org/10.1007/978-3-030-63416-2_217)
- [4] Learn openCV URL: <https://learnopencv.com/color-spaces-in-opencv-cpp-python/> (besökt 11/11/2021)
- [5] K. Liang, J. Wang and T. Zhao, "Variation temporal differencing for moving target detecting and tracking," 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, pp. 2380-2382, doi: 10.1109/ICECC.2011.6066359.
- [6] Trackman website URL: <https://blog.trackmangolf.com/full-ball-flight-tracking-provides-full-picture/> (besökt 05/12/2021)
- [7] HiSpeedCams website URL: <http://www.hispeedcams.com/p40-pro-super-slow-motion-extends-huaweis-reign/> (besökt 05/09/2021)
- [8] W. Shuigen, C. Zhen and D. Hua, Motion Detection Based on Temporal Difference Method and Optical Flow field," 2009 Second International Symposium on Electronic Commerce and Security, 2009, pp. 85-88, doi: 10.1109/ISECS.2009.62.
- [9] Z. Yu and Y. Chen, "A real-time motion detection algorithm for traffic monitoring systems based on consecutive temporal difference," 2009 7th Asian Control Conference, 2009, pp. 1594-1599.
- [10] Shotvision app URL: <http://www.shotvisionapp.com/> (besökt 2022-01-14)

- [11] Garmin r10 informationssida URL: <https://www.garmin.com/sv-SE/p/695391> (besökt 2022-01-14)
- [12] Yosuke Yamaguchi, Motoki Miura, Real-time Analysis of Baseball Pitching Using Image Processing on Smartphone,Procedia Computer Science, 2016, Pages 1059-1066.
- [13] Jaewook Jung, Hanbit Park, Shin Kang, Seungwoo Lee, & Minsoo Hahn, Measurement of initial motion of a flying golf ball with multi-exposure images for screen-golf,"IEEE Transactions on Consumer Electronics, 2010
- [14] Royal&Ancient rules URL: <https://www.randa.org/en/rules-of-equipment/2019/rules/equipment-rules/part-4-conformance-of-balls> (besökt 18/09/2021)
- [15] Mygolfspey titleist test URL: <https://mygolfspey.com/titleist-pro-v1-review/> (besökt 15/10/2021)
- [16] OpenCV license URL: <https://github.com/opencv/opencv/blob/4.4.0/LICENSE>
- [17] Android developer documentation URL: <https://developer.android.com/reference/android/hardware/camera2/pa> summary (besökt 02/12/2021)