



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ZOBRAZENÍ VIRTUÁLNÍ GRAFIKY A ANIMACÍ PRO
SPORTOVNÍ ANALÝZU**

DISPLAYING VIRTUAL GRAPHICS AND ANIMATIONS FOR SPORTS ANALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL MILOSLAV OČENÁŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. BERAN VÍTĚZSLAV, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Očenáš Daniel Miloslav**
Program: Informační technologie
Název: **Zobrazení virtuální grafiky a animací pro sportovní analýzu**
Displaying Virtual Graphics and Animations for Sports Analysis
Kategorie: Počítačová grafika

Zadání:

1. Prostudujte metody vhodné pro renderování virtuálních animací a 3D grafiky do videa.
2. Navrhněte vhodný mechanismus, jak efektivně renderovat virtuální 3D grafiku do video sekvencí, které obsahují kalibraci scény. Navržená metoda bude využívat dostupné video sekvence s definovanými projekcemi virtuální kamery pro každý snímek.
3. Implementujte řešení podle navržené metody. Očekávaným výstupem je sada funkcí a/nebo nástrojů, pomocí kterých lze vkládat do videa virtuální grafiku.
4. Připravte demonstrační video a ověřte funkčnost na různých typech sportovních záznamů a navrhněte možná vylepšení.

Literatura:

- M. Sonka, V. Hlaváč, R. Boyle. *Image Processing, Analysis, and Machine Vision*, CL-Engineering, ISBN-13: 978-0495082521, 2007.
- Gary R. Bradski, Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*, ISBN 10: 0-596-51613-4, September 2008.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a částečně bod 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Beran Vítězslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Cielom bakalárskej práce je vytvoriť súbor nástrojov použitých pri analýze videozáznamov športových podujatí, predovšetkým futbalových zápasov. Užívateľ môže nástroje použiť na zvýraznenie vybraných hráčov, ktorí sú ďalej trackovaní a zvýrazňovaní. Ďalej manuálne vytvárať smerovníky, znázorňujúce smer pohybu hráča alebo lopty. Trojrozmerné objekty sú vytvárané perspektívnym skreslením korešpondujúcim so scénou, ktorá je zachytená kamerou. Teoretická časť bakalárskej práce uvádza prehľad odbornej literatúry potrebnej na správne zobrazenie grafických objektov do videa v duchu rozšírenej reality. Praktická časť práce je zameraná na implementáciu a vizualizáciu grafických nástrojov s použitím knižnice OpenGL.

Abstract

The aim of this bachelor thesis is to create graphic tools used for the analysis of video recordings of sports events, especially football matches. The user can use the tools to highlight selected football players, which are further tracked and highlighted with three-dimensional objects rendered into video scene. Secondly, user can manually create pointed arrows representing player or football movement. Three-dimensional objects are created by perspective distortion corresponding with the scene taken by camera. The theoretical part focuses on the proper rendering of graphical objects into calibrated scene. The practical part focuses on implementation and visualization of mentioned tools by using OpenGL library.

Kľúčové slová

virtuálna grafika, 3D grafika, rozšírená realita, OpenGL, animácie, futbalová analýza, počítačové videnie, spracovanie obrazu

Keywords

virtual graphics, 3D graphics, augmented reality, OpenGL, animations, soccer analysis, computer vision, image processing

Citácia

OČENÁŠ, Daniel Miloslav. *Zobrazení virtuální grafiky a animací pro sportovní analýzu*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Beran Vítězslav, Ph.D.

Zobrazení virtuální grafiky a animací pro sportovní analýzu

Vyhlásenie

Vyhlasujem, že som svoju bakalársku prácu vypracoval samostatne pod vedením Ing. Vítězslava Berana, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Daniel Miloslav Očenáš
29. júla 2021

Podakovanie

Ďakujem školiteľovi Ing. Vítězslavovi Beranovi, Ph.D., za poskytnuté rady a trpezlivosť pri vedení bakalárskej práce. Veľká vďaka patrí aj mojej rodine a blízkym za nekonečnú podporu.

Obsah

1	Úvod	2
2	Teoretická časť	4
2.1	Náhľad do problematiky	4
2.2	Zobrazovanie grafických objektov	6
2.3	Detekcia hráčov	7
2.4	Homogénne súradnice	9
2.5	Model kamery	9
2.6	Odhad pozície kamery	12
2.7	Homografia	12
3	Návrh nástrojov a aplikácie	14
3.1	Popis scény	14
3.2	Zvýraznenie hráčov	17
3.3	Orientovaný smerovník	18
3.4	Ofsajdová čiara	19
3.5	Návrh aplikácie	19
4	Demonštračná aplikácia	21
4.1	Použitie technológie	21
4.2	Dôležité triedy	23
4.3	Ukážka implementácie virtuálnej grafiky	29
4.4	Užívateľské rozhranie	31
4.5	Testovanie a výsledky	32
5	Záver	33
	Literatúra	35
A	Obsah priloženého pamäťového média	37
B	Postup inštalácie a spustenia	38

Kapitola 1

Úvod

V 19. storočí došlo k veľkému množstvu spoločenských zmien a k zásadným technickým objavom, napr. parovozu čiže parného rušňa, šijacieho stroja, telegrafu, telefónu, semaforu, fotografie, gramofónu alebo plniaceho pera. Nasledujúce 20. storočie je neodmysliteľne spojené s počítačmi a ich vývojom od počítačích strojov po univerzálne zariadenia na automatické spracovanie a uchovávanie informácií. V roku 2021 uplynulo už 83 rokov od skonštruovania prvého samočinného počítača (Konrad Zuse v Nemecku, 1938), ktorý pracoval s číslami v dvojkovej sústave. Počas tohto obdobia, ktoré zodpovedá štandardnej dĺžke života jedného človeka, sa technika a technologické postupy posunuli závažnou rýchlosťou vpred. Na prelome sedemdesiatych a osemdesiatych rokov 20. storočia uzrel svetlo sveta prvý osobný počítač (Personal Computer, IBM PC, 1981) a vtedy sa začína aj ohromný boom využívania počítačov vo všetkých oblastiach života človeka. Dnes si už ani nedokážeme predstaviť život a prácu bez počítačových, resp. operačných systémov, ktoré sa uplatňujú v nástupcoch počítačov – bez mobilných telefónov s operačným systémom, t. j. smartfónov umožňujúcich okrem hlasovej či textovej komunikácie prístup na internet a používanie rôznych ďalších aplikácií (napr. lokalizáciu a navigáciu – globálny lokalizačný systém (GPS z angl. Global Positioning System), tak ako ani bez streamovacích médií a pod.

V čase počítačov prvých generácií analyzovali športoví tréneri alebo novinári výkony športovcov alebo ich kolektívnu hru (futbal, hokej a pod.) iba verbálne, prípadne s pomocou tabule či papiera s fixkou, aby podrobnejšie vysvetlili jednotlivé herné situácie, pozície a odôvodnenie úspechov a víťazstiev, ale aj nedostatkov alebo prehiev, teda silné aj slabé stránky športovcov a ich výkonov. V 21. storočí im v tom veľmi efektívne aj efektne pomáha virtuálna grafika.

Samotný výber témy bakalárskej práce *Zobrazenie virtuálnej grafiky a animácií pre športovú analýzu* podmienili dva dôvody. Prvý je subjektívny a druhý študijný. Umelecká grafika a oblasť virtuálnej grafiky a animácií ma silno zaujali už v rodinnom prostredí a ako aktívneho športovca – plavca a cyklistu, ale aj fanúšika športových prenosov v televízii. Ako študent študijného programu informačné technológie si môžem na atraktívnej, a nie jednoduchej téme skúsiť a overiť samostatne vytvoriť, organizovať, realizovať a prezentovať nadobudnuté vedomosti a skúsenosti. Zvolená téma však bola pre mňa tiež neľahkou výzvou, pretože vyžadovala aj štúdium ďalších nových informácií a ovládnuť nové vývojové prostredia.

Cielom bakalárskej práce je vytvoriť grafické nástroje pre športovú analýzu v duchu rozšírenej reality s využitím virtuálnej grafiky a animácií. Nástroje nachádzajú využitie predovšetkým pri pozápasovej analýze športového podujatia. Športový tréner alebo tímový videoanalytik môže využiť virtuálnu grafiku pre zvýraznenie situácií, ktoré sa udiali počas

zápasu. Predpokladom použitia grafických nástrojov je panoramatické video športového ihriska obdĺžnikového tvaru, kde sú v rámci videosnímkov viditeľné všetky rohy ihriska. Inšpirácia pre navrhované grafické nástroje vznikla v spoločnosti Camvision, s. r. o., ktorá sa zaoberá športovou analýzou s využitím kamerového systému vytvárajúcim požadovaný videozáznam. Vytvorenými nástrojmi, ktoré používajú prvky virtuálnej grafiky, bude možné modelovať taktiku tímu, zahŕňajúc rozmiestnenie hráčov na ihrisku, smer pohybu hráčov a prihrávkov a môžu byť použité aj na analýzu činnosti jednotlivca. Modelovanie možných situácií a analýza odohraných zápasov vedie k zlepšeniu výkonu športovcov. Použitie virtuálnej grafiky, ktorá zobrazuje modelované situácie v rámci rozšírenej reality, umožňuje lepšie porozumenie zo strany športovca, ale aj korektnejšie vytváranie daných situácií. Takisto to má mimoriadny význam pri vizualizácii stratégie hry počas taktickej prípravy hráčov, pretože hráčom ušetrí čas pri teoretickej príprave a vizualizácie herných situácií uľahčujú ich pochopenie. Okrem iného je používanie virtuálnej grafiky interaktívne a zaujímavejšie. Primárnym cieľom týchto nástrojov je vizualizácia modelovanej situácie, nie podrobná analýza a zber štatistík o podujatí. Na naplnenie tohto cieľa je na demonštráciu použitia grafických nástrojov v bakalárskej práci potrebná znalosť o situácií, ktorá sa odohráva vo videozázname a preto sa časť práce bude venovať aj detekcii hráčov v športových zápasoch a nastaveniu kamery pre virtuálnu scénu.

Kapitola 2

Teoretická časť

V teoretickej časti bakalárskej práce sa rieši problematika interaktívnej grafickej analýzy športových podujatí a grafických nástrojov, ktoré môžu byť na takúto analýzu využité. Jednotlivé kapitoly sú preto venované vhodným metódam na zobrazenie virtuálnej grafiky. Pozornosť sa tiež venuje nastaveniu zobrazenia virtuálnej grafiky v rámci rozšírenej reality, kde teoretické znalosti dierkového „pinhole“ modelu kamery hrajú dôležitú úlohu (na zobrazenie virtuálnej grafiky v rámci rozšírenej reality musí byť nastavenie virtuálnej kamery zhodné s kamerou, ktorá sníma danú scénu). Záver teoretickej časti patrí Phongovmu osvetľovaciemu modelu, ktorý sa zaoberá výpočtom osvetlenia grafických objektov.

2.1 Náhľad do problematiky

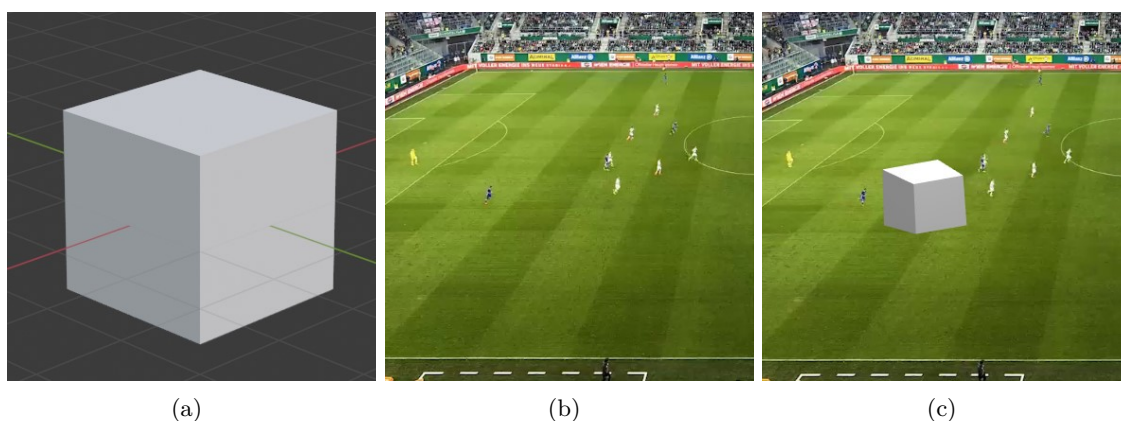
Slovom **grafika** v pôvodnom význame pomenúvame vo všeobecnosti veľmi staré „výtvarné umenie zahŕňajúce kresbové umelecké reprodukčné techniky“ [2]. Základným médiom grafiky bol od jej vzniku papier. Novšie významy slova grafika zahŕňajú už techniku tvorby a spracovania grafiky pomocou počítača, resp. „grafické zobrazovanie informácií na výstupných zariadeniach počítača (na obrazovke, tlačiarňou a pod.) akéhokoľvek druhu a na akýkoľvek účel“ [2]. V tomto význame už ide o počítačovú grafiku ako samostatnú kategóriu grafiky a odbor informatiky. Jej využitie je veľmi široké od sprievodných obrázkov a ilustrácií k textovej informácii (grafy, schémy) cez reklamné a informačné materiály (pozvánky, letáky, plagáty), počítačové spracovanie fotografických predlôh, prezentácie ako ucelené sekvencie a grafický sprievod k prednáške, animovanú (pohyblivú, často ozvučenú) grafiku, grafické užívateľské rozhranie (GUI – Graphical User Interface), technické výkresy (CAD – Computer-aided design), mapy až po grafiku pre web a multimédiá.

Počítače v súčasnosti umožňujú priblížiť reálny priestor aj činnosť človeka. Počítačovým modelom reálneho trojdimenzionálneho (3D) priestoru vzniká virtuálna, t. j. možná, potenciálna, podmienené jestvujúca realita, čiže skutočnosť simulovaná pomocou počítačovej techniky a jej účastník sa v tomto priestore pohybuje.

Virtuálna grafika je teda grafika fungujúca na pomedzí reality a fikcie zobrazujúca virtuálnu realitu, virtuálny svet, teda niečo nereálne. Virtuálna grafika má svoje veľké využitie v hernom priemysle. Od vytvorenia priestoru, v ktorom sa hra odohráva, cez vytvorenie postáv spracované v 3D alebo prostredníctvom digitálnej maľby až po samotné naprogramovanie hry. Virtuálnu grafiku charakterizuje tiež interaktivita, ktorá ponúka umelcom, novinárom, tvorcom športových alebo zábavných programov nové spôsoby, ako osloviť a zaujať divákov.

Rozšírená realita je umelé prostredie v digitálnej podobe, ktoré obohacuje skutočné prostredie o digitálny obsah. V porovnaní s virtuálnou realitou, ktorá pomocou špeciálnych okuliarov ponúka užívateľovi kompletne ponorenie do virtuálneho sveta a prakticky umožňuje vidieť čokoľvek a kdekoľvek, rozšírená realita umožňuje súbežne pozorovať reálny svet s pridanými prvkami. Medzi prvky rozšírenej reality patrí text, obrázky, 2D a 3D modely a taktiež animácie. Aplikácie rozšírenej reality sú zvyčajne využívané v reálnom čase s kamerami alebo videozáznamami na počítačoch a smartfónoch. Príkladom takých aplikácií je vkladanie rôznych druhov nábytku do miestnosti, zobrazenie cesty v priestore pre navigáciu na určité miesto alebo aj súčasť zadnej parkovacej kamery osobných automobilov, kde zobrazuje vodiace čiary smeru pohybu [3].

Rozšírená realita je efektívny spôsob, ako obraz alebo video, resp. ich kombináciu doplnením grafických prvkov vytvorených v počítači urobiť pre užívateľa veľmi príťažlivými, lákavými a pútavými. Táto technológia sa najskôr začala využívať hlavne vo filmovom priemysle, ktorý využíva práve kombináciu filmových záberov so skutočnými hercami a priestorom, ktorý je vytvorený v 3D programe. Aj v bežnom televíznom programe ako predpoveď počasia sa moderátor či moderátorka nachádzajú v štúdiu so zelenou stenou a diváci prostredníctvom grafiky majú možnosť sledovať predpoveď počasia spolu s rôznymi grafmi a animáciami.



Obr. 2.1: Obrázok (a) znázorňuje kocku ako prvok virtuálnej grafiky pomocou modelovacieho softvéru Blender. Obrázok (b) znázorňuje reálne prostredie prostredníctvom snímky zachytenej kamerou. Obrázok (c) znázorňuje kocku z obrázku (a) ako virtuálny grafický prvok rozšírenej reality.

Existuje mnoho systémov zaoberajúcich sa športovou grafickou analýzou. Všetky vyhľadané systémy sú dostupné len v platenej verzii, a preto je ich podrobnejšie štúdium možné len prostredníctvom verejne dostupných videí, z ktorých nie je možné zistiť technickú realizáciu. Štúdium týchto videozáznamov však vytvára predstavu pre návrh nového systému. V nasledujúcej časti spomeniem niektoré systémy z hľadiska rozličných funkcií, ktoré ponúkajú. Systém LongoMatch¹ používa dvojrozmerné grafické objekty, ktoré na zobrazenie kompletne nastavuje užívateľ (určuje veľkosť objektov, pozíciu prípadne rotáciu). Systém ChyronHego² obsahuje podrobnejšiu technológiu sledovania a vytvárania masky hráčov.

¹Dostupné z : <https://longomatch.com/en/>. Video použitia: <https://youtu.be/WCjw7uhEctA?t=922>

²Dostupné z: <https://chyron.com/solutions/sport-broadcast-solutions/>. Video použitia: <https://www.youtube.com/user/chyronmelville/videos>.

Animáciu pohybu grafiky pre video záznamy je možné nastavovať pomocou časových kľúčov vo videu. S novými aktualizáciami systém umožňuje virtuálnu rekonštrukciu zápasu na základe predchádzajúcej analýzy celého videozáznamu. Veľmi zaujímavou integráciou rozšírenej reality do prostredia športovej analýzy ponúka systém wTVision³, ktorý umožňuje pri pohľade na futbalovú plochu zobrazit akékoľvek snímky/videa. Z užívateľského pohľadu táto grafika vyzerá ako veľké projekčné plátno nachádzajúce sa na hracej ploche.

Skúmaním systémov používaných na interaktívnu grafickú analýzu športových podujatí som zistil, že takmer všetky softvéry zobrazujú dvojrozmernú grafiku. V softvéri Panoris⁴ spoločnosti Camvision, s. r. o., je zobrazovanie grafiky podobné ako v ostatných systémoch pomocou dvojrozmernej grafiky. Spoločnosť prišla s nápadom pre zadanie tejto práce vytvoriť trojrozmerné zobrazovania grafiky do športových videozáznamov. Predpokladá sa, že vytvorené nástroje budú integrované do softvéru spoločnosti Camvision, ktorý je vytvorený v programovacom jazyku C++ a využíva OpenGL API na vykresľovanie grafiky prostredníctvom grafických kariet.

2.2 Zobrazovanie grafických objektov

Jedným z východísk na vypracovanie tejto práce bolo použitie OpenGL ako grafickej knižnice na zobrazovanie 3D virtuálnej grafiky, ktoré je založené na použití matematických výpočtov na polohovanie a osvetlenie grafických objektov vo virtuálnej scéne. OpenGL definuje prístup k grafickému procesoru (grafickej karte alebo GPU – graphics processing unit), pomocou ktorého je pomerne náročný výpočet urýchlený a umožňuje zobrazenie v reálnom čase. Výsledná virtuálna scéna vzniká pre násobením súradníc vrcholov, ktoré definujú jednotlivé grafické objekty, a transformačných matíc rozličných súradnicových priestorov (objektového, pozorovateľského, orezového, priestoru normalizovaných súradníc zariadenia a priestoru okna). Na zobrazenie vrcholov, ktoré popisujú jednotlivé objekty, sú použité počítačové programy vytvárajúce reťazec spracovania GPU. Každý objekt scény vykresľovaný pomocou grafickej knižnice je definovaný z vrcholov v 3D euklidovskom priestore. Po načítaní vrcholov do pamäte GPU sú vrcholy transformované pomocou vertex shaderu a vizuálne vytvárajú obrisy modelov. Na urýchlenie výpočtu transformácií bodu sú súradnice vrcholu upravené do homogénneho tvaru. Výsledná farba je následne počítaná pomocou fragment shaderu pre každý pixel objektu a môže byť určená osvetľovacím modelom, textúrou alebo inými grafickými technikami. V nasledujúcej časti práce je opísané použitie grafických techník pre zobrazenie virtuálnej grafiky v rámci našej virtuálnej scény.

Phongov osvetľovací model

Phongov osvetľovací model je empirický model (nie je založený na fyzikálnej podstate), ktorý sa zaoberá výpočtom odrazeného svetla a tieňovania. Nasledujúci vzorec popisuje výslednú hodnotu I farby objektu pre daný pixel. Skladá sa súčtom troch zložiek: ambientnej I_A , difúznej I_D a spekulárnej I_S [7].

$$I = I_A + I_D + I_S \quad (2.1)$$

Ambientná zložka predstavuje odrazené svetlo zo všetkých okolitých objektov. Pre každý objekt je po celý čas konštantné a umožňuje čiastočne vnímať farbu objektu v miestach

³Dostupné z: <https://www.wtvision.com/en/>. Video použitia: <https://www.youtube.com/channel/UCAvcF04rPCivDUDVQNcW4Sg>.

⁴Dostupné z: <http://www.panoris.com/>.

na ktoré nedopadá priamy zdroj svetla. Čím je hodnota ambientnej zložky vyššia, tým väčšie je celkové osvetlenie scény. Ambientná zložka je vyjadrená rovnicou 2.2, kde hodnota ambientnej zložky je označená ako I_A , I_a predstavuje koeficienty intenzity ambientnej zložky a r_a predstavuje koeficient schopnosti materiálu odrážať svetlo.

$$I_A = I_a r_a \quad (2.2)$$

Difúzna zložka predstavuje hodnotu odrazeného svetla od jeho povrchu. Jedná sa o najdôležitejšiu vizuálnu časť Phongovho osvetľovacieho modelu, keďže priamo určuje farbu objektu. Difúzna zložka je daná rovnicou 2.3 a označuje sa ako *I_diff*. Koeficient I_d predstavuje intenzitu difúznej zložky svetla, r_a je koeficient odrazivosti svetla od materiálu, N je normála povrchu v danom mieste a L je smer odrazu svetla.

$$I_D = I_d r_d (N \cdot L) \quad (2.3)$$

Spekulárna zložka znázorňuje odrazené odlesky lúčov primárneho zdroja svetla od povrchu objektu. V prípade kovu, ktorého povrch je tvorený z lesklého materiálu bude intenzita odrazivosti vyššia ako pri drevenom matnom materiáli. Spekulárna zložka je daná vzťahom 2.4 a jej označenie je I_S . Podobne ako pri difúznej a ambientnej zložke je I_s intenzita spekulárnej zložky a r_s koeficient odrazivosti svetla. Odlesky sú vnímané z miesta pozorovania, a V určuje tento smer. Farba spekulárnej zložky svetla je v súlade s farbou svetla, ktoré ho produkuje. Smer odrazu tejto zložky svetla vychádza zo zákona odrazu svetla a je označený ako R .

$$I_S = I_s r_s (V \cdot R)^n \quad (2.4)$$

Animácie

Animácie v počítačovej grafike predstavujú postupnosť meniacich sa obrázkov. Ak chceme nejaký objekt animovať, musíme meniť jeho pozíciu po malých kúskoch s rýchlym opakovaním. Na základe vnímania rýchlo meniacich sa zobrazení objektu vytvoríme dojem, že sa daný objekt pohybuje. Ako uvádza Guha [5], existujú tri možnosti kontrolovanie animačného pohybu, ktorými sú:

- Interaktívne kontrolovanie animácie pomocou vstupu z klávesnice alebo myši.
- Automatická animácia, ktorej chod nemôže užívateľ ovládať.
- Poloautomatická animácia, pre ktorú je špecifikovaná časová funkcia na základe systémového času a je možné nastavovať jej dĺžku.

2.3 Detekcia hráčov

Táto kapitola je zameraná na detekciu hráčov na športovom ihrisku a ich klasifikáciu. Detekcia hráčov vychádza z rozpoznania viacerých osôb – hráčov v skúmanej snímke. Keďže detekcia osôb je pomerne populárna téma v oblasti počítačového videnia, postupom času vznikali viaceré metódy detekcie osôb v reálnom čase. Princípy a konkrétne metódy detekcie osôb sú popísané v článku [10] a hovoria o detekcii na základe segmentácie popredia, skúmania charakteristík vektorov toku pohybujúcich sa objektov v postupnosti obrázkov a metód využívajúcich časopriestorový filter.

Na športových podujatiach je detekcia sťažená mnohými nepredvídateľnými pózami (stoj, chôdza, beh, výskok, pád...), v ktorých sa osoba môže vyskytnúť, a zároveň prostredím, v ktorom osoby často menia svoju pozíciu v svižnej dynamike pohybu, a preto detekcii bola venovaná samostatná pozornosť. Metódy patriace k segmentácii popredia [14, 12] skúmajúce detekciu osôb – hráčov na futbalových zápasoch častejšie využívajú na detekciu spektrum farieb, keďže pozadím detekcie je zelená trávnatá plocha. Zároveň sa medzi segmentáciu popredia zaraďuje Gaussian Mixture Model [17]. Ďalej je využívaný histogram orientovaných gradientov [9], ktorý na základe hranových orientácií určuje objekty.

Pri detekcii objektov je potrebné oddeliť popredie od pozadia snímky, tento postup sa nazýva segmentácia popredia. Získané informácie o popredí sú využité pri klasifikácii objektov, medzi ktoré patria: športové ihrisko, hráči a prípadne rozhodcovia a športové náčinie – lopta. S klasifikáciou objektov súvisí sledovanie – tracking objektov.

Vyskytujúcim sa problémom pri detekcii osôb môže byť rozlíšenie obrazu, keď kamera sníma celú hraciu plochu a detegované objekty sú vo veľkosti 10 pixelov a menej, prekryvanie osôb inou osobou alebo objektom, alebo objekt, ktorý je nesprávne vyhodnotený ako osoba.

Segmentácia popredia

Segmentáciu popredia [17] popísal Zivkovic ako detekciu objektov, ktoré nejakým spôsobom narúšajú analyzovanú scénu. Predpokladá, že objekty, ktoré do scény patria, vykazujú podobné vlastnosti a môžu byť opísané štatistickým modelom. Pozorovaním štatistického modelu na analýzu snímok je možné zistiť časti scény, ktoré do nej nepatria, a teda rozpoznať objekty popredia. Pri scéne, ktorá je sledovaná statickou kamerou, je možné určiť objekty popredia podľa toho, či medzi jednotlivými snímkami menia svoju polohu.

Vytvorenie štatistického modelu vychádza z rozhodnutia, či skúmaný pixel patrí alebo nepatrí do pozadia snímky. Na začiatku analýzy videa nie je nič známe o objektoch popredia alebo pozadia, a preto je nutné predpokladať totožnosť modelu popredia s modelom pozadia. Ďalším skúmaním sekvencie snímok je možné rozhodnúť, či daný pixel patrí pozadiu alebo poprediu použitím vzorca:

$$p(x^{(t)}|BG) > c_{\text{thr}} \quad (2.5)$$

Pričom $x^{(t)}$ je hodnota pixla v spektre farieb, c_{thr} je hodnota prahu a $p(x^{(t)}|BG)$ je vektor hodnôt farby pixla pre model pozadia [17].

Gaussian mixture model

Gaussian mixture model je metóda založená na skúmaní postupnosti snímok s cieľom rozpoznať objekty, ktoré menia polohu na snímke. Vhodné použitie metódy je pri analyzovaní snímok zachytených statickou kamerou. Metóda je schopná reagovať na zmeny svetla v skúmanej scéne. Pri analýze využíva súbor údajov z analyzovaných snímok a detekuje pohybujúce sa objekty porovnávaním snímky voči referenčnej snímke, ktorá predstavuje model pozadia. Pixely, ktoré sú kategorizované ako model pozadia, majú nulovú hodnotu, respektíve čiernu farbu. V rámci metódy sú skúmané všetky pixely snímky [17].

$$\chi_T = x^{(t)}, \dots, x^{(t-T)} \quad (2.6)$$

Model dosahuje dostatočnú presnosť po analyzovaní počtu snímok vyjadrených ako χ v časovej perióde T a časom t . Tento model môžeme označiť ako $p'(x|\chi_T, BG)$.

2.4 Homogénne súradnice

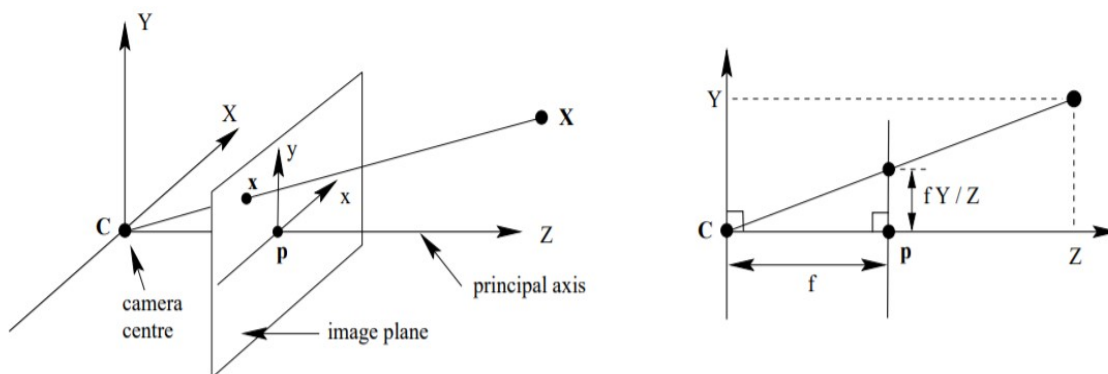
Bod je základným útvarom geometrie a v rôznych priestoroch alebo súradnicových systémoch má iné vyjadrenie. V tomto prípade ho vyjadríme číslom. Na použitie homogénnych súradníc je potrebné, aby bol bod vyjadrený spolu s jedným ďalším homogénnym parametrom w . Bod v 3D euklidovskom priestore, zapísaný pomocou vektora o veľkosti 1×3 ako $[x, y, z]$, vyjadríme pomocou homogénnych súradníc v tvare $[x, y, z, w]$, ktorý je rozšírený o jednu dimenziu s veľkosťou vektora 1×4 a w určuje váhu bodu, pričom $w \neq 0$. Pomocou vektorového zápisu môžeme homogénny bod v 3D priestore vyjadriť ako vektor o veľkosti 1×3 a jeho zápis je $[x, y, z]^T \mapsto [x/w, y/w, z/w]^t$. Ak je homogénna zložka bodu nenulové číslo, potom výsledné súradnice v 3D priestore reprezentujú ten istý bod.

Rozšírenie bodu do homogénneho tvaru umožňuje jednoduchšie použitie transformácií bodu pomocou násobenia matic. V prípade 3D priestoru môžeme bod transformovať pomocou matice rotácie o veľkosti 3×3 a vektora translácie o veľkosti 1×3 . Vytvorením matice transformácie o veľkosti 4×4 , ktorá vznikne spojením matice rotácie a vektora translácie, môžeme vynásobiť homogénne súradnice bodu s transformačnou maticou zachovaním pravidla o násobení matic, kde počet stĺpcov ľavej matice musí byť rovnaký ako počet riadkov pravej matice [7].

2.5 Model kamery

Zobrazenie objektov v scéne je relatívne voči orientácii a pozícii kamery v rámci scény. Vizualizácia 3D priestoru v počítačovej grafike je založená na projekcii 3D bodov do projekčnej roviny v 2D priestore. Perspektívna projekcia teda umožňuje vnímať 3D priestor zobrazovaný na obrazovke. Princíp perspektívnej projekcie je využitý aj v systémoch reálnych kamier, keď je snímaný priestor zobrazovaný na senzore kamery.

V počítačovej grafike sa toto zobrazenie nazýva pozorovateľský alebo kamerový priestor a je vyjadrený transformačnou maticou kamerového priestoru. Pri tvorbe virtuálnej grafiky pre rozšírenú realitu s cieľom zhodného vnímania perspektívy grafických objektov a snímaného videa vychádza nastavenie transformačnej matice kamerového priestoru z parametrov kamery, ktorá sníma video. V nasledujúcej časti práce sa budeme podrobnejšie venovať základnému dierkovému modelu kamery a jeho parametrom. Hartley a Zisserman [6] podrobne rozoberajú problematiku parametrov kamery, a preto väčšina teoretických poznatkov pre túto časť práce vychádza z ich skúmania.



Obr. 2.2: Obrázok znázorňujúci model dierkovej “pinhole” kamery. Prevzaté z [6]

Na obrázku 2.2 je znázornený všeobecný model „pinhole“ kamery. Bod \mathbf{C} určuje polohu kamery, ktorá sa nachádza v pôvode euklidovského súradnicového systému. Bod \mathbf{p} je stredom projekcie, nazývaný tiež úbežník, a nachádza sa v strede projekčnej roviny. Potom bod $X = [X, Y, Z]^T$ je premietnutý do projekčnej roviny v bode $x = [x, y]$, ktorý dostaneme pretnutím priamky spájajúcej bod \mathbf{C} a bod \mathbf{p} s projekčnou rovinou. Použitím vety o podobnosti trojuholníkov môžeme vypočítať súradnice bodu \mathbf{x} s použitím vzťahu: $[X, Y, Z]^T \mapsto [fX/Z, fY/Z]^t$ [6].

$$P = \begin{bmatrix} \frac{1}{\text{aspect} * \tan(\frac{fov}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{fov}{2})} & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2.7)$$

Na zobrazenie objektov pomocou dierkového modelu kamery je použitá transformačná matica perspektívnej projekcie, ktorá je aplikovaná po prevedení scény do kamerového priestoru. Maticové vyjadrenie perspektívnej projekcie 2.7 orezáva objekty podľa veľkosti obrazovky a vzdialenosti voči kamere pričom koeficient n určuje najbližšiu vzdialenosť zobrazených objektov a $n > 0$. Koeficient f určuje maximálnu vzdialenosť zobrazených objektov. Na zostavenie matice je tiež potrebné stanoviť uhol zorného poľa fov . Na obrázku 2.2 je predstavený ako uhol medzi priamkami z bodu \mathbf{C} k obidvom stranám obrazovej roviny. *Aspect* je pomer dĺžky a výšky obrazovej roviny.

Vnútorne parametre kamery

Vnútorne parametre kamery určujú optické vlastnosti kamery, ktorými sú ohnisková vzdialenosť, stred snímky a koeficienty skreslenia. Slúžia na mapovanie bodov z obrazovej roviny na pixely. Vnútorne parametre kamery je možné vyjadriť pomocou matice 2.8, kde f je ohnisková vzdialenosť a c určuje stred snímky. Pre danú ohniskovú vzdialenosť ostáva maticové vyjadrenie parametrov konštantné.

$$I = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Ako uvádza Hartley [6], parameter c určujúci stred snímky sa nemusí nachádzať nevyhnutne v presnom strede. Pri použití kvalitných kamier je však tento nedostatok redukovaný, a teda môžeme uvažovať, že $c_x = \frac{\text{sirkasnimku}}{2}$ a $c_y = \frac{\text{vyskasnimku}}{2}$.

$$\text{koeficienty skreslenia} = (k_1, k_2, p_1, p_2, k_3) \quad (2.9)$$

Medzi vnútorné parametre kamery patria tiež koeficienty skreslenia. Skreslenie vzniká pri prechode svetelného lúča cez zaoblenú šošovku kamery. Základné typy skreslenia sú radiálne k a tangenciálne p [1].

Pri radiálnom skreslení sa rovné okraje snímky zaoblujú. Čím sú body vzdialenejšie od stredu snímky, tým je skreslenie intenzívnejšie. Úprava radiálneho skreslenia do korektného obdĺžnikového tvaru snímky pomocou nasledujúcich rovníc pričom r určuje stupeň rozvoja Taylorovho polynómu (vysvetlené v citeBradski:OpencvLibrary).

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (2.10)$$

Tangenciálne skreslenie vzniká nerovnoběžnosťou šošovky a senzoru kamery. Odstránenie tangenciálneho skreslenia je možné pomocou rovníc:

$$\begin{aligned}x_{corrected} &= x + [2p_1xy + p_2(r_2 + 2x^2)] \\y_{corrected} &= y + [p_1(r^2 + 2y^2) + 2p_2xy]\end{aligned}\tag{2.11}$$

Vonkajšie parametre kamery

Vonkajšie parametre kamery definujú pozíciu a orientáciu kamery voči svetovému súradnicovému systému. Popísané sú rotačnou maticou R s rozmermi 3×3 , ktorá vznikne spojením rotácií okolo osí X, Y, Z a translačného vektora t veľkosti 3×1 , ktorý obsahuje súradnice kamery x, y, z svetového súradnicového systému.

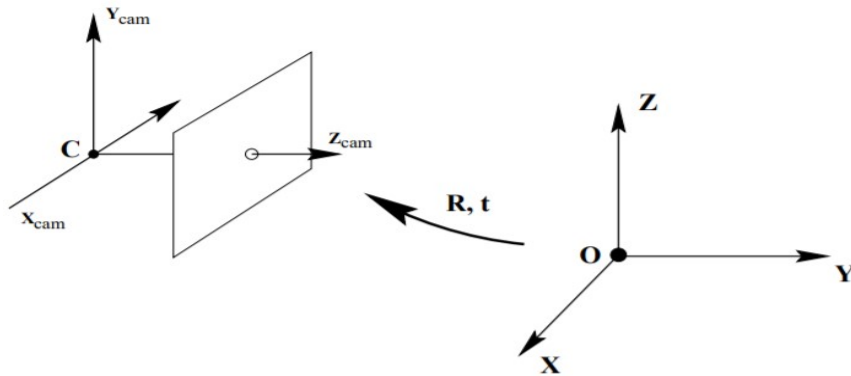
$$R|t = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{array} \right]\tag{2.12}$$

Na použitie matice 2.12 vonkajších parametrov kamery je potrebné transformovať maticu do homogénneho tvaru. Podľa [1] je homogénny tvar takýto:

$$R|t = \left[\begin{array}{ccc|c} R & t \\ 0 & 1 \end{array} \right]\tag{2.13}$$

Ako uvádza Hartley [6], súradnicový systém kamery je odlišný od svetového súradnicového systému v invertovaní osí Y a Z . Túto situáciu môžeme vidieť na obrázku 2.3, a preto rovnica 2.13 upravená na tvar 2.14 vyjadruje transformáciu bodu \mathbf{X} do súradnicového systému kamery. Maticu 2.14 je možné priamo použiť na určenie polohy kamery v svetovom súradnicovom systéme.

$$X_{cam} = \left[\begin{array}{c|c} R & -Rt \\ 0 & 1 \end{array} \right] X\tag{2.14}$$



Obr. 2.3: Obrázok popisujúci vzťah medzi kamerovým súradnicovým systémom so stredom v bode \mathbf{C} a svetovým súradnicovým systémom so stredom v bode \mathbf{O} . Transformácia zo svetového do kamerového súradnicového systému je vykonaná pomocou rotačnej matice R a translačného vektora t . Obrázok prevzatý z [6]

2.6 Odhad pozície kamery

Pomocou perspektívnej projekcie a parametrov kamery je možné odhadnúť podobnosti zhodných bodov na výpočet pozície a orientácie kamery v svetovom priestore. Odhad výpočtu pozície kamery sa tiež nazýva problém perspektívnych n bodov. Ako základ na výpočet pozície a orientácie kamery je potrebné určiť množinu bodov nazývanú objektové body s pozíciou v 3D súradnicovom systéme a množinu bodov nazývanú obrazové body, ktoré korešpondujú s množinou objektových bodov premietnutím na 2D projekčnú rovinu. Na správny výpočet je potrebné súradnice objektových bodov nastaviť podľa rozmerov v reálnom svete.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.15)$$

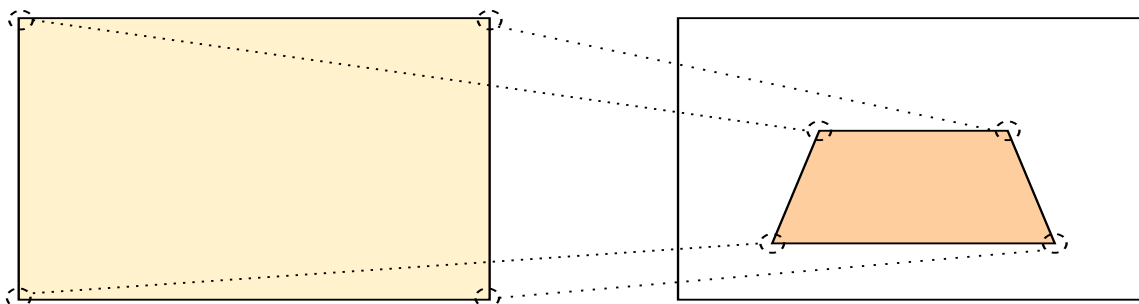
Výsledok riešenia je následne možné vypočítať na základe tejto sústavy matic, kde matica $[u, v, 1]$ reprezentuje obrazové body, matica $[X_w, Y_w, Z_w, 1]$ určuje objektové body a všetky ostatné matice sú známe.

Na odhad transformačnej matice je možné použiť tieto metódy:

- **Iteratívna metóda** je založená na Levenberg-Marquardt optimalizačnej metóde. Pomocou tejto metódy je možné vyhľadať pozíciu kamery na základe najmenej vzdialenosti medzi obrazovými bodmi a premietnutými modelovými bodmi. Jej cieľom je minimalizovať reprojekčnú chybu [1].
- **Metóda troch perspektívnych bodov** metóda pre výpočet používa korešpondenciu troch obrazových a objektových bodov a ako výsledok výpočtu môže stanoviť až štyri reálne riešenia [4].
- **Efektívny odhad polohy kamery pomocou perspektívnych n bodov (EPNP)** je metóda, ktorá používa pre výpočet pozície štyri a viac korešpondenčných bodov pre nie len rovinné objekty [8].

2.7 Homografia

Homografia je definovaná ako kolineácia alebo projektívna transformácia, lineárneho mapovania $P^d \mapsto P^d$ v priestore P^{d+1} [15]. Metóda teda vychádza zo znalosti zhodných bodov aspoň v dvoch snímkach. Známe body slúžia na výpočet transformačnej matice homografie označenej ako H . Nájdená transformačná matica sa potom použije na transformáciu jednej zo snímkov. Túto metódu je však možné použiť len v prípade, keď je základom rekonštruovanej 3D scény plocha a hĺbková hodnota 3D bodov je rovná nule [13]. Najčastejšie využitie homografie je pri rektifikácii obrazu alebo vytváraní panoramatických snímkov.



Obr. 2.4: Obrázok znázorňujúci mapovanie zhodných bodov v dvoch snímkach pomocou D homografie. V tomto prípade snímka v ľavej časti obrázku predstavuje obdĺžnik a snímka v pravej časti obrázku znázorňuje perspektívne zobrazenie toho istého obdĺžnika.

Ako uvádza Szeliski [16], výpočet transformačnej matice H musí pozostávať zo štyroch a viacerých zhodných bodov. Následné použitie transformačnej matice je dané rovnicou:

$$x' = Hx, \quad (2.16)$$

pričom $x' = [x', y', 1]^T$ označuje homogénne súradnice výsledného bodu, H je transformačná matica homografie a $x = [x, y, 1]^T$ určuje homogénne súradnice transformovaného bodu.

Kapitola 3

Návrh nástrojov a aplikácie

Cieľom tejto časti bakalárskej práce je navrhnúť zobrazenie trojrozmerných grafických objektov ako súčasti virtuálnej reality videozáznamov športových podujatí snímaných kamerovým systémom spoločnosti Camvision, s. r. o.

Na dosiahnutie hlavného cieľa je potrebné vykonať niekoľko čiastkových úloh:

1. Navrhnuť spôsob tvorby virtuálnej scény na základe projekčných údajov o kamere.
2. Navrhnuť detekciu hráčov na demonštráciu použitia grafických nástrojov.
3. Navrhnuť prvky grafických nástrojov a ich správne polohovanie vo virtuálnej scéne.

Na začiatok je potrebné opísať a vymedziť nástroje, ktorých funkciou sa budeme zaoberať z hľadiska analýzy videa (detekcia hráčov, rekonštrukcia 3D scény pomocou homografie s výpočtom pozície kamery v analyzovanom videu) a zobrazenia virtuálnej grafiky (definícia objektov a ich funkcionality pre nástroje).

Motiváciou na vytvorenie grafických nástrojov, ktoré bude používať užívateľ (tréner, video-analytik), je znázornenie:

- pozície hráčov,
- smeru pohybu hráčov a smeru pohybu lopty,
- postavenie hráčov v rámci pravidiel hry.

V nasledujúcej časti bude definovaná scéna na použitie nástrojov a zároveň návrh nástrojov na znázornenie pohybu hráčov s názvom – Zvýraznenie hráčov, znázornenie smeru pohybu hráčov a smeru pohybu lopty – Orientovaný smerovník a znázornenie postavenie hráčov v rámci pravidiel hry – Ofsajdová čiara.

3.1 Popis scény

Predmetom skúmania pre analyzačné nástroje a zobrazovanie virtuálnej grafiky je videozáznam tímových športov snímaný statickou kamerou. Podmienkou pre tímové športy je ihrisko pozostávajúce z obdĺžnikovej hracej plochy (futbal, hádzaná, basketbal a taktiež hokej avšak nastavenie rohov ihriska vyžaduje väčšiu pozornosť). V tejto práci sa budeme zaoberať futbalovými ihriskami a verejne dostupnými videozáznamami, ktoré vyhovujú danej podmienke.

Reálna scéna je to, čo vníma oko kamery v daný moment. Ako virtuálnu scénu potom označíme súbor všetkých virtuálnych grafických prvkov zobrazených v rámci digitálneho zobrazenia scény. Grafický systém vytváraný v tejto práci zobrazuje kombináciu reálnej scény reprezentovanej videom a virtuálnej scény. Spojením týchto dvoch scén dostaneme výslednú scénu tejto práce, s ktorou bude pracovať užívateľ narábajúci s nástrojmi. Výslednú scénu budeme ďalej používať pod názvom scéna.

Pri tvorbe grafických aplikácií vo virtuálnom svete môžeme spomenuté transformácie nastaviť podľa potrieb danej aplikácie. Rozlišovať sa môžu napríklad nastavením zorného poľa v horizontálnej rovine, kde pre panoramatické (širokouhlé) zobrazenie bude použitá hodnota 180° – 360° alebo priblížené pozorovanie pod mikroskopom s hodnotou cca 20° .

Súčasťou scény je reálna scéna reprezentovaná videozáznamom snímaným kamerou, ktorá funguje na princípe základného kamerového modelu. Ak je reálna scéna obsahom analyzovaného videa a virtuálna scéna obsahuje všetky virtuálne grafické objekty, výslednú scénu potom dostaneme spojením videa a virtuálnej scény. Ak v rámci scény chceme vnímať virtuálnu grafiku v podobe rozšírenej reality, potrebujeme na zobrazenie virtuálnej grafiky použiť rovnaký model kamery ako v prípade snímanej reálnej scény. Zároveň na správne zobrazenie virtuálnej scény v rámci videa pre tvorbu rozšírenej reality musí pozícia oka kamery vo virtuálnom svete korešpondovať s pozíciou kamery v analyzovanom videu, keďže použijeme rovnaký model kamery 2.5 pre reálny aj virtuálny svet. Takýto pohľad sa nazýva pozorovateľský priestor. Nastavenie transformačnej matice pozorovateľského priestoru musí byť presne stanovené analýzou danej scény (videa).

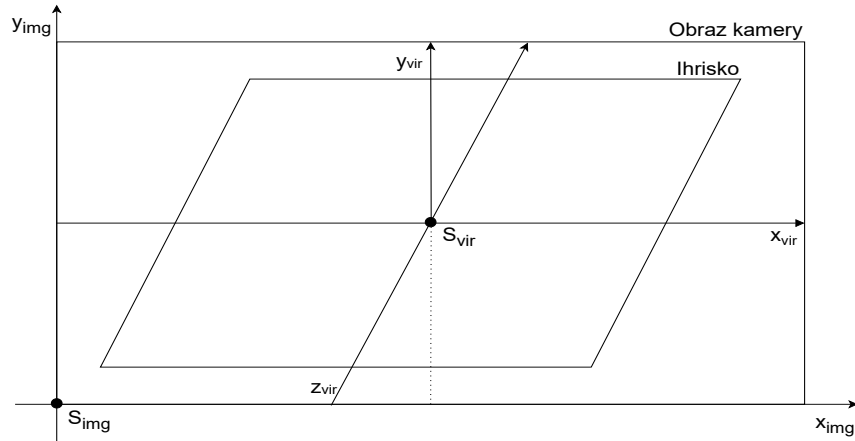
Keďže „správny“ je relatívny pojem, je potrebné stanoviť hodnotenie správneho a nesprávneho zobrazenia. Správne zobrazenie objektu znamená, že sa objekt zobrazí na stanovenom mieste (napríklad ak chceme zvýrazniť hráča, pozorovateľovi je na prvý pohľad jasné, o ktorého hráča ide). Formálnejší popis správneho zobrazenia virtuálnej grafiky je taký, že premietnuté rohy virtuálneho ihriska pomocou pozorovateľského a projekčného priestoru do súradníc snímky sa zhodujú s manuálne zadanými rohmi ihriska v rámci snímky alebo je pre všetky body priemerná chyba zobrazenia menšia ako stanovená hodnota. Táto chyba sa udáva v pixeloch a je určená ako priemer celkovej vzdialenosti medzi obrazovými bodmi zodpovedajúcimi premietnutými objektovými bodmi do súradnicového systému obrazovky.

V rámci popisu reálnej scény je potrebné vymedziť rozmery športového ihriska, ktorého parametre sú použité v rovnakých rozmeroch pre virtuálnu grafiku. V tejto bakalárskej práci sa budeme zaoberať futbalovými zápasmi, ktoré sa hrajú na ihrisku s rozmermi 90 – 120 m dĺžky a 45 – 90 m šírky (pre medzinárodné stretnutie platí 100 – 110 m dĺžky a 64 – 75 m šírky)¹, a musíme rátať s tým, že vždy narábame s iným reálnym priestorom. Znamená to, že divák nerozoznáva rozdiely rozličných rozmerov hracej plochy (keďže to nie je pre neho z diváckeho hľadiska podstatné).

Keď vieme, že ihrisko má určité rozmery, môžeme tieto rozmery použiť na nastavenie parametrov virtuálneho ihriska a ostatných grafických objektov. Vytvorí sa tak korešpondencia reálnych objektov a virtuálnej grafiky. Ak má reálne ihrisko rozmery dĺžka x šírka, súradnice štyroch rohov virtuálneho ihriska sú dané nasledujúcim vzťahom.

$$\begin{aligned} & [dlzka/2, 0, sirka/2] \\ & [dlzka/2, 0, -sirka/2] \\ & [-dlzka/2, 0, sirka/2] \\ & [-dlzka/2, 0, -sirka/2] \end{aligned} \tag{3.1}$$

¹Podľa: <https://sk.wikipedia.org/wiki/Futbal>



Obr. 3.1: Súradnicový systém snímky je tvorený 2D súradnicami. Reálny svet je však daný reálnymi rozmermi, na základe ktorých vytvárame korešpondenčný súradnicový systém pre virtuálne objekty. Na obrázku môžeme pozorovať túto situáciu, kde pre virtuálnu scénu je bod S_{vir} pôvod súradnicového systému a v bode S_{img} sa nachádza pôvod súradnicového systému snímky.

Nastavenie kamery pre virtuálnu scénu

Na výpočet pozície kamery v súradnicovom systéme virtuálnej scény je potrebné otestovať použitie rôznych metód popísaných v časti 2.6. Keďže skúmané videozáznamy v tejto práci obsahujú panoramatické zábery celého ihriska a je známa geometria scény, predpokladá sa, že metóda EPNP [8] bude vhodným výberom.

Na výpočet pomocou tejto metódy je potrebné zvoliť štyri body pre rohy ihriska v analyzovanom videu, ktoré sa použijú ako obrazové body pre danú metódu. Zadanie rohov ihriska musí prebehnúť pred načítaním prvej snímky videa pomocou grafického užívateľského rozhrania alebo použitím vopred uložených údajov. Následne vyberieme zodpovedajúce rohy virtuálneho ihriska – objektové body – dané rovnicou 3.1.

Drobné nepresnosti rozmerov ihriska, ktoré sú použité ako vstupné hodnoty na výpočet pozície kamery, produkujú pri výpočte pozície kamery veľmi odlišné výsledky, keďže výpočet pracuje na princípe odhadu. Ďalej môžu vzniknúť nepresnosti manuálneho zadania rohov ihriska, čo tiež ovplyvňuje finálny výpočet.

Mapovanie súradníc videa do súradníc scény

Vo videu máme súradnice v 2D systéme a grafické objekty používajú 3D súradnicový systém. Detegovanými objektmi (hráči) získavame pozíciu v 2D súradnicovom systéme a potrebujeme získať pozíciu v našej virtuálnej scéne, ktorá používa 3D súradnicový systém. Na výpočet transformačnej matice homografie použijeme podobne ako pri kalibrovaní pozície kamery virtuálnej scény manuálne zadané obrazové body štyroch rohov futbalového ihriska. Takisto podľa rovnice 3.1 nastavíme virtuálny 3D model ihriska.

Na základe známych obrazových a objektových bodov, ktoré dosadíme do rovnice 2.4, pričom za x' dosadíme obrazové body a za x objektové body, vypočítame transformačnú maticu homografie. Pomocou tejto matice je možné akýkoľvek bod v súradniciach videa transformovať alebo namapovať do súradníc scény.

3.2 Zvýraznenie hráčov

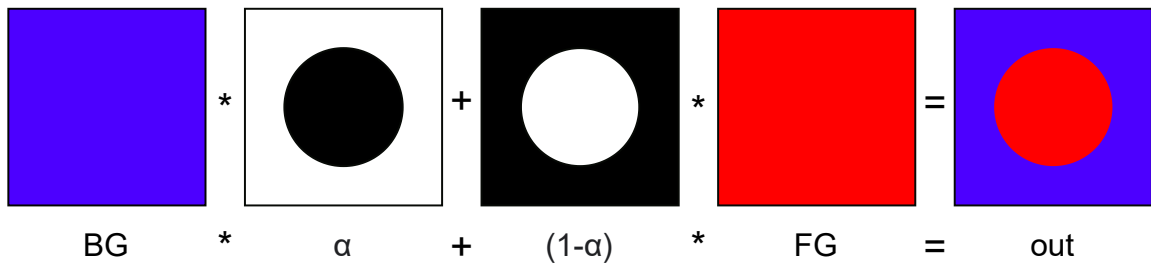
Na ploche ihriska je vo futbalovom zápase vždy okolo dvadsiatky hráčov. Nás však nezaujímajú vždy všetci. Zaujímajú nás tí, ktorí vytvárajú nejakú pozíciu, ktorú potrebujeme s trénerom analyzovať. Na to ich potrebujeme označiť. Označenie môže byť graficky znázornené rôznymi spôsobmi, napríklad zobrazenie grafického objektu pod hráčom na úrovni zeme formou obruče alebo nad hlavou hráča formou šípky, alebo pomyselného snečného líča. Na to, aby sme mohli zvýrazniť hráča ako objekt nášho záujmu, potrebujeme vedieť jeho polohu na ihrisku.

Detekciu hráčov pre potreby ich zvýraznenia v skúmanom videozázname vytváranom statickou videokamerou je možné dosiahnuť nasledujúcim spôsobom. Na počiatku sa vymedzí oblasť detekcie objektov, ktorú je potrebné skúmať. Zvyčajne sa detekcia objektov vykonáva na celej snímke, avšak v prípade znalosti výskytového priestoru objektov pomocou manuálne zadaných rohov ihriska je oblasť detekcie presne ohraničená a nie je potrebné skúmať celú snímku (pre všetky pixely, ktoré nepatria vymedzujúcej oblasti, nastavíme nulovú hodnotu, respektíve čiernu farbu pixelu). Následne pomocou detekčného algoritmu získame obrysy jednotlivých detegovaných hráčov a ich súradnice v priestore obrazovky.

Na detekciu hráčov ako funkciu určíme analyzovanú snímku ako vstupný parameter a pole prvkov, pričom každý prvok poľa je vyjadrený dvomi celočíselnými hodnotami určujúcimi pozíciu hráča v analyzovanej snímke. V tejto práci použijeme ako algoritmus detekcie objektov Gaussian mixture model (časť 2.3), ktorý je schopný detekcie pohybujúcich sa objektov v analyzovanom videu po natrénovaní modelu pozadia danej scény. Trénovanie modelu pozadia vyžaduje stanovený počet analyzovaných snímok videa. Čím je stanovený počet snímok na trénovanie modelu pozadia vyšší, tým je presnejšia detekcia objektov pomocou Gaussian mixture modelu. Po detekcii objektov popredia pomocou Gaussian mixture modelu je potrebné overiť, či detegované objekty sú skutočne hľadaní hráči. Keďže model pozadia pre skúmané futbalové zápasy má prevažne konštantnú farbu, môžeme overiť, či detegované obrysy hráčov obsahujú minimálny počet pixelov (podľa povahy skúmaných videozáznamov je minimálny počet pixelov rovný hodnote 10) farebnej škály podľa farby tímových dresov jednotlivých hráčov. Ak je táto podmienka splnená, môžeme do poľa prvkov pridať súradnice obrazovky v mieste nôh hráča, teda súradnice stredu spodnej hrany pomyselného obrysového štvorca detegovaného objektu.

Získané obrysové štvorce hráčov použijeme na vytvorenie alpha masky, ktorú postavíme do popredia grafických objektov a tým zamedzíme, aby vkladaná virtuálna grafika prekrývala oblasť hráčov. Výsledný obraz pomocou alpha masky dostaneme použitím nasledujúcej rovnice, pričom BG je farba pozadia, FG je farba popredia, out určuje výslednú farbu a α je hodnota alpha masky $\alpha \in \langle 0, 1 \rangle$ (čierna farba označuje hodnotu 0 a biela farba označuje hodnotu 1, pričom určuje výsledný obraz) [5].

$$BG * \alpha + (1 - \alpha) * FG = out \quad (3.2)$$



Obr. 3.2: Znáznornenie použitia alfa masky.

Na maskovanie hráčov voči grafickým objektom sa ako popredie FG použije aktuálna snímka videa a pozadie BG je tvorené z kombinácie aktuálnej snímky videa a grafických objektov. Snímka videa v oblasti hráčov potom za použitia masky vytvorenej pomocou detekčnej metódy prekryje zobrazenú virtuálnu grafiku.

Následne sú získané súradnice hráčov posunuté sledovaciemu algoritmu viacerých objektov v postupnosti snímok, v ktorých objekty menia svoju polohu. Pri sledovaní hráčov je potrebné správne priradiť pozíciu hráča v aktuálnej snímke k pozícii hráča v predošlej snímke. Možným riešením tohto problému je triediaci algoritmus, ktorý vyhľadá najmenší rozdiel vzdialenosti polôh objektov medzi dvomi snímkami.

Problémy tohto triedenia vznikajú v prípade, keď detekčný algoritmus nevyhľadá polohu niektorého z objektov. Ako uvádza Manoranjan [10], možným riešením spresnenia detekcie je použitie Kalmanovho filtra, ktorý v prípade neidentifikovanej polohy hráča v aktuálnej snímke vypočíta predikciu jeho polohy na základe množiny predošlých polôh. Ďalším problémom je výskyt viacerých hráčov v bezprostrednej blízkosti (respektíve prekrytie jedného alebo viacerých hráčov iným hráčom), ktorý je redukovaný detekciou hráčov z rôznych tímov na základe farby ich dresov a zároveň je možné v prípade prekrytia použiť predikovanú polohu pomocou Kalmanovho filtra.

Na zobrazenie grafického objektu, ktorý zvyrazňuje hráča, som vybral obruč pod nohami hráča a na polohovanie obruče do scény použijem súradnice stredu spodnej hrany obrysového štvorca z detekcie hráčov.

Zvýraznenie je možné animovať takým spôsobom, že obruči vytvoríme výřezy, čo umožní animovať rotáciou. Zároveň samotné sledovanie a smerovanie objektu na rôznej pozícii v súlade s hráčom je tiež druhom animácie, keďže sa daná obruč hýbe.

3.3 Orientovaný smerovník

Na to, aby sme načrtli zobrazenie pohybu hráčov, ako aj zobrazenie pohybu lopty, potrebujeme orientovaný smerovník pripomínajúci tvar šípky. Vytvára sa manuálne užívateľom pomocou grafického užívateľského rozhrania.

Na vytvorenie smerovníka je potrebné určiť počiatočný a koncový bod. Body určujú pozíciu v rámci dĺžky a šírky ihriska. Body môžu byť vybrané kliknutím na obrazovke alebo počiatočná poloha môže vychádzať z hráča. Môžu potom vzniknúť kombinácie týchto prvkov, a teda zobrazenie smerovníka od hráča k hráčovi, od hráča do priestoru, z priestoru k hráčovi alebo iba smerovník znázornený v priestore.

Spôsoby zobrazenia smerovníka môžu byť rôzne. Keďže máme pozíciu v rámci dĺžky a šírky pevne danú počiatočným a koncovým bodom, smerovník polohujeme v rôznej výške.

Môžeme ho teda nakresliť na úrovni zeme, vo vzduchu alebo zobraziť ako krivku (ktorú vykoná lopta pri vzlete a následnom padaní).

Grafický objekt smerovníka nie je celý definovaný ako samostatný 3D objekt, ale generovaný dynamicky a pozostáva zo spojenia škálovateľného kužela a valca. Kužel je následne potrebné posunúť o dĺžku valca v danom smere. Zobrazenie smerovníka môže byť statické alebo dynamické. Pri dynamickom uplatňujeme animáciu rastu z počiatočného do koncového bodu smerovníka.

V grafickom prostredí OpenGL nie je možné objekt zobraziť z počiatočného do koncového bodu, ale je potrebné vypočítať vzdialenosť medzi danými bodmi a podľa tejto vzdialenosti škálovať objekt. Tým dostávame správnu dĺžku objektu medzi počiatočným a koncovým bodom, avšak orientácia nie je správna. Výpočet potrebnej rotácie na správnu orientáciu smerovníka ku koncovému bodu dosiahneme zostrojením priamky medzi počiatočným a koncovým bodom smerovníka a využitím vektorovej algebry vypočítame uhol priamky voči osi x . Tento výpočet môžeme použiť vzhľadom na počiatočnú pozíciu, ktorá kopíruje os x v základnej pozícii.

$$\alpha = \arctangent\left(\frac{p_2 \cdot y - p_1 \cdot y}{p_2 \cdot x - p_1 \cdot x}\right) \quad (3.3)$$

Orientácia smerovníka je daná uhlom α , bod \mathbf{p}_1 určuje počiatočný bod smerovníka a bod \mathbf{p}_2 určuje jeho koncový bod.

3.4 Ofsajdová čiara

Na to, aby sme vedeli, či je herné postavenie hráčov správne, potrebujeme vytvoriť ofsajdovú čiaru, ktorá vymedzuje líniu postavenia hráčov v rámci ihriska. Pri aktivácii tohto grafického nástroja vyberieme na obrazovke východiskový bod zobrazenia patriaci ihrisku v reálnej scéne a ofsajdová čiara sa zobrazí po celej šírke ihriska vo virtuálnej scéne, pričom každý bod ofsajdovej čiary bude rovnako vzdialený od kratších hraničných čiar ihriska.

Východiskový bod môže reprezentovať pozíciu brániacich alebo útočiacich hráčov. Ak sa útočiaci hráč nachádza v ofsajdovej pozícii, tak sa čiara automaticky vykreslí červenou farbou, v opačnom prípade zelenou farbou. Pozíciu, na ktorej sa bude ofsajdová čiara nachádzať, je jednoduché určiť vďaka polu prvkov, ktoré obsahuje pozície hráčov. Pole použitých prvkov je výsledkom detekcie objektov v snímke a mapovania súradníc obrazov do súradníc scény.

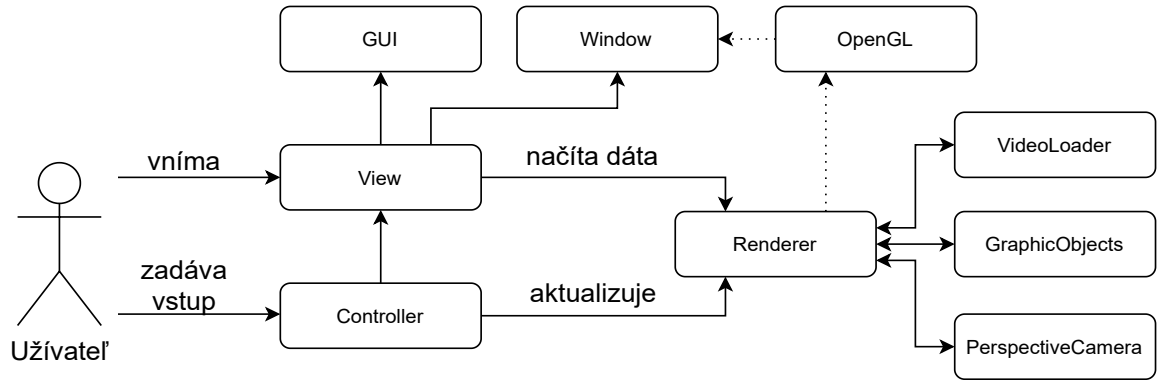
Určenie východiskového bodu na zobrazenie ofsajdovej čiary je možné takisto manuálnym spôsobom zadaním pomocou kurzoru myši. Pri aktivácii takéhoto zobrazenia ofsajdovej čiary bude pozícia východiskového bodu kopírovať pozíciu kurzoru.

3.5 Návrh aplikácie

Cieľom tejto časti práce je opísať vstupné parametre na aplikáciu a navrhnúť vhodnú aplikačnú štruktúru.

Aplikácia bude spracovávať vstupné video športového zápasu tvorené kamerovým systémom spoločnosti Camvision. Tento kamerový systém vytvára panoramatickú snímku celého ihriska. Na vypracovanie tejto práce boli použité tri voľne dostupné panoramatické videozáznamy.

Aplikačná štruktúra je tvorená podľa softvérovej architektúry model – view – controller (MVC), ktorá rozdeľuje jednotlivé časti do čiastkových komponentov. Prvý komponent určuje dátový model obsahujúci dáta jednotlivých nástrojov (definíciu grafických objektov a polí na ukladané pozície hráčov). Komponent view predstavuje užívateľské rozhranie potrebné na demonštráciu použitia navrhovaných nástrojov. Posledný komponent controller zahŕňa aplikačnú logiku, riadi chod aplikácie a zároveň reaguje na užívateľské vstupy.



Obr. 3.3: Diagram aplikačnej štruktúry

Kapitola 4

Demonštračná aplikácia

V tejto kapitole bakalárskej práce je prezentované demonštračné použitie navrhovaných nástrojov na základe teoretickej časti práce v kapitole 2 a návrhu nástrojov v kapitole 3.

Aplikáciu demonštrujúcu navrhované nástroje tvorí pre každú snímku videa opakujúca sa sekvencia troch krokov: zadanie vstupu, spracovanie a zobrazenie. Systém najskôr spracuje vstup zadaný užívateľom, adekvátne načíta zobrazované objekty, z ktorých generuje výslednú scénu zobrazovanú v aplikačnom okne. Počas fázy spracovania vyhodnocuje systém zadaný vstup a vyberie objekty (grafické objekty a video) na zobrazenie.

Aplikácia bola implementovaná modulárnym spôsobom s využitím objektovo orientovaného prístupu. Hlavným programovacím jazykom je jazyk C++¹ a jazyk GLSL použitý na programovanie grafickej karty s použitím knižnice OpenGL. Všetky použité technológie v implementácii práce sú popísané v časti 4.1.

Počas implementácie programovej časti tejto práce sa zistilo, že knižnice OpenCV a OpenGL nepoužívajú rovnaký súradnicový systém a zároveň zápis údajov v transformačných maticiach používa rozličnú notáciu, čo spočiatku sťažilo implementačnú časť práce. Pomerne náročnou úlohou pri implementácii tejto práce bolo všetky rozdiely identifikovať a podrobnejšie naštudovať prevod medzi jednotlivými notáciami. Zistil som, že pri OpenCV je používaná notácia riadkovej priority a orientácia osí súradnicového systému je používaná v postupnosti $[X, Y, Z]$. Na rozdiel od notácie OpenGL, ktorá používa maticovú notáciu stĺpcovej priority a postupnosť súradnicového systému osí je v tvare $[X, Z, Y]$. Rovnako sa vyskytujú rozdiely pri indexovaní počiatku súradnicového systému OpenCV snímok a OpenGL textúr.

4.1 Použité technológie

Všetky použité knižnice v tejto práci sú voľne dostupné. Na analýzu nasnímaného videa som sa rozhodol použiť knižnicu OpenCV pre jej nekomplikované použitie a podrobnú dokumentáciu. Na zobrazenie grafických objektov som použil knižnicu OpenGL, ktorá je zároveň použitá v existujúcom riešení softvéru spoločnosti Camvision. Zároveň použitie kombinácie knižníc OpenCV a OpenGL na zobrazovanie v reálnom čase sa ukázalo ako vhodné a v tejto aplikácii nie je potrebné použiť ďalšie externé knižnice. Hoci prehrávanie videa použitím knižnice OpenCV trpí nedostatkom spätného prehrávania, ktoré by bolo vhodné na reálne použitie aplikácie, táto knižnica umožňuje súbežnú analýzu snímky a

¹Dostupné z: <https://docs.microsoft.com/en-us/cpp/?view=msvc-160>

zároveň jej zobrazenie pri jedinom načítaní snímky. Implementácia spätného prehrávania videa nebola pre náročnosť práce z časových dôvodov možná.

OpenCV

Knižnica OpenCV[1] je zameraná na implementáciu algoritmov používaných pri spracovávaní obrazu v rámci odboru počítačového videnia. Táto open-source multiplatformová knižnica je navrhnutá na spracovanie snímok v reálnom čase. Knižnica je použitá vo verzii 4.5.0.

OpenGL

Multiplatformová knižnica OpenGL² špecifikuje rozhranie softvérových technológií k hardvérovej grafickej karte, na ktorej prebiehajú akcelerované výpočty na vykreslenie scény. Knižnica je zameraná výhradne na vykresľovanie dvoj- a trojrozmernej počítačovej grafiky na základe jazyka GLSL. V bakalárskej práci je použitá verzia OpenGL 4.6.

GLFW

GLFW³ je multiplatformová knižnica, ktorá vytvára systémové okná a kontext vykresľovania grafiky pre OpenGL. Knižnica taktiež obsahuje rozhranie pre užívateľské vstupy z klávesnice a myši vo forme spätných odoziev (tzv. callbackov). Knižnica je používaná vo verzii 3.3.2.

GLM

Ďalšia používaná knižnica GLM⁴ poskytuje štandardnú definíciu tried pre matice, vektory a matematické operácie medzi nimi. Knižnica je navrhnutá v súlade s matematickými operáciami používanými v jazyku GLSL a je dostupná ako open-source pod licenciou MIT.

Assimp

Knižnica Assimp⁵ slúži na načítanie objektových súborov popisujúcich vrcholy, normály a UV mapy do vektorových štruktúr vhodných na efektívne načítanie pomocou grafickej karty.

Dear ImGui

ImGui⁶ je grafická multiplatformová knižnica na tvorbu grafického užívateľského rozhrania grafických knižníc OpenGL, DirectX, Metal alebo Vulkan. Definuje rozhranie na použitie s rozličnými typmi kontextov, medzi ktoré patria napríklad okenné kontexty GLFW, SDL, OSX alebo Win32.

²Dostupné z: <https://www.opengl.org/>

³Dostupné z: <https://www.glfw.org/>

⁴Dostupné z: <https://glm.g-truc.net/0.9.9/index.html>

⁵Dostupné z: <https://www.assimp.org/>

⁶Dostupné z: <https://github.com/ocornut/imgui>

4.2 Dôležité triedy

V nasledujúcej časti bakalárskej práce sa nachádzajú implementačné detaily jednotlivých súčastí demonštračnej aplikácie. Všetky triedy popísané v tejto časti práce pozostávajú zo zdrojových súborov s príponou .cpp a hlavičkových súborov s príponou .hpp.

Trieda Controller

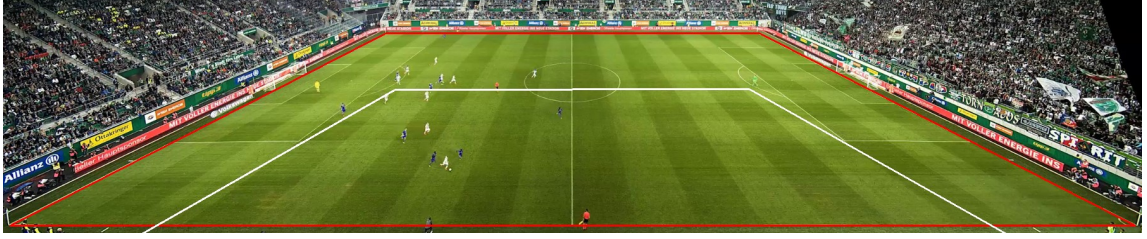
Trieda **Controller** zabezpečuje inicializáciu všetkých potrebných zdrojov aplikácie, medzi ktoré patrí inicializácia kontextu okna, grafickej knižnice OpenGL a ďalších potrebných modulov. V rámci tejto triedy je taktiež realizované spracovanie užívateľského vstupu pomocou myši a klávesnice. Riadenie demonštračnej aplikácie je vykonávané v rámci hlavnej slučky, kde sa odohráva volanie riadiacich funkcií jednotlivých modulov. Ak je pri spustení demonštračnej aplikácie použitý parameter „-p“, riadenie hlavnej slučky vykoná výkonnostné testy nad jednotlivými blokmi.

V rámci triedy **Controller** je realizovaná kalibrácia transformačnej matice pozorovateľského priestoru a rekonštrukcia 3D scény pomocou homografie.

Transformačnú maticu pozorovateľského priestoru, ktorá je zhodná s maticou vonkajších parametrov kamery (rovnice 2.13), som vytvoril použitím OpenCV funkcie **SolvePnP** a otestoval rôzne metódy na základe časti 2.6 pričom metóda určujúca najpresnejší odhad pozície kamery bola Iteratívna metóda [1]. Funkcia na základe obrazových a objektových bodov a nastavenia kamery odhadne vektor rotácie *rvec* a vektor translácie *tvec* kamery, ktorý môžeme použiť na zostavenie transformačnej matice.

V prvom kroku zostrojíme transformačnú maticu rotácie R vo veľkosti 3×3 na základe vektoru rotácie vo veľkosti 1×3 . Zostavenie je možné využitím OpenCV funkcie **Rodrigues**. Následne zostrojíme transformačnú maticu pozorovateľského priestoru na základe spojenia transformačnej matice rotácie R a vektora translácie *tvec* do rozšíreného homogénneho tvaru vo veľkosti 4×4 . Keďže je výpočet vykonaný pomocou OpenCV funkcií, na použitie transformačnej matice v prostredí OpenGL je potrebné najskôr invertovať os Y a Z a následne maticu transponovať. Takto zostrojenú transformačnú maticu pre OpenGL zobrazenie označujeme R_{GL} .

Na nasledujúcich obrázkoch je znázornené zobrazenie pomocou jednotlivých výpočtov transformačnej matice. Pomocou červenej čiary je zobrazený okraj športového ihriska daný manuálnym zadaním jeho rohov. Biela čiara zobrazuje okraje ihriska pomocou rozličných daných výpočtov. Pri každom obrázku je pridané označenie výpočtu a priemerná chyba zobrazenia. Táto chyba sa udáva v pixeloch a je určená ako priemer celkovej vzdialenosti medzi manuálne zadaným bodom rohu ihriska a zodpovedajúcim premietnutým rohom ihriska.



(a)



(b)

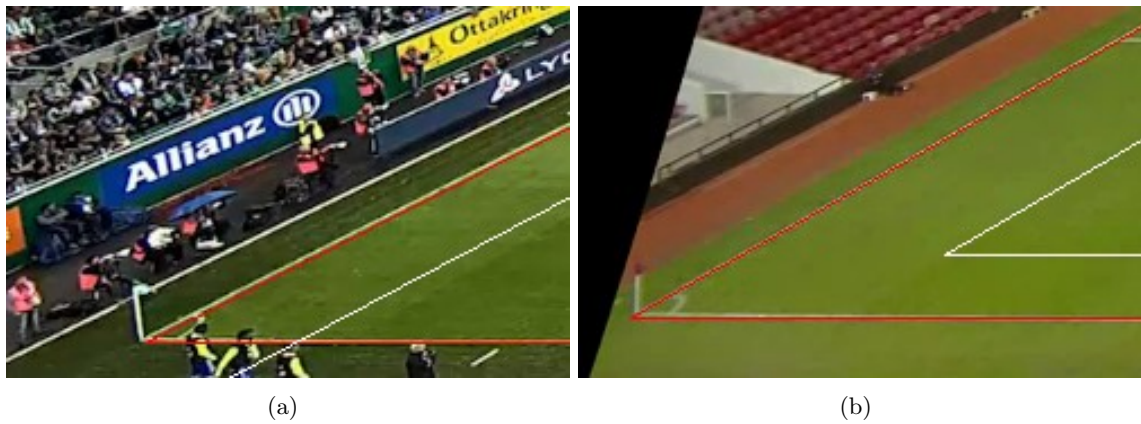
Obr. 4.1: Biela čiara na obrázkoch znázorňuje virtuálne ihrisko zobrazené pomocou transformačnej matice pozorovateľského priestoru R_{GL} . Priemerná chyba zobrazenia pre roh ihriska je pre (a) 114.00 pixelov, (b) 108.17 pixla.

Z obrázku 4.1 je možné pozorovať, že výsledné zobrazenie virtuálneho ihriska do snímky pomocou R_{GL} sa nepribližuje k očakávanému zobrazeniu, keď by sa rohy ihriska mali namapovať na manuálne zadané rohy ihriska. Môžeme si však všimnúť že perspektívne zobrazenie virtuálneho ihriska sa geometricky podobá reálnemu ihrisku.

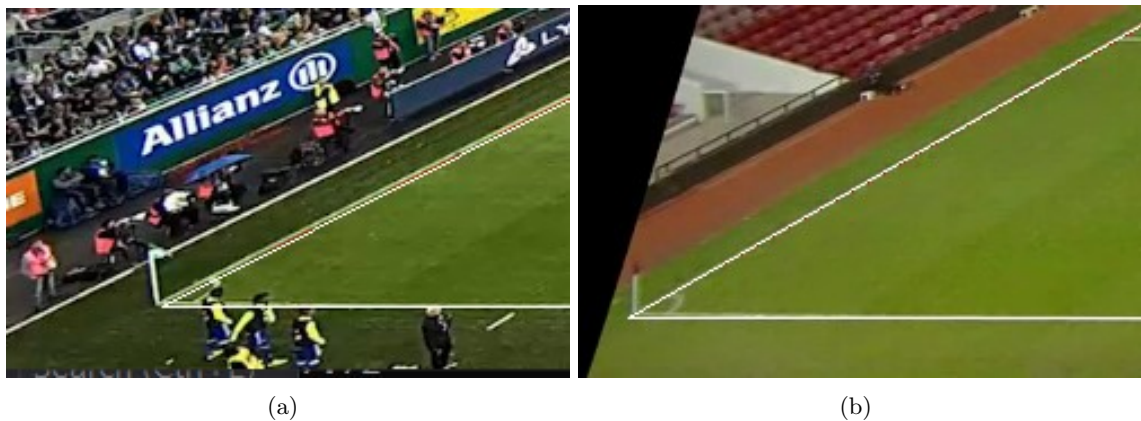
Výslednú maticu R_{GL} som preto iteratívne upravoval pomocou translácie podľa osí Y (ďalej ako O1) a Z (ďalej ako O2) danej maticou T , pričom t_y je posun po osi Y a t_z je posun po osi Z . A rotácie okolo osi X danej maticou $R_x(\theta)$ pričom θ je uhol rotácie (ďalej ako O3). Cieľom použitia týchto operácií bolo znížiť chybu reprojekcie. V prípade, že úprava matice pomocou operácií (pre O1 koeficient t_y , pre O2 koeficient t_z a pre O3 uhol $theta$) začne chybu reprojekcie zvyšovať, je potrebné hodnoty týchto koeficientov postupne znižovať, dokým žiadna úprava nezmenší reprojekčnú chybu.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} \quad (4.1)$$

Videozáznamy, do ktorých zobrazujeme virtuálne objekty, sníma kamera pod klesajúcim náklonom, a preto je v rámci transformačnej matice podstatná len rotácia okolo osi X . Pozície v rámci transformačnej matice, ktoré nie sú ovplyvnené rotáciou okolo osi X , preto vynulujeme, čím sa zlepši chyba reprojekcie medzi výpočtami v obrázkoch 4.1 a 4.2.



Obr. 4.2: Biela čiara na obrázkoch zobrazuje jednoduché vylepšenie reprojekcie transformačnej matice pomocou odstránenia nežiaducich rotácií s priemernou chybou (a) 29.10 pixla (b) 90.69 pixla.



Obr. 4.3: Biela čiara na obrázkoch predstavuje manuálne nastavenie transformačnej matice po odstránení nežiaducich rotácií a aplikovaní používaných úprav na čo najmenšiu priemernú chybu, ktorá je pre (a) 2.50 pixla, (b) 2.06 pixla (červené čiary sú na týchto obrázkoch takmer celé prekryté bielymi).

Z obrázkov vidieť, že transformovaním matice R_{GL} pomocou operácií O1, O2 a O3 som minimalizoval chybu zobrazenia. Ďalej je potrebné určiť akceptovateľné hodnoty pre chybu zobrazenia, pri ktorej nebude zmena pozície objektov ovplyvňovať celkový dojem zobrazenia scény. Objekty, ktoré sa zobrazujú na okrajoch ihriska, sú citlivejšie na reprojekčnú chybu a nesprávne zobrazenie sa v tomto prípade začne prejavovať pri nižšej chybe. Túto situáciu môžeme pozorovať na nasledujúcom obrázku, keď pri reprojekčnej chybe 25.00 pixla sa objekt zvýrazňujúci hráča začína zobrazovať mimo pozície daného hráča.



Obr. 4.4: Na obrázku môžeme vidieť zobrazenie zvýraznenia hráčov pri priemernej chybe reprojekcie nastavenej na hodnotu 25.00 pixla.

V súlade s vytvoreným postupom manuálneho hľadania najnižšej reprojekčnej chyby použitím operácií O1, O2 a O3 navrhujem algoritmus automatického vylepšenia matice R_{GL} . Výsledkom algoritmu sú hodnoty koeficientov pre operácie O1, O2 a O3, ktorými minimalizujeme reprojekčnú chybu matice R_{GL} . Algoritmus musí znižovať koeficient zmeny pre operácie O1 a O2, aby použitie operácií viedlo k zníženiu reprojekčnej chyby. Krok algoritmu sa bude opakovať, pokiaľ koeficient zmeny pre operácie O1 a O2 bude vyšší ako jedna desať tisícina. Oproti manuálnym úpravám bude algoritmus zmenený v prípade použitia operácie O3, keď sa pri každej zmene pomocou operácií O1 alebo O2 použije operácia O3 s hodnotou uhla θ , ktorý je určený uhlom medzi priamkami obrazových a objektových bodov (priamka č. 1 je tvorená obrazovými bodmi s indexom 0 a 1 a priamka č. 2 je tvorená objektovými bodmi s rovnakými indexmi). Touto úpravou je zabezpečená správna rotácia kamery podľa osi X pri jej výškovej a hĺbkovej pozícii a je vypočítaná pomocou funkcie `GetRotationToParallel`.



Obr. 4.5: Biela čiara na obrázkoch zobrazuje reprojekciu bodov na základe navrhnutého vylepšenia transformačnej matice s priemernou chybou (a) 4.79 pixla, (b) 4.18 pixla.

Trieda View

Trieda `View` slúži na abstrakciu prezentačnej vrstvy pomocou aplikačného okna `GLFW`, ktoré je použité na renderovanie objektov vytváraných prostredníctvom `OpenGL` funkcií.

Algoritmus 1: Výpočet matice pozorovateľského priestoru.

```
Input:  $R_{GL}$  matica(4x4) pozorovateľského priestoru
Input: O1.value – posun  $Y$ , O2.value – posun  $Z$ , O3.value – rotácia osi  $X$ 
1 minError = DBL_MAX, newError = DBL_MAX
2 O1.value = 0, O1.sensitivity = 1.0
3 O2.value = 0, O2.sensitivity = 1.0
4 O3.value = 0, O3.sensitivity = 1.0
5 while true do
6    $O_n$ , newError, valueD = GetBestOperation( $R_{GL}$ , O1, O2, O3)
7   if newError < minError then
8     minError = newError
9      $O_n$ .value += valueD
10    O3.value += GetRotationToParallel()
11  end
12  else
13     $O_n$ .sensitivity /= 10
14  end
15  if  $O_n$ .sensitivity < 0.0001 then
16    break
17  end
18 end
```

Umožňuje aplikácii pristupovať k užívateľskému vstupu pomocou callbackov, ktoré sú realizované knižnicou GLFW a viažu sa na aplikačné okno. Trieda ďalej vytvára grafické užívateľské rozhranie pre dané aplikačné okno.

Trieda VideoLoader

Trieda `VideoLoader` načíta video pomocou funkcie `InitVideoFrame`, ktorá využíva OpenCV triedu `VideoCapture` na načítanie videa zo súboru. Načítaná snímka je následne prevedená do OpenGL textúry pomocou funkcie `MatToTexture`. Na prvej snímke videa je inicializovaná textúra OpenGL pomocou funkcie `glTexImage2D`. Aby sa urýchlila exekúcia aplikácie a zamedzilo inicializácii a mazaniu textúry pre každú snímku videa, bolo pre všetky následné snímky použité aktualizovanie textúry funkciou `glTexSubImage2D`. Štruktúra OpenGL textúry je následne zmazaná až po ukončení prehrávania videa. Zobrazenie snímky v prostredí OpenGL okna zabezpečuje `DrawTexture`. Počas inicializácie vrcholov OpenGL textúry treba brať do úvahy indexáciu OpenCV snímok, ktorých počiatok súradnicovej sústavy sa nachádza v pravom hornom rohu. Pri OpenGL textúre je počiatok súradnicovej sústavy nastavený na pravý dolný roh.

V rámci triedy `VideoLoader` je implementovaná detekcia hráčov v snímke, keďže trieda priamo disponuje prístupom k videosnímke. Podľa návrhu detekčného algoritmu v časti 3.2 je detekcia a vytvorenie masky vytvorená nasledujúcim spôsobom. Funkcia detegujúca hráčov sa nazýva `FindPlayerPositions` a je realizovaná pomocou Gaussian mixture modelu (časť 2.3) implementovaného pomocou OpenCV triedy `BackgroundSubtractor`, pričom deteguje objekty popredia volaním metódy `apply`. Pomocou funkcie `findContours` sú vyhľadane obrisy detegovaných objektov s ich pozíciou v analyzovanom snímku. Či je daný objekt hráč a do ktorého patrí tímu, sa vyhodnocuje podľa počtu pixelov zhodných s jeho

farbou dresu (potrebných aspoň 15 pixelov na pozitívne vyhodnotenie), teda prevedením oblasti skúmaného obrysu do spektra farieb HSV(Hue, Saturation, Value) pomocou funkcie `inRange` a `countNonZero` určuje počet pixelov, ktoré vyhovujú danému spektru farieb. Farbu dresu jednotlivých tímov je potrebné manuálne zadať pri otvorení analyzovaného videa. V prípade pozitívneho určenia, že skúmaný objekt je hráč, uloží sa pozícia obrysu ako (x, y) súradnica stredu spodnej hrany obrysového štvorca. Do novej snímky s nulovou inicializačnou hodnotou je následne pridaný detegovaný obrys podľa jeho pôvodnej pozície. Táto snímka je použitá na vytvorenie alfa masky.

Funkcia `CreatePlayerMask` vytvára alfa masku pre obrisy detegovaných hráčov. Jej úloha je spojenie masky hráčov a modelu popredia, ktorým je vstupná snímka. Výsledný čierny obraz s farbenými hráčmi je prevedený do OpenGL textúry a zobrazený v popredí objektov.

Trieda `Match`

Trieda obsahuje objekty tímov s hráčmi, ktorým prideluje detegované polohy hráčov na základe triedenia podľa najmenšej vzdialenosti. Tvorba štruktúr hráčov a pridelovanie polôh vychádza z detekčného algoritmu [11]. V prvej snímke prebieha vytvorenie hráčov na základe detekcie v danej snímke a v ostatných snímkach sú použité nasledujúce operácie:

- Aktualizácia polohy hráčov
- Vymazanie hráčov s neaktívnou polohou
- Vytvorenie nových hráčov

Trieda `GraphicObject`

Trieda `GraphicObject` definuje grafické objekty, ktorých vrcholy, indexovanie vrcholov, normály a uv indexy sú načítané pomocou triedy `ModelLoader` a uložené do štruktúry `Mesh` a OpenGL objektu `VertexArray`. Vytvára inštancie tohto istého typu objektu, pričom inštancia uchováva modelovú maticu a materiál. Hlavnou metódou tejto triedy je metóda `Draw`, ktorá pomocou unikátneho identifikátora objektu `VertexArray` a počtu vrcholov vykreslí daný objekt za použitia jediného volania OpenGL funkcie `glDrawArrays`.

Trieda `Shader`

Trieda `Shader` slúži na použitie shader programov. Shader program pozostáva z vertex a fragment programu, ktoré sú načítané z textových súborov do reťazovej reprezentácie, s ktorou pracuje OpenGL. Vytvorenie a kompilácia shader programov bola použitá podľa návodu⁷. Trieda ďalej uľahčuje nastavenie jednotlivých uniformných hodnôt.

V tejto práci bolo použitých šesť shader programov. Patria medzi ne shadere na zobrazenie videa ako textúry pozadia, masky hráčov, grafických objektov s tieňmi, ďalej vytvorenie tieňovej mapy a jej vizuálne zobrazenie a napokon shader na vizualizáciu tieňov v rámci textúry pozadia, ktorého funkčnosť nie je súčasťou implementácie.

⁷Dostupné z: <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-2-the-first-triangle>

4.3 Ukážka implementácie virtuálnej grafiky

Implementáciu grafických nástrojov ilustrujem v nasledujúcej časti pomocou obrázkov a popisu animácie. Na obrázkoch sú zobrazené snímky virtuálnych objektov, ktoré sú tvorené daným nástrojom.

Zvýraznenie hráčov

Zvýrazňovanie hráčov predstavuje kruhová obruč s výrezmi. Animáciu tohto objektu vytváram neustálym rotovaním obruče v čase čím vytváram poloautomatickú animáciu, ktorá je prehrávaná vždy, kedy je povolené vykreslenie tohto objektu. Pri zobrazovaní tohto objektu je pre každý snímok nastavená rotácia na základe príkazu, pričom `rotationInTime` je rotácia v intervale $\langle 0,360 \rangle$ za čas troch sekúnd. Návrátová hodnota funkcie `glfwGetTime` je hodnota časovača, ktorý bol inicializovaný pri spustení aplikačného okna.

```
rotationInTime = 120 * fmodf(glfwGetTime(), 3)
```



Obr. 4.6: Ukážka nástroja zvýraznenie hráčov.

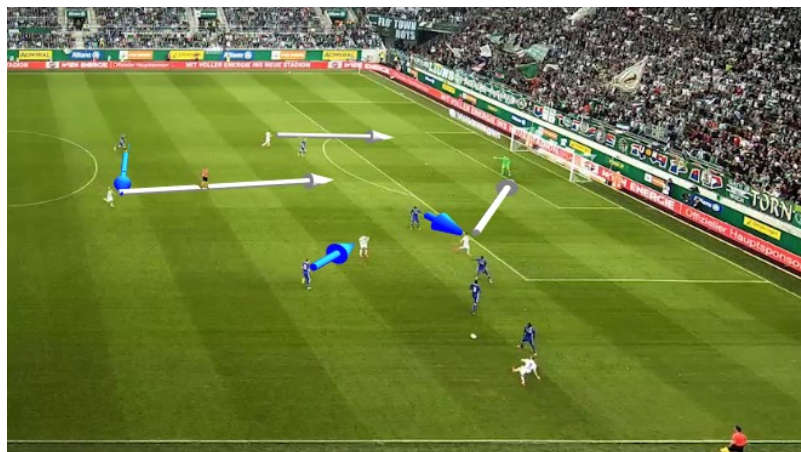
Orientovaný smerovník

Animácie orientovaného smerovníku sú vytvorené na základe poloautomatickej animácie. Vytvoril som dve varianty animovania smerovníka. Prvý variant je automatický, kedy je pri vytváraní smerovníku špecifikovaný počiatočný bod smerovníka pomocou myši. Až do potvrdenia výberu koncového bodu smerovníka druhým kliknutím myši, je smerovník animovaný posunom koncového bodu, ktorý kopíruje pozíciu kurzoru. Pre správne zobrazenie sú pozície kurzoru v rámci obrazovky konvertované do súradníc obrazovky.

Druhou animáciou smerovníka je takzvaný rast alebo predlžovanie veľkosti, kedy animácia znázorňuje predlžovanie smerovníka z počiatočného do koncového bodu.

Ofsajdová čiara

Animovanie ofsajdovej čiary som vytvoril ako interaktívne kontrolovanie pomocou užívateľského vstupu. Pri aktivácii tohto nástroja sa pozastaví video a zobrazí sa virtuálny grafický prvok ofsajdovej čiary, ktorý následne kopíruje x súradnicu kurzoru transformovanú do súradníc scény. Virtuálna ofsajdová čiara je potom zobrazená po celej šírke ihriska v určenom



Obr. 4.7: Ukážka nástroja orientovaný smerovník. Smerovníky na obrázku znázorňujú obrannú pozíciu hráčov v modrých dresoch a útočiacu pozíciu hráčov v bielych dresoch.

mieste. Po inicializovaní ofsajdovej čiary je možné povoliť rozšírenú automatickú animáciu, ktorá vyhledá predposledného brániaceho hráča, ktorý určuje ofsajdovú pozíciu.



Obr. 4.8: Ukážka nástroja ofsajdová čiara, ktorý zvýrazňuje ofsajdovú pozíciu bieleho hráča.

Virtuálne ihrisko

Dôležitou súčasťou tejto práce je zobrazenie virtuálneho ihriska. Znázornil som ním potrebu výpočtu pozície kamery, kedy po otvorení videozáznamu prebehne automatická animácia v rámci ktorej virtuálna kamera mení svoju pozíciu. Na počiatku animácie a zároveň pri otvorení aplikácie kamera pozoruje virtuálne ihrisko z výšky pričom sa ihrisko zobrazuje ako obdĺžnik. Kamera následne začne meniť svoju pozíciu a smer pohľadu, čím je tvar zobrazenia ihriska menený z obdĺžnikového na lichobežníkový (vzdialená hrana lichobežníka je kratšia). Zobrazenie virtuálneho ihriska je nastaviteľné pomocou užívateľského rozhrania a môže byť aktívne po celú dobu používania nástrojov.

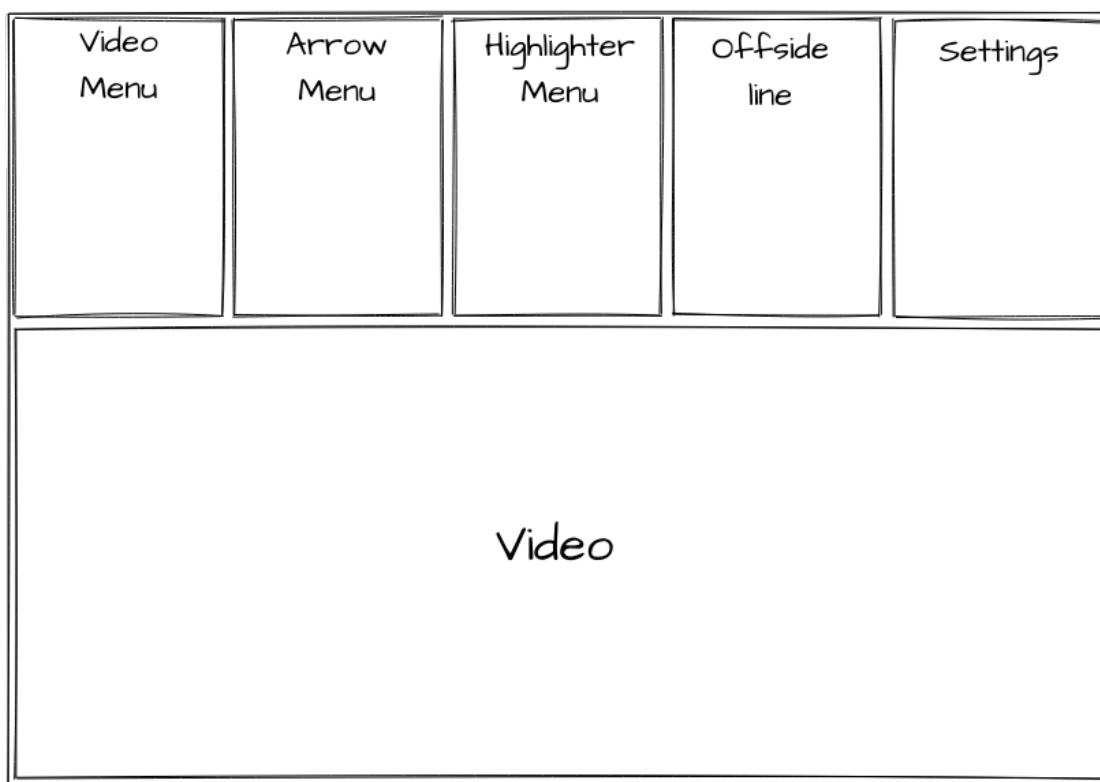
Aplikácia prostredníctvom grafického užívateľského rozhrania umožňuje výber zobrazení scény (menu s názvom settings), medzi ktoré patrí zobrazenie s hĺbkovou maskou, zobrazenie masky s detegovanými hráčmi, zobrazenie scény v rámci rozšírenej reality a zobrazenie

virtuálnej scény. V rámci virtuálnej scény môžeme meniť pozíciu a orientáciu kamery, čím je umožnený pohyb po virtuálnej scéne.

4.4 Užívateľské rozhranie

Grafické užívateľské rozhranie je tvorené aplikačným oknom, v ktorom sa nachádzajú ponuky ovládania jednotlivých nástrojov, ponuka na výber rozličných metód výpočtu transformačnej matice pozorovateľského priestoru, ponuka na manipuláciu s videom a zároveň priestor pre samotné video analyzovaného športového zápasu.

V prvotnom štádiu tvorby grafického užívateľského rozhrania jednotlivé prvky prekryvali priestor videa a bolo umožnené minimalizovanie ich okien. Toto použitie sa ukázalo ako nevhodné z dôvodu potreby súčasného ovládania nástrojov a práce s priestorom videa. Na odstránenie tohto nedostatku som rozšíril aplikačné okno a vytvoril samostatný priestor pre ponuky ovládania jednotlivých nástrojov. Znázornenie navrhnutého užívateľského rozhrania sa nachádza na nasledujúcom obrázku, ktorý je vytvorený pomocou nástroja MockFlow⁸.



Obr. 4.9: Skica znázorňujúca grafické užívateľské rozhranie.

Ovládanie aplikácie je tiež možné pomocou klávesnice, pričom medzerníkom pozastavíme alebo spustíme prehrávanie videa. Tlačidlom Escape uzatvoríme aplikáciu. A nasledujúcimi klávesami je možné meniť pozíciu kamery vo virtuálnej scéne: w,s – hore, dole; a,d – doprava, doľava; q,e – priblíženie, oddialenie; y,x – rotácie podľa osi X, c,v – rotácie podľa osi Y, i,o – zväčšenie, zmenšenie citlivosti posunu kamery.

⁸Dostupné z: <https://www.mockflow.com/>

4.5 Testovanie a výsledky

Na potreby testovania s cieľom zistiť výkonnosť aplikácie v reálnom čase som vytvoril výkonnostné testy spustiteľné pomocou argumentu „-p“. Testy sumarizujú zaťaženie výpočtovej jednotky podľa jednotlivých blokov a sú vykonané na videozázname s veľkosťou 1980 x 396 pixelov s počtom snímok 1792. Ďalej uvádzam zoznam jednotlivých blokov s ich priemerným časom spracovania:

- Vykreslenie užívateľského rozhrania: 0,6296 ms na obrázok.
- Kalibrácia a vylepšenie pozície kamery: 181.6544 ms. Tento blok je vykonaný jediný raz pri inicializácii analyzovaného videa.
- Načítanie snímky videa zo súboru a detekcia hráčov v snímke: 42.5109 ms na obrázok.
- Príprava grafických objektov na zobrazenie: 3.6694 ms na obrázok.
- Vykreslenie videosnímky: 0.0027 ms na obrázok.
- Vykreslenie virtuálnych objektov: 0.2751 ms na obrázok.
- Vykreslenie masky hráčov do popredia objektov: 0.0030 ms na obrázok.
- Čas uspatia programu, ktorý slúži na obmedzenie zobrazovania podľa frekvencie snímok videa: 0.4995 ms na obrázok.
- Priemerný čas spracovania jednej snímky: 48.0813 ms.

Z nameraných výsledkov môžeme pozorovať, že priemerné spracovanie videosnímky je 48.0913 ms, čo zodpovedá spracovaniu 20,79 snímok za sekundu, pričom frekvencia snímok v analyzovanom videu je 25 snímok za sekundu. Zobrazovanie videa s použitím vytvorených nástrojov je preto pomalšie ako jeho skutočná rýchlosť. Najväčšie zaťaženie procesoru je počas načítania videosnímky a jej spracovania na detekciu hráčov. Ďalej môžeme vidieť, že priemerný čas nečinnosti je 0.4995 ms na snímku, ktorý je uskutočnený len v prípade, že je snímka spracovaná rýchlejšie ako 1/25 sekundy. Nízky čas zobrazovania grafických objektov je daný tým, že riadenie programu definuje potrebné štruktúry na vykreslenie objektov a nemusí čakať na samotné dokončenie vykreslenia. Keďže zdroje procesora boli dostupné v priemere 0,4995 ms na snímku, tento čas by mohol byť využitý zavedením vláknového spracovania pre načítanie a spracovanie videosnímkov.

Kapitola 5

Záver

Hlavným cieľom bakalárskej práce bolo vytvoriť súbor grafických nástrojov na vkladanie trojrozmernej virtuálnej grafiky do videozáznamov športových podujatí využívaných pri analýze športových podujatí, predovšetkým futbalových zápasov. Využitie nástrojov spočíva predovšetkým vo vizualizácii zozbieraných dát videozáznamov.

V bakalárskej práci som navrhol a implementoval spôsob vkladania 3D grafiky do videozáznamov futbalových zápasov na základe odhadu pozície kamery pomocou iteratívnej metódy a vylepšenia tohto výpočtu pre používané videozáznamy. Grafické nástroje vytvorené v tejto bakalárskej práci zobrazujúce virtuálnu grafiku a animácie môžu byť využité najmä pri pozápasovej analýze športového podujatia. Športový tréner alebo tímový videoanalytik môže využiť nástroje na grafické zvýraznenie situácií, ktoré sa udiali počas zápasu. Vytvorenými nástrojmi bude možné tiež modelovať taktiku tímu, zahŕňajúc rozmiestnenie hráčov na ihrisku, smer pohybu hráčov a prihrávkov a môžu byť použité aj na analýzu činnosti jednotlivého športovca.

Užívateľ môže nástroje použiť na zvýraznenie vybraných hráčov, ktorí sú ďalej trackovaní a zvýrazňovaní trojrozmernými objektmi vkladacími do videoscény. Užívateľ môže ďalej manuálne vytvárať smerovníky znázorňujúce smer pohybu hráča alebo lopty. Trojrozmerné objekty sú vytvárané perspektívnym skreslením korešpondujúcim so scénou, ktorá je zachytená kamerou. Teoretická časť bakalárskej práce predstavila prehľad odbornej literatúry potrebnej na správne vykreslenie grafických objektov do kalibrovannej scény. Praktická časť práce bola zameraná na implementáciu a vizualizáciu spomenutých nástrojov s použitím knižnice OpenGL.

Východiskovou metódou práce bolo riešenie perspektívnych bodov, pomocou ktorých som zisťoval pozíciu kamery v analyzovanom videu. V prípade panoramatických videí, ktorými som sa zaoberal, mala táto metóda rezervy. Metóda nevypočítavala presné vonkajšie parametre kamery na zobrazenie virtuálnej scény do používaných videozáznamov, a tak som vytvoril algoritmus, ktorý rozšíril tento problém a značne znížil nedostatky tohto problému.

Na demonštráciu grafických nástrojov bolo potrebné použiť detekciu a sledovanie hráčov. Keďže to nebol primárny zámer mojej práce, vybral som si metódu Gaussova zmes na detekciu a Kalmanov filter na sledovanie objektov. Takto som implementoval grafické nástroje, ktoré som navrhol.

V ďalšej časti bakalárskej práce som opísal postup návrhu aplikácie a vymedzil nástroje z hľadiska pozície hráčov, smeru pohybu hráčov a pohybu lopty a postavenia hráčov v rámci futbalových pravidiel. Hráčov som preto najskôr detegoval, na zobrazenie ich pohybu, ako aj pohybu lopty som použili orientovaný smerovník a na sledovanie správneho herného

postavenia hráčov som ako nástroj vytvoril ofsajdovú čiaru. Na určené ciele boli využité analyzačné a grafické nástroje.

Problematika mojej práce ostáva naďalej otvorená. V budúcnosti je možné jej rozšírenie detegovaním rohov ihriska, výpočtom pozície kamery pre videosnímky, ktoré nie sú panoramatické, a detegovaním pózy jednotlivých hráčov, na základe čoho by bolo možné vytvoriť virtuálnu grafickú reprezentáciu hráča, a teda kompletnú virtuálnu reprezentáciu analyzovaného športového podujatia.

Tvorba bakalárskej práce *Zobrazenie virtuálnej grafiky a animácií pre športovú analýzu* nebola pre mňa štandardnou úlohou. Dala mi veľa nového poznania a mnoho nových zručností. Zvlášť som si uvedomil fenomén času, ktorý som intenzívne venoval príprave práce, na nové poznávanie, programovanie, nekonečné vylepšovanie a overovanie navrhovaných metód.

Komplikované a rozsiahle postavené úlohy prinášali znovu a znovu mnohé nápady, čo by bolo potrebné riešiť.

Pochopil som, že mi nevyhovuje online komunikácia ani so zástupcami firmy, ktorá tému zadala, ani so školiteľom, keďže som na začiatku nevedel pomenovať všetky problémy, s ktorými som zápasil. Musel som naštudovať aj problémy, s ktorými som sa dosiaľ počas štúdia nestretol. V priebehu realizácie bakalárskej práce sme sa s mojím ústretovým a chápaným školiteľom dopracovali k tomu, že môžeme riešiť iba niekoľko parciálnych tém.

Literatúra

- [1] BRADSKI, G. a KAEHLER, A. *Learning OpenCV*. 1. vyd. O'Reilly Media, Inc., 2008. 555 s. ISBN 978-0-596-51613-0.
- [2] BUZÁSSYOVÁ, K. a JAROŠOVÁ, A. *Slovník súčasného slovenského jazyka A – G*. [online]. Veda, vydavateľstvo Slovenskej akadémie vied, 2006 [cit. 2021-07-02]. Dostupné z: <https://slovník.juls.savba.sk/?w=grafika&s=exact&c=a89e&cs=&d=sss#>.
- [3] DE PAOLIS, L. T. a MONGELLI, A. *Augmented and Virtual Reality* [online]. 1. vyd. Springer, 2014, sv. 8853, s. 466. ISBN 978-3-319-13969-2. Dostupné z: <https://doi.org/10.1007/978-3-319-22888-4>.
- [4] GAO, X.-S., HOU, X.-R., TANG, J. a CHENG, H.-F. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2003, zv. 25, č. 8, s. 930–943. DOI: 10.1109/TPAMI.2003.1217599. Dostupné z: <https://doi.org/10.1109/TPAMI.2003.1217599>.
- [5] GUHA, S. *Computer Graphics Through OpenGL* [online]. 3. vyd. Taylor & Francis Group, 2019. 731 s. ISBN 978-0-429-46417-1. Dostupné z: <https://doi.org/10.1201/9780429464171>.
- [6] HARTLEY, R. a ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2. vyd. Cambridge University Press, 2004. 655 s. ISBN 978-0-521-54051-3.
- [7] HEARN, D. D., BAKER, M. P. a CARITHERS, W. R. *Computer Graphics with OpenGL*. 4. vyd. Pearson, 2010. 812 s. ISBN 978-0-13-605358-3.
- [8] LEPETIT, V., MORENO-NOGUER, F. a FUA, P. EPnP: An Accurate $O(n)$ Solution to the PnP Problem. *International Journal of Computer Vision*. 2009, zv. 81, č. 2, s. 155–166. DOI: 10.1007/s11263-008-0152-6. Dostupné z: <https://doi.org/10.1007/s11263-008-0152-6>.
- [9] MACKOWIAK, S., KONIECZNY, J., KURC, M. a MAĆKOWIAK, P. Football Player Detection in Video Broadcast. In: BOLC, L., TADEUSIEWICZ, R., CHMIELEWSKI, L. J. a WOJCIECHOWSKI, K. W., ed. *ICCVG*. Springer, September 2010, sv. 6375, s. 118–125. DOI: 10.1007/978-3-642-15907-7-15. ISBN 978-3-642-15907-7.
- [10] MANORANJAN, P., HAQUE, S. a CHAKRABORTY, S. Human detection in surveillance videos and its applications - A review. *EURASIP Journal on Advances in Signal Processing*. 1. vyd. November 2013, zv. 2013, č. 1, s. 25. DOI: 10.1186/1687-6180-2013-176.

- [11] MATHWORKS. *Motion-Based Multiple Object Tracking: User's Guide (R2021a)* [online]. Natick, Massachusetts: The MathWorks Inc., 2021 [cit. 2021-07-16]. Dostupné z: <https://www.mathworks.com/help/vision/ug/motion-based-multiple-object-tracking.html>.
- [12] NAEMURA, M., FUKUDA, A., MIZUTANI, Y., IZUMI, Y., TANAKA, Y. et al. Morphological segmentation of sport scenes using color information. *IEEE Transactions on Broadcasting*. 1. vyd. 2000, zv. 46, č. 3, s. 181–188. DOI: 10.1109/11.892154.
- [13] ŘÍHA, K. a HUJKA, P. *Epipolární geometrie* [online]. Elektrověue, 2005 [cit. 2021-07-08]. Dostupné z: <http://www.elektrověue.cz/clanky/05017/index.html>.
- [14] RODRIGUES, J., CARDOSO, P. J. S., VILAS, T., SILVA, B., RODRIGUES, P. et al. *A Computer Vision Based Web Application for Tracking Soccer Players* [online]. 1. vyd. Springer International Publishing, 2014, s. 450–462. ISBN 978-3-319-07437-5. Dostupné z: https://doi.org/10.1007/978-3-319-07437-5_43.
- [15] SONKA, M., HLAVAC, V. a BOYLE, R. *Image Processing, Analysis, and Machine Vision*. 4. vyd. Cengage Learning, 2014. 920 s. ISBN 978-1-133-59360-7.
- [16] SZELISKI, R. *Computer Vision: Algorithms and Applications* [online]. 1. vyd. Springer, 2010. 957 s. ISBN 978-1-84882-935-0. Dostupné z: <https://szeliski.org/Book/1stEdition.htm>.
- [17] ZIVKOVIC, Z. *Improved adaptive Gaussian mixture model for background subtraction* [online]. IEEE, 2004, sv. 2, s. 28–31. ISBN 0-7695-2128-2. Dostupné z: <https://doi.org/10.1109/ICPR.2004.1333992>.

Príloha A

Obsah priloženého pamäťového média

- source – Adresár so zdrojovými súbormi aplikácie.
- source/x64/Release/ – Adresár so spustiteľnými súbormi aplikácie.
- source/x64/Release/graphics/videos – Adresár s vybranými videozáznamami.
- BP-xocena06.pdf – Textová časť bakalárskej práce.
- BP-latex.zip – Zdrojové súbory textovej časti bakalárskej práce.
- video.mp4 – Videoukážka demonštračnej aplikácie.

Príloha B

Postup inštalácie a spustenia

Prvým spôsobom prekladu zdrojových súborov je spustenie Visual Studio 2019 Solution zo súboru `source/Engine.sln`, pričom je potrebné vybrať konfiguráciu Release x64.

Druhý spôsob prekladu zdrojových súborov je použitie nástroja CMake¹, pričom parameter `-S` určuje priečinok so zdrojovými súbormi a `-B` priečinok s výstupnými preloženými súbormi. Nástroj CMake je možné použiť aj pomocou grafického užívateľského rozhrania.

Pre unixové systémy je použitie nasledujúcich príkazov, pričom prvý príkaz slúži na inštaláciu OpenGL knižníc, ak neboli vopred nainštalované.

```
$ sudo apt install libglu1-mesa-dev freeglut3-dev
$ cmake -S "~/source" -B "C:/BP-xocena06-build"
      -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Release
$ make
```

Príklad pre systém Windows kde je cieľovou verziou Visual Studio 2019 nasledujúci príkaz:

```
$ cmake -S "~/source" -B "C:/BP-xocena06-build"
      -G "Visual Studio 16 2019"
```

Spustenie aplikácie na systéme Windows použitím priložených spustiteľných súborov je možné spustením súboru `source/x64/Release/Engine.exe`, pričom priečinok so spustiteľnými súbormi obsahuje všetky potrebné knižnice na spustenie prostredníctvom tohto systému. Spustenie na iných systémoch vyžaduje preklad aplikácie pomocou nástroja CMake.

¹Dostupné z: <https://cmake.org/>