



Počítačové vidění

Analýza hry stolního fotbalku

30. prosince 2016

Autoři: Adam Jež,
Tomáš Mlynarič,

xjezad00@stud.fit.vutbr.cz
xmlyna06@stud.fit.vutbr.cz

Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1	Úvod	2
2	Příprava	2
3	Detekce	3
3.1	Hrací plocha	3
3.2	Míček	3
3.3	Hráči	4
3.3.1	Nalezení pozice na tyčce	4
3.3.2	Nalezení pozice nohou	5
4	Analýza	6
4.1	Dotyk	6
4.2	Gól	6
5	Implementace a spuštění	7
5.1	Rozdělení práce	7
6	Vyhodnocení	7
6.1	Formát výsledků	7
6.2	Výsledky	8
7	Závěr	9
	Reference	10

1 Úvod

Cílem tohoto projektu je analyzovat hru stolního fotbalku. K úspěšné analýze je zapotřebí detekovat hráče a míček. Pozice ostatních objektů (hřiště, branky, tyčky), které nemění svou pozici, jsou manuálně zadány v konfiguraci, kterou je možné nadefinovat pro každé vstupní video samostatně. Vstupem projektu jsou videa pořízená kamerou, která byla připevněna do speciálně vytvořené konstrukce a malý stolní fotbalík (viz sekce 2). V sekci 3 jsou popsány způsoby detekce prvků fotbalíku, v sekci 4 pak analýza hry. Výstupem je video samotné s probíhající analýzou a soubor popisující události, které se vyskytly ve videu (viz sekce 6).

2 Příprava

Pro uskutečnění projektu bylo zapotřebí udělat několik kroků přípravy. Mezi těmito kroky bylo vytvoření konstrukce s fotbalíkem, která umožňovala vhodné umístění kamery, vzdálenost a osvětlení hrací plochy. Celá konstrukce je představena na obrázku 1. Nahrávky byly pořízeny fotoaparátem *Olympus PEN E-PL7*¹. Vstupní video bylo překonvertováno na nižší rozlišení pro rychlejší zpracování. Informace o vstupním videu jsou představeny v tabulce 1.

FPS	30 snímků/s
Rozlišení	1920 × 1080 px
Rozlišení po konverzi	960 × 540 px

Tabulka 1: Parametry vstupního videa



Obrázek 1: Konstrukce pro natáčení videa

¹Specifikace na https://www.olympus.cz/site/cs/c/cameras/pen_cameras/pen/e_pl7/e_pl7_specifications.html

3 Detekce

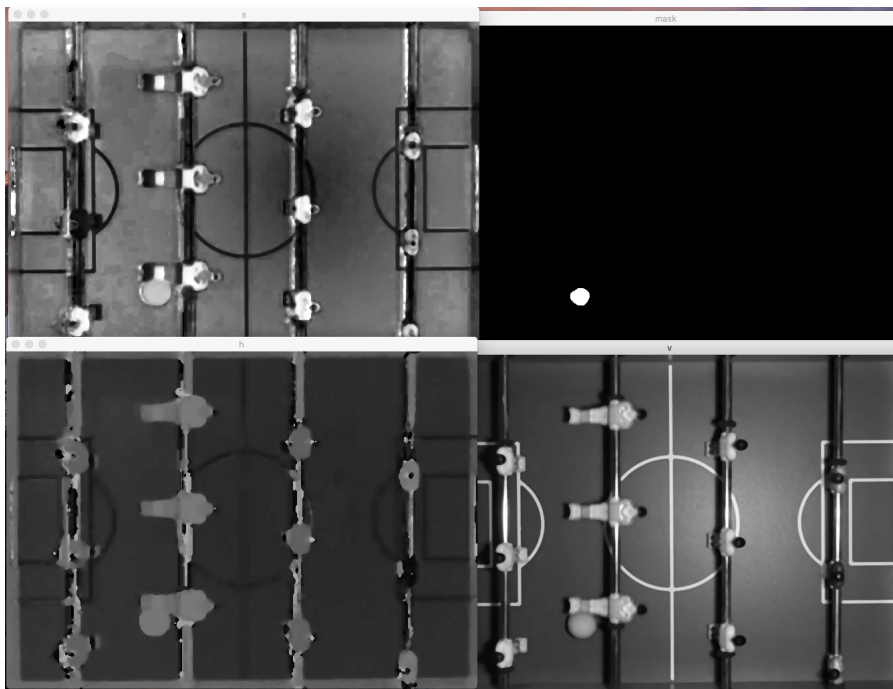
Hlavní částí projektu je detekce několika oddělených částí fotbalu a také událostí ve hře.

3.1 Hrací plocha

Hrací plocha, tj. plocha, ve které se může pohybovat míček je staticky vybrána výřezem ze vstupního videa. Tímto způsobem jsou také vybrány branky a tyčky, na kterých se nacházejí hráči. Ohraničení plochy je definováno staticky. Jelikož ale různá vstupní videa byla pořízena s trochu odlišným nastavením, bylo umožněno nastavit tyto statické údaje pro každé video samostatně pomocí konfiguračních souborů.

3.2 Míček

Jednou z hlavní částí detekce hry bylo nalezení pozice míčku ve hře. Pro detekci míčku byla zvolena technika segmentace pomocí barvy. Proto bylo potřeba, aby míček byl co nejvíce kontrastní vůči ostatním částem hrací plochy. Po odzkoušení několika dostupných míčků byl zvolen kontrastní červený míček. Pro segmentaci míčku je obraz nejprve převeden do *HSV* barevného prostoru a rozmazán pomocí mediánového filtru. Následně je pomocí prahování (funkce *cv2::inRange*) získána maska míčku. Práh je zvolen na základě barvy míčku a přidána tolerance horní a dolní hranice barvy. Po aplikaci prahování bylo potřeba odstranit nežádoucí elementy, čehož bylo dosaženo pomocí morfologické transformace otevření (eroze následovaná dilatací) [2]. Výsledek tohoto kroku je znázorněn na obrázku 2.



Obrázek 2: Získání masky míčku

Dalším krokem detekce míčku bylo vyhledání kontur v obraze. Na nalezené kontury bylo aplikováno několik filtračních metod, aby se zamezilo zvolení nesprávného objektu [1] [4]. Nejprve se zahodily objekty, kterých plocha byla menší než zvolený limit, dále se pro konturu vytvořila opsaná kružnice a zahodily objekty, které nesplňovaly limit velikosti kružnice. Posledním krokem bylo porovnání nalezených objektů vůči předem vytvořené šabloně míčku (kruh známé velikosti převeden na konturu) a zvolení nejlépe odpovídajícího. K reprezentaci míčku se používá střed opsané kružnice a poloměr míčku, který je znám předem. Výsledek je představen na obrázku 3.



Obrázek 3: Nalezení kontury míčku

3.3 Hráči

Další důležitou součástí detekce jsou samotní hráči (panáci) umístění na tyčce. Pozice tyček není součástí detekce a je tedy zadána staticky v konfiguraci daného videa. Vzdálenost mezi panáky na jedné tyčce se nemůže měnit a jejich natočení je identické. Dále je pevně zadán počet hráčů na tyčce. Tyto apriorní informace jsou při detekci využity.

3.3.1 Nalezení pozice na tyčce

První částí detekce je nalezení přesné pozice panáka na tyčce. Pro každou tyčku se vytvoří výřez, který obsahuje celou délku tyčky s danou šířkou, která je zvolena tak, aby výřez vždy obsahoval celé tělo hráče. Tělo hráče je zvoleno z důvodu jeho kontrastní barvy viz. 4 (červená a žlutá). Menší komplikací jsou brankáři, kteří mají od ostatních hráčů jiné barvy.

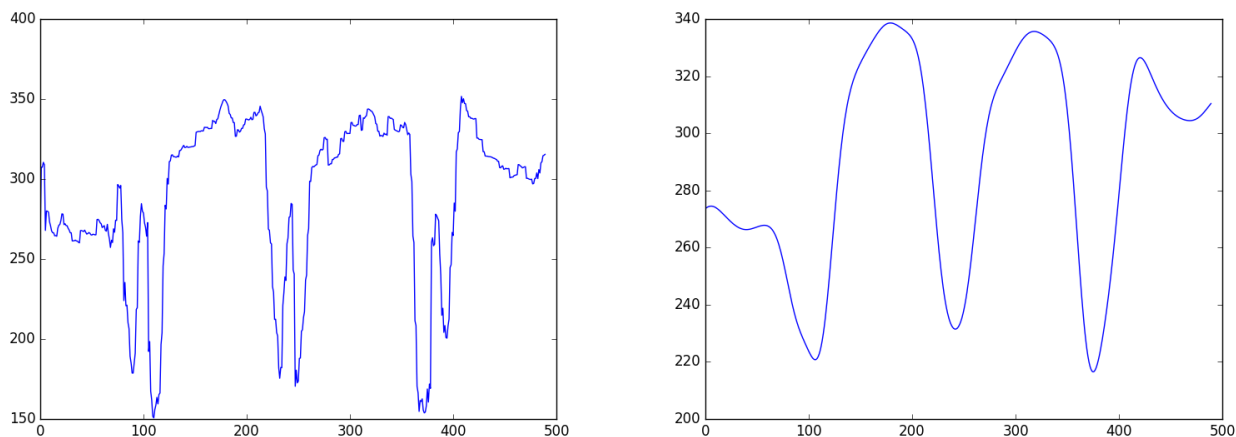


Obrázek 4: Výřez tyčky pro detekci pozice hráče

Před samotným výpočtem je výřez převeden do *HSV* barevného prostoru a při výpočtu je zanedbána složka *Value*, což zajistí větší odolnost při výpočtu rozdílu s barvou těla. Algoritmus pro nalezení pozice nejprve vezme každý řádek výřezu (řádek je kolmý k samotné tyčce). Pro každý takový řádek pixelů se vypočítá střední kvadratická odchylka (anglicky *mean squared error*) s barvou těla hráče (barvy hráčů jsou zadány staticky v konfiguraci). Výsledkem je hodnota rozdílu s danou barvou pro každý řádek. Výsledné hodnoty jsou dosti zašuměné, což je způsobeno odlesky a pohybem hráčů. Proto hodnoty zkonvolujeme s gaussovským jádrem, čímž dostáváme dobře rozlišitelné minima, ve kterých se nachází pozice hráčů (viz obrázek 5). Při konvoluci jsou také vyhlazeny špičky, které jsou způsobené odlišnou barvou hlavy od těla.

Následující technika nalezení minima je inspirována technikou prezentovanou v [3]. Aby bylo hledání minima robustnější, využíváme znalosti o daném počtu hráčů N na tyčce a o fixní vzdálenosti D mezi hráči. To nám umožňuje vzít paralelně N bodů, které jsou od sebe vzdáleny D pixelů. Proto se zaměřujeme na nalezení minima pomocí následující funkce:

$$f'(i) = \sum_{n=0}^{N-1} f(i + n * D) \quad (1)$$



Obrázek 5: Vlevo: vertikální rozdíl s barvou těla hráčů, vpravo: vertikální rozdíl po konvoluci s gaussovským jádrem.

Odhadovaná pozice panáka je ještě zpřesněna s použitím algoritmu *Hill Climbing*. Vstupem algoritmu je odhadovaná pozice získána pomocí uvedeného nalezení minima. Poté je prohledáván prostor sousedních hodnot. Algoritmus končí při nalezení lokálního minima.

3.3.2 Nalezení pozice nohou

Pro správnou detekci dotyku hráče s míčkem je nutné znát přesnou pozici nohou, která se může od pozice těla na tyčce hodně lišit. Detekce nohou probíhá do jisté míry podobně jako detekce těla, s tím rozdílem, že u detekce těla byla brána v potaz barva těla a u nohou je bráno v potaz bílé zakončení postavičky. Nejprve se vezme výřez, který je dán pozicí těla a maximální šířkou hráče. Následně se výřez předzpracuje:

1. Převedení do tónu šedi,
2. Inverzní binární segmentace, která má za úkol segmentaci bílé barvy,
3. Morfologická operace otevření, která má za úkol opravení nedokonalé segmentace bílé barvy.

Následuje sečtení všech před zpracovaných výřezů, patřících na jednu tyčku, do jednoho, což by mělo metodu trochu zrobustnit. Jednotlivé sloupce ve výsledném výřezu jsou sečteny do jedné hodnoty. Výsledkem je seznam hodnot korespondující k sloupcům výřezů, který představuje pravděpodobnost, že v daném sloupci je bílá noha. Následuje nalezení nejvzdálenější minima od středu těla (jelikož červení hráči mají bílé trenýrky, které jsou ale blíže tělu). Jelikož noha hráče může být zakryta jeho tělem, bylo potřeba určit práh, při kterém není noha detekována. Tento práh byl určen empiricky. Jestliže žádná hodnota nespadá do určeného prahu, je pozice nohy ve stejném místě jako pozice těla.



Obrázek 6: Příklad finální detekce pozice hráče na tyčce a detekce nohou

4 Analýza

Po detekci je nutné analyzovat průběh hry s využitím detekovaných součástí. Jako základní úloha analýzy bylo vybráno zjištění události dotyku hráče s míčkem a vstřelení gólu. Z těchto dvou události lze dále získávat informace o střelci gólu nebo o asistencích.

4.1 Dotyk

Pro detekci dotyku hráče s míčkem je využita jednoduchá eukleidovská vzdálenost. Jelikož známe pozici nohou všech hráčů a přesnou pozici míčku, můžeme mezi nimi spočítat eukleidovskou vzdálenost. Následně zkontrolujeme, jestli nejbližší hráč je dostatečně blízko, neboli jestli jeho vzdálenost je menší než určitá tolerance. Tuto toleranci jde v konfiguračním souboru měnit a lze tak hledat ideální poměr mezi *True Positive* a *False Positive* alarmem.

Uvažováno bylo taky o vylepšení této detekce, kdy u míčku bychom měli informace o aktuální rychlosti a směru jeho pohybu. Při signifikantní změně jednoho z parametrů bychom předpokládali dotyk hráče. Takové to vylepšení by však moc dobře nefungovalo u mantinelů hřiště, kde se míček odrazí a nemusí se dotknout hráče, i když hráč může být dostatečně blízko pro detekci.

4.2 Gól

Prvním krokem detekce gólu bylo vytvoření masky hrací plochy s označenými místy branek. Dále bylo potřeba porovnat, zda nejlevější bod míčku se nachází v oblasti levé branky (nápodobně pro pravou stranu) a pokud ano, pak bylo potřeba zjistit zda míček spadne do branky. K této detekci bylo potřeba používat historii, pro kterou byly vytvořeny 3 cyklické buffery. První buffer slouží pro zjišťování, zda je míček uvnitř oblasti branky a druhý, zda se míček nachází ve hřišti. Detekce gólu tedy probíhá podle algoritmu popsaného níže:

```
if míček v bufferu oblasti branky and not míček v bufferu hřiště then
|   Gól padl
else
|   Není gól
end
```

V tabulce 2 je představena situace, pro kterou nastala gólová situace. V prvních 3 řádcích je znázorněna situace před gólem, ve 4. řádku pak výše zmíněná podmínka platí a tedy padl gól.

Míček v hřišti	Míček v oblasti branky
[0 0 1 1 0]	[0 0 0 0 0 0 0 0 0 0]
[0 0 0 1 0]	[0 0 1 1 0 0 0 0 0 0]
[0 0 0 1 0]	[0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0]	[0 0 1 1 0 0 0 0 0 0]
[0 0 0 0 0]	[0 0 0 0 0 0 0 0 0 0]

Tabulka 2: Buffery pro detekci gólu

5 Implementace a spuštění

Program byl napsán v programovacím jazyce *Python 3* a pro práci s obrazem použity knihovny *Numpy* a *OpenCV*. Spuštění je možné z příkazové řádky s volbou těchto parametrů:

```
./POV.py -v cesta-k-video
```

Při spuštění se načítá soubor stejného jména, jako vstupní video s koncovkou `.json`, který umožňuje předefinovat výchozí nastavení (je možné upravit např. souřadnice tyček, branek, hrací plochu, ale i barvu míčku, barvy hráčů atd.). Po ukončení videa je vygenerován soubor se stejným názvem jako vstupní video, ale s příponou `_result.txt`. V tomto souboru jsou zapsány události detekované ve hře (viz sekce 6).

Pro ukázkou vstupů byla vytvořena testovací sada, kterou je možné získat po spuštění skriptu `./samples.sh`, který stáhne nahrávky, anotace a konfigurace na disk do složky `./Samples`.

5.1 Rozdělení práce

- Adam Jež – Detekce hráče, analýza doteků, skript pro ověření výsledku, anotování.
- Tomáš Mlynarič – Příprava (nahrání, střih, úprava videa + konstrukce), detekce míčku, analýza gólů, anotování.

6 Vyhodnocení

Pro vyhodnocení vytvořeného detektoru bylo potřeba vytvořit video záznam ze hry a následně zanotovat celý průběh hry krok po kroku. Zanotovaný průběh hry byl následně porovnán s výstupem programu, tedy seznamem událostí detekovaných ve hře.

6.1 Formát výsledků

Pro zaznamenávání událostí byl zvolen následující formát:

```
%H:%M:%S.%f EVENT_TYPE EVENT_INFO
```

kde na začátku se nachází časová značka následovaná `EVENT_TYPE`, což je jeden ze dvou typů událostí: dotek (`TOUCH`) a gól (`GOAL`). Událost doteku obsahuje v `EVENT_INFO` identifikátor hráče, který se dotkl. Při střelení gólu `EVENT_INFO` obsahuje identifikátor strany, kde padl gól.

Vytvořený skript pro ověření vytvořeného detektoru zkouší napasovat shodné události k sobě. Pracuje s pevně danou časovou tolerancí, do které se shodující se události musí vejít. Výstupem skriptu je počet shodujících se událostí a jejich časový odstup, ale jelikož už při anotování mohla vzniknout časová chyba (v milisekundách), není na tuto hodnotu brán větší zřetel. Dále skript zobrazí, kolik událostí nebylo detektorem zachyceno a naopak, kolik událostí, které se ve skutečnosti nestaly, bylo mylně přidáno.

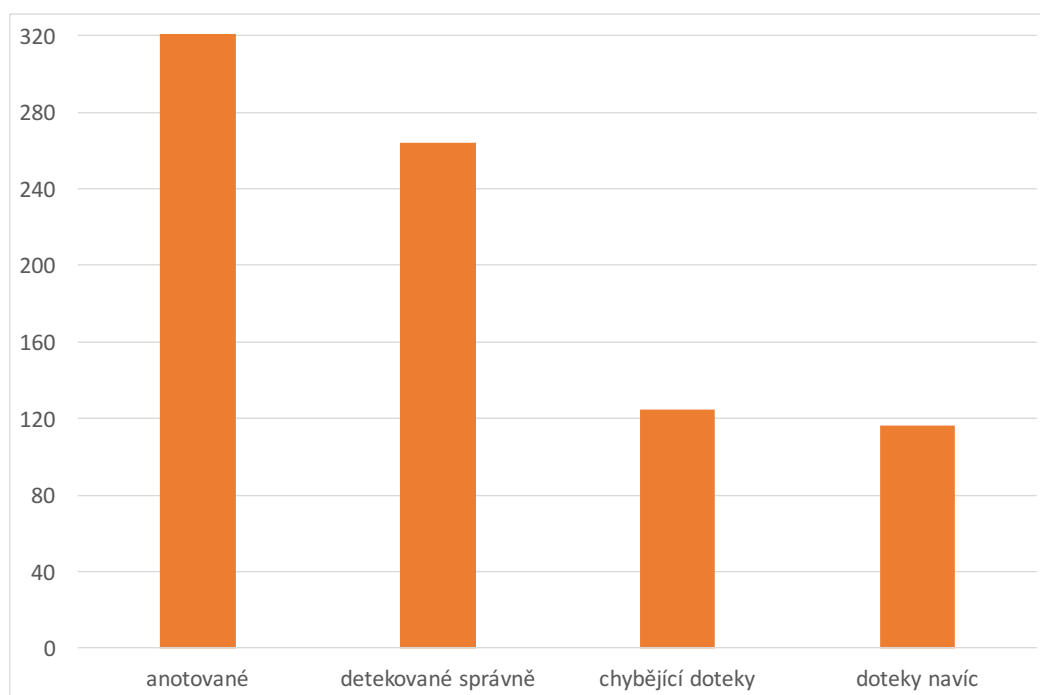
6.2 Výsledky

Pro výsledné vyhodnocení bylo analyzováno **6.88** minut záznamu s celkovým počtem 407 událostí. Jednotlivé výsledky her jsou zobrazeny v tabulkách níže, tedy počty událostí doteků v tabulce 3 a v tabulce 4 počty událostí gólů. V obou tabulkách sloupce představují jednotlivé vstupní záznamy her, řádky pak popisují počty událostí pro danou hru.

	game_00	game_01	game_02	game_03	game_04	game_05
anotované	44	86	62	65	62	70
správně	37	53	45	47	41	41
chybějící	7	33	17	18	21	29
navíc	8	22	16	34	12	24

Tabulka 3: Výsledky vyhodnocení doteků pro testované záznamy

Na obrázku 7 je znázorněná úspěšnost doteků pro všechny hry dohromady. Je možné vidět, že z celkových 389 anotovaných doteků jich bylo úspěšně detekováno 264, což je **67.87%** úspěšnost. Bylo detekováno o 116 doteků navíc a 125 doteků naopak nebylo nalezeno. Doteky navíc byly způsobeny tím, že míček projel velmi blízko detekovaných nohou, naopak nenalezení dotyku bylo nejčastěji způsobeno nesprávnou detekcí hráčových nohou případně momentálním nenalezením míčku v hrací ploše.

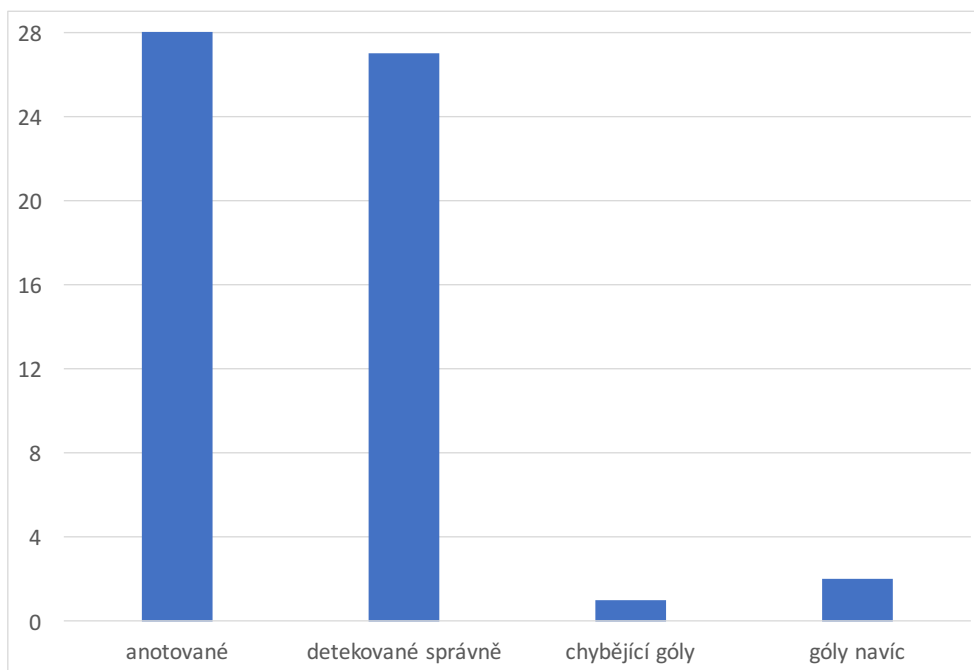


Obrázek 7: Vyhodnocení doteků

Nápodobně na obrázku 8 je možné vidět vyhodnocení gólů. Z celkových 28 anotovaných gólů bylo detekováno správně 27 gólů, nicméně byly detekovány 2 góly navíc a 1 scházel. Ze zmíněných hodnot vyplývá **96.4%** úspěšná detekce gólu. Neúspěšnost detekce gólu byla způsobena občasným nenalezením míčku v hřišti během jeho výskytu v brankové oblasti, nebo naopak nezaznamenání míčku v oblasti branky (při velké rychlosti míčku).

	game_00	game_01	game_02	game_03	game_04	game_05
anotované	4	6	4	5	3	6
správně	4	6	4	5	2	6
chybějící	0	0	0	0	1	0
navíc	0	0	0	1	1	0

Tabulka 4: Výsledky vyhodnocení gólů pro testované záznamy



Obrázek 8: Vyhodnocení gólů

7 Závěr

Cílem tohoto projektu bylo navrhnout a implementovat aplikaci, která dokáže snímat hru stolního fotbalu a detekovat části nutné pro analýzu probíhané hry. Výsledná aplikace umožňuje analyzovat již předem nahrané videa hry. Konfigurace pro dané video ze hry je dynamicky načítána z konfiguračního souboru ve formátu *JSON*. Průběh analýzy je uživateli prezentován v přehledné formě a výslednou úspěšnost detektoru lze ověřit pomocí skriptu na porovnání se souborem obsahující manuálně anotované události.

Největším defektem byla analýza doteků, tudíž detekce nohou hráče způsobené nevhodnou barvou hráčů a tedy obtížnějším zpracováním. V rámci zkoušení různých metod detekce byl pokus o detekci pomocí *template matching*, nicméně nepodařilo se dojít k vhodným a tedy lepším výsledkům, než v případě zmíněné metody.

V rámci rozšíření aplikace by bylo možné přidání komplexnější analýzy hry, jako je nejčastější cesta míčku do branky či případně, jak dlouho každý hráč držel míč. Bylo by také možné natrénovat inteligentnější detekci hráčů, aby úspěšnost nalezených událostí byla co největší.

Reference

- [1] OpenCV. Contours features [online]. http://docs.opencv.org/trunk/dd/d49/tutorial_py_contour_features.html, [cit. 2016-12-28].
- [2] OpenCV. Morphological Transformations [online]. http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html, [cit. 2016-12-28].
- [3] Thilo Weigel, Dapeng Zhang, Klaus Rebert, and Bernhard Nebel. Adaptive vision for playing table soccer. Technical report, Institut für Informatik, 2004.
- [4] Španěl Michal, Beran Vítězslav. Obrazové segmentační techniky [online]. http://www.fit.vutbr.cz/~spanel/segmentace/#_Toc125769327, [cit. 2016-12-30].