

System pre analýzu a animáciu chôdze človeka
Dokumentácia

Vedúci projektu: doc. Ing. Martin Šperka, PhD.
Tím č. 5 : Bc. Blanárik Ivan
Bc. Kočiš Ladislav
Bc. Kotrba Attila
Bc. Kročka Lukáš
Bc. Lenický Mário

Október, 2005

Zadanie

Analyzujte problematiku animácie pohybu dolných končatín človeka pri chôdzi z nasnímaných polôh kĺbov pomocou video kamery. S použitím výsledkov tímového projektu [1] a [2] z roku 2003/2004 a 2004/5 použite spôsob reprezentácie kostry končatín a ich animácie. Sústreďte sa hlavne na snímanie pohybu pomocou viac kamier, synchronizácie videosekvencií, získania súradníc kĺbov, ich transformovanie do animačného pod systému a 3D animácie.

Na základe analýzy modifikujte a doplňte existujúci systém pre nasledovné činnosti:

- zosnímanie pohybu pomocou video kamery, pričom jednotlivé kĺby sú označené kontrastnými značkami. Overte použitie LED diód a filtrov z predchádzajúceho projektu, experimentujte a vylepšite kontrast kombináciou rôznych farieb LED a filtrov
- zlepšite digitalizáciu video sekvencií, rozloženie na jednotlivé snímky, predspracovanie a segmentácia obrazu, určenie súradníc kĺbov v 3D (použijú sa dve kamery alebo obraz z dvoch priemetov) z dvoch na seba kolmých kamier
- odvodenie vzťahov pre výpočet súradníc kĺbov pre prípad, ak pohľady kamier nie sú na seba kolmé pomocou epipolárnej geometrie
- transformácia súradníc kĺbov do formátu animačného systému
- vytvorenie niekoľkých animácií z nameraných dát (s použitím pod systému vytvoreného v predchádzajúcom projekte)
- analýza animácie z nameraných výsledkov
- zhrnutie dosiahnutých výsledkov vo forme článku na medzinárodnú vedeckú konferenciu (v angličtine)

Odporúčaná literatúra:

1. Marešák, J., Matuška, M., Petreje, J., Sás, I.: Animácia a vizuálna analýza chôdze človeka. Tímový projekt FIIT STU, Bratislava 2004.
2. Lučenič, Ľ. A kol., Animácia a vizuálna analýza chôdze človeka. Tímový projekt FIIT STU, Bratislava 2005.
3. K. Tsutsuguchi , Y. Suenaga , Y. Watanabe , K. Shimohara: Human Walking Animation System in Three-dimensional Modeled Scene
<http://www.ipsj.or.jp/members/Journal/Eng/3804/article013.html>
4. Bipedal Human Gait : A 3d Interactive Model with Animation:
http://www.frontiernet.net/~imaging/gait_model.html

5. Amr A. Awadallah: A Hierarchical Kinematic-Dynamic Approach for the Graphical Animation of Human Figure Walking", Master's Thesis
<http://klamath.stanford.edu/~aaa/human.html>
6. A Survey of Computer Vision-Based Human Motion Capture: Journal: Computer Vision and Image Understanding: CVIU,
<http://citeseer.nj.nec.com/moeslund01survey.htm>
7. K.R. Kaufman: Future Directions in Gait Analysis.
<http://www.vard.org/mono/gait/kaufman.htm>

Obsah

0	Úvod	1
0.1	Použitá literatúra	1
1	Špecifikácia požiadaviek	3
1.1	Proces vytvorenia animácie	3
1.1.1	Zosnímanie chôdze viacerými kamerami	4
1.1.2	Synchronizácia kamier	4
1.1.3	Digitalizácia videozáznamu	4
1.1.4	Zistenie polohy kĺbov	4
1.1.5	Vypočítanie 3D súradníc polohy kĺbov	5
1.1.6	Transformácia 3D súradníc do formátu súboru animačného programu	5
1.1.7	Animácia pohybu spodnej časti ľudskej kostry	5
2	Analýza	6
2.1	Analýza existujúcich riešení	6
2.1.1	APAS/Gait	6
2.1.2	The motion monitor	8
2.1.3	Kinemetrix motion analysis system	9
2.1.4	Záver analýzy existujúcich riešení	11
2.2	Analýza súčasného stavu projektu	12
2.2.1	Analýza vytvorených programov	12
2.2.2	Záver analýzy súčasného stavu projektu	16
2.3	Značková analýza (Marker Analysis)	16
2.3.1	Aktívne a pasívne značky	16
2.3.2	Umiestnenie značiek na povrchu tela	17
2.3.3	Aktívne značky s použitím LED	17
2.3.4	Pasívne značky v oblasti infračerveného svetla	18
2.4	Epipolárna geometria	19
2.4.1	Geometrický model	21
2.4.2	Základné vzťahy v epipolárnej geometrii	22
2.4.3	Určovanie fundamentálnej matice	23
2.5	Úprava scény	24
2.5.1	Rozmiestnenie kamier	25
2.5.2	Kontrast medzi objektami a pozadím	25

2.5.3	Osvetlenie scény	26
2.6	Predspracovanie obrazu	28
2.6.1	Ekvalizácia (vyrovnanie) histogramu.....	29
2.6.2	Filtrácia šumu (lokálne predspracovanie)	29
2.6.3	Obyčajné spriemerovanie	30
2.6.4	Ďalšie metódy filtrácie	30
2.7	Segmentácia.....	30
2.7.1	Histogram	31
2.7.2	Prahovanie	32
2.7.3	Globálne prahovanie	33
2.7.4	Poloprahovanie	33
2.7.5	Spektrálne prahovanie	33
2.7.6	Multispektrálne prahovanie	33
2.7.7	Určenie prahovej hodnoty T	34
2.7.8	Metódy detekcie hrán	34
2.7.9	Metódy segmentácie založené na oblastiach	35
2.8	Záver analýzy	35
3	Hrubý návrh riešenia	36
3.1	Proces analýzy chôdze	36
3.2	Systém videokamier.....	36
3.3	Synchronizácia kamier	38
3.4	Spracovanie videa.....	38
3.4.1	Prevod na snímky.....	39
3.5	Animácia chôdze	40
3.6	Analýza chôdze.....	41
4	Prototyp	42
4.1	Tvorba histogramu	42
4.2	Určenie prahovej hodnoty	43
4.3	Detekcia značiek.....	43
4.4	Určenie stredu značiek	44
4.5	Zápis do súboru	44
4.6	VideoSQC	44
4.7	Výsledky segmentácie.....	45
4.8	Záver zimného semestra.....	46

5	Algoritmy a experimenty	47
5.1	Lokálny histogram	47
5.2	Experimenty zamerané na zlepšenie segmentácie	49
5.3	Epipolárna geometria	53
5.4	Segmentácia pomocou programu VideoSQC.....	60
5.5	Integrácia BlueBone s VideoSQC	61
5.6	Výsledky	66
6	Vysokorýchlostné kamery	68
7	Záver.....	70

0 Úvod

Tímový projekt sa zameriava na oblasť animácie pohybu ľudského tela. V informačných technológiách táto oblasť nie je veľmi rozšírená, aj keď existujú komerčné systémy z danej oblasti, ktoré sú ale nedostupné a veľmi drahé. Náš tímový projekt sa zaoberá konkrétne analýzou chôdze človeka a jej vizualizáciou.

Táto oblasť, a konkrétne aj tento projekt zahŕňa mnoho rôznych procesov, ktoré treba riešiť. V prvom rade je to získanie obrazových údajov, z ktorých sa vykonáva analýza, tiež problém ich digitalizácie a rozlíšenia, ako získať požadované údaje, ktoré nám umožnia zobrazit' dynamický pohyb ľudskej nohy v špeciálnom na to určenom softvéri. Taktiež konštrukcia tohto softvéru nieje triviálna a vyžaduje veľa úsilia od riešiteľov projektu. Projektov z danej oblasti nieje veľa, preto sa snažíme vymyslieť čo najlepšie a z hľadiska celkových nákladov aj prijateľné riešenie, v danej oblasti neexistujú štandardy, z ktorých by bolo na prvý pohľad jasné, ako sa bude riešiť tento projekt, preto veľká časť úloh bude výskumného charakteru, a hlavnou činnosťou bude vymyslieť čo najlepšie riešenie.

Tento projekt je riešený pre výskumné účely v danej oblasti analýzy a vizualizácie pohybu ľudskej nohy, projekt sa rieši pre lekára, ktorý je odborníkom v danej oblasti.

Projekt bol riešený viac či menej úspešne už dvoma predchádzajúcimi tímami, tímom „Stopári“ a tímom „Five fit feet of FIIT“. Našou úlohou v projekte je dotiahnuť ho do podoby, aby bola celá súprava softvéru a zariadení použiteľná v praxi, tj. v lekárskej praxi doktora, a mohla sa využívať na analýzu porúch chôdze.

0.1 Použitá literatúra

[1] <http://www.sportsci.com/adi2001/adi/services/support/manuals/gait/markersets.asp>

[2] <http://www.innsport.com/>

[3] <http://www.mie-uk.com/kin/>

[4]

<http://www.laboratorium.dist.unige.it/~piero/Teaching/Gait/BONTRAGER%20Instrumented%20Gait%20Analysis%20Systems.htm>

- [5] Borghese N. A., Cerveri P. - Calibrating a video camera pair with a rigid. 1999
http://homes.dsi.unimi.it/~borghese/References/Papers/00_SelfCal_PattRecogn.pdf
- [6] Dařílková K. – Modelovanie 3D objektov z fotografie. 2004
<http://fractal.dam.fmph.uniba.sk/~darilkova/RigoroznaPracaFINE.pdf>
- [7] Zhang Z. - Determining the epipolar geometry and its uncertainty: A review. 1996
<http://graphics.cs.msu.su/library/publications/determ.pdf>
- [8] <http://neuron.tuke.sk/~pluchta/Pocitacove%20Videnie/Prednasky/minimovka.doc>
- [9] <http://fractal.dam.fmph.uniba.sk/~darilkova/RigoroznaPracaFINE.pdf>

1 Špecifikácia požiadaviek

Analýza chôdze človeka je sofistikovaná laboratórna technika, ktorá používa modernú elektroniku na analýzu dynamiky pohybu človeka. Slúži pre lekárov, chirurgov a terapeutov pre určenie kvantitatívnych a dynamických hraníc pohybu ľudského tela a jeho končatín. Tiež pomáha lekárom rozhodnúť, čo spôsobuje poruchy pohybu končatín a aký je dôsledok pri kompenzovaní týchto porúch.

Pre chirurgický tím je analýza chôdze človeka užitočná pri rozhodnutiach o aplikácii príslušnej chirurgickej procedúry na nápravu nesprávnej chôdze. Pre chirurgiu je tiež užitočná z hľadiska učenia sa, či bola táto porucha použitím danej procedúry napravená.

Použitím 3D systémov pre analýzu pohybu je možné dosiahnuť [1]:

- Presné meranie odchýlok od normálu
- Zistiť príčiny poruchy chôdze
- Poskytnúť informácie pre lekárov pri robení rozhodnutí o chirurgickom zákroku
- Ohodnotiť dôsledok zákroku na maximalizovanie výsledku
- Porovnať chôdzu pacienta pred a po vykonaní zákroku

Analýza chôdze kombinuje odborné znalosti z ortopédie, biomechaniky a „fyzickej terapie“.

Oblasť nášho projektu sa zameriava na zosnímanie ľudskej chôdze a následné animovanie tohoto pohybu na zmenšenom modeli spodnej časti ľudskej kostry. Projekt má využitie v oblasti medicíny, pri zisťovaní a analýze porúch chôdze.

Celý proces od zosnímania chôdze až po jej animáciu pozostáva z nasledujúcich krokov:

- Zosnímanie chôdze viacerými kamerami
- Synchronizácia kamier
- Digitalizácie videozáznamu
- Zistenie polohy kĺbov
- Vypočítanie 3D súradníc polohy kĺbov
- Transformácia 3D súradníc do formátu súboru animačného programu
- Animácia pohybu spodnej časti ľudskej kostry

1.1 Proces vytvorenia animácie

V tejto časti popíšeme podrobne jednotlivé časti procesu vytvorenia animácie podľa krokov uvedených v úvode kapitoly.

1.1.1 Zosnímanie chôdze viacerými kamerami

Videozáznam možno vo všeobecnosti spraviť dvoma typmi kamier, a to digitálnou a analógovou. Záznam spravený analógovou kamerou musíme previesť do digitálneho záznamu A/D prevodníkom, ktorý môže byť umiestnený napr. na televíznej karte v počítači. Spôsoby záznamov a ich prevod analyzovali predchádzajúce tími spolu s mnohými formátmi záznamov, preto sa tejto časti nebudeme venovať.

1.1.2 Synchronizácia kamier

Je dôležitá lebo kamery sa nedajú spustiť v rovnakom čase a vznikne medzi nimi časová odchylka, ktorá spôsobí desynchronizáciu. Z tohto dôvodu je potrebné navrhnúť mechanizmus, ktorý kamery zosynchronizuje a umožní tak správne spracovať výstupné dáta. Predchádzajúce tími vyriešili tento problém bleskom v momente, kedy všetky kamery bežali. Snímky zachytené pred bleskom sa zmažú a zvyšné snímky sa použijú pri ďalšom výpočte.

1.1.3 Digitalizácia videozáznamu

Pod digitalizáciou sa rozumie prevod do takého formátu, s ktorým už v počítači možno ďalej pracovať. Výber vhodného formátu je veľmi dôležitý kvôli schopnosti korektne rozoznať polohu kĺbov na jednotlivých snímkoch. S formátom videozáznamu súvisí aj chybovosť rozoznávania snímkov a v neposlednom rade veľkosť, ktorú zaberá videozáznam na disku počítača.

Po zdigitalizovaní videa ho musíme rozdeliť na jednotlivé snímky, z ktorých sa budú určovať súradnice značiek. Rozdelenie na snímky sa zabezpečí funkciami v programovacom jazyku, konkrétne z knižnice MFC. Tiež treba vhodne zvoliť formát snímkov, aby práca s nimi a ich predspracovanie bolo čo najefektívnejšie a vykonávalo sa čo najrýchlejšie.

1.1.4 Zistenie polohy kĺbov

Toto je najdôležitejšia úloha. Rozpoznávanie polohy kĺbov zo snímkov videozáznamu musí byť dostatočne spoľahlivé a efektívne. Predchádzajúce tímy označovali kĺby značkami. Vyskúšali rôzne typy a farby značiek, avšak s malým úspechom. My chceme vyskúšať nový typ, a to reflexné značky. Úlohou bude vytvoriť algoritmus s čo najväčšou úspešnosťou

rozoznávanie značiek. Metóda na zisťovanie polohy značiek zo snímok sa nazýva segmentácia. Bude podrobne popísaná v analýze.

1.1.5 Vypočítanie 3D súradníc polohy kĺbov

3D polohu kĺbov bude treba vypočítať z polohy jednotlivých značiek zachytených viacerými kamerami v tom istom čase. Využije sa tu epipolárna geometria, ktorá nám umožní umiestniť kamery v ľubovoľnom uhle, čo výrazne uľahčí adaptovanie sa na rôzne prostredia.

1.1.6 Transformácia 3D súradníc do formátu súboru animačného programu

Po vypočítaní 3D súradníc polôh kĺbov je potrebné previesť ich do takého formátu, aby ich dokázal animačný program zobrazit'. Na animáciu použijeme program BlueBone, ktorý vytvorili predchádzajúce tímy.

1.1.7 Animácia pohybu spodnej časti ľudskej kostry

Toto je finálna časť projektu. Na animáciu použijeme už vytvorený program BlueBone. Vstupné dáta animácie si načíta zo vstupného súboru, z ktorých vytvorí animáciu. Na animáciu sa použije spodná časť ľudskej kostry.

2 Analýza

V tejto časti zanalyzuje súčasný stav projektu a možné prístupy, ktoré sa využívajú vo svete. Toto zadanie má hlavne výskumný charakter, problematika segmentácie obrazu je pomerne komplikovaná. Od času riešenia posledného tímu ubehla pomerne dlhá doba, vo svete riešenie pokročilo, využívajú sa nové prístupy. V tejto časti sa zameriame aj na analýzu najzaujímavejších riešení alebo prístupov. Problematikám synchronizácie kamier a extrakcie snímok z video záznamu sa nebudeme venovať, lebo už boli dostatočne analyzované predchádzajúcimi tímami a už sú vyriešené.

2.1 Analýza existujúcich riešení

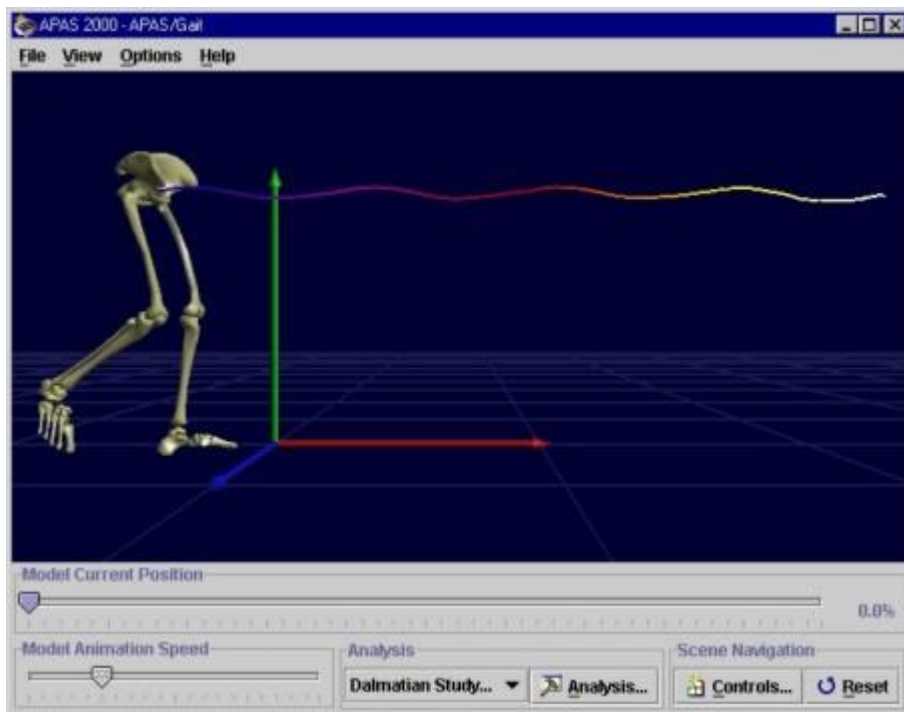
Pri analýze sme sa zamerali aj na pozorovanie a analýzu existujúcich softvérových systémov pre snímanie dynamického pohybu nohy. Predchádzajúce tímy sa tiež venovali tejto problematike. V priebehu dvoch rokov sa v tejto oblasti ale mnoho zmenilo, resp. vylepšilo a existujúci softvér bol vylepšený o nové verzie. V tejto kapitole popíšeme niektoré významné existujúce systémy z oblasti analýzy pohybu ľudského tela, nielen nohy. Popíšeme aj spôsoby, akými zisťovali polohy jednotlivých značiek, ako aj spôsoby animácie, samozrejme podľa dostupných informácií.

2.1.1 APAS/Gait

Tento softvér umožňuje 3D animáciu spodnej časti ľudského tela. Na animáciu použili rôzne množiny značiek [1]. Tieto množiny hovoria o umiestnení značiek na ľudskom tele.

V scéne je možné sa pohybovať použitím myši. Pohybom myši a držaním príslušného tlačidla je možná navigácia v scéne. Tlačidlá myši majú nasledovné efekty:

- Ľavé – otáčanie scény
- Stredné (alebo Alt + Ľavé) – zoom
- Pravé – prekladanie scény

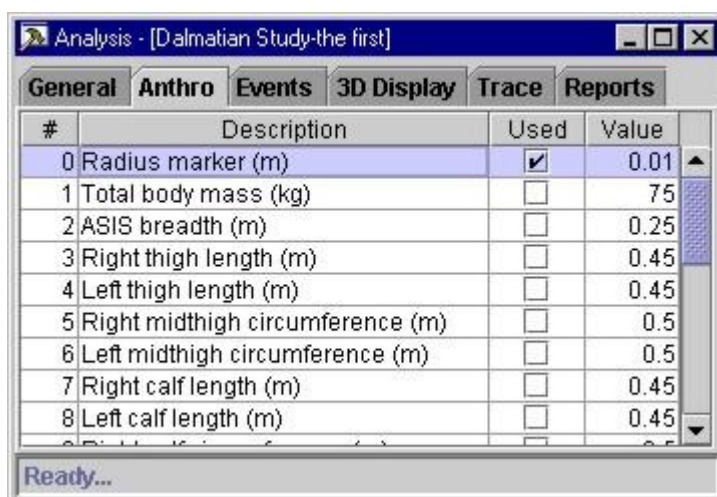


Obr. č. 1 Animácia v APAS/Gait

V programe je možné nastavovať rôzne antropomorfné vzdialenosti a sledovať tak vplyv ich zmeny na výsledok animácie a meraní. Medzi takéto antropomorfné vzdialenosti patrí napríklad polomer značky, dĺžka pravého stehna, dĺžka ľavého stehna a iné.

Okrem toho je možné nastavovať mnoho ďalších vecí, ako je napríklad tzv. „trace mód“.

V tomto móde je možné vybrať konkrétne kĺby (konkrétny členok, koleno, panva alebo bedro) a sledovať tak pohyb tejto časti. Umožňuje tiež vkladať rôzne reporty – komentáre, dátumy a iné.



Obr. č. 2 Nastavenie značiek v APAS/Gait

APAS/Gait používa reflexné značky a svetelné zdroje na zväčšenie kontrastu medzi značkami a pozadím. Svetelné zdroje používajú pre každú kameru zvlášť a smerujú tým istým smerom, ako je uhol pohľadu kamery. Intenzita svetla nemusí byť vysoká. Čím je väčšia táto intenzita, tým má väčší zásah do protiľahlej kamery.

Predchádzajúci tím tiež analyzoval tento systém, avšak asi nemali dostatočne naštudovanú ich web stránku. Niektoré časti ich stránky boli totiž naposledy modifikované v roku 2001, niektoré v roku 2002 ale i v r. 2004. Podľa toho boli prístupné i skôr v takej kvalite, ako je v súčasnosti. Sú tu k dispozícii podrobné tutoriály a manuály k systému. Systém je tiež dostupný k stiahnutiu – 30 dňová trial verzia.

Podrobné hardvérové nároky na systém sú uvedené na adrese

<http://www.sportsci.com/adi2001/adi/services/support/tutorials/gait/chapter3/3.2.asp>.

Požadované hardvérové nároky sú nasledovné:

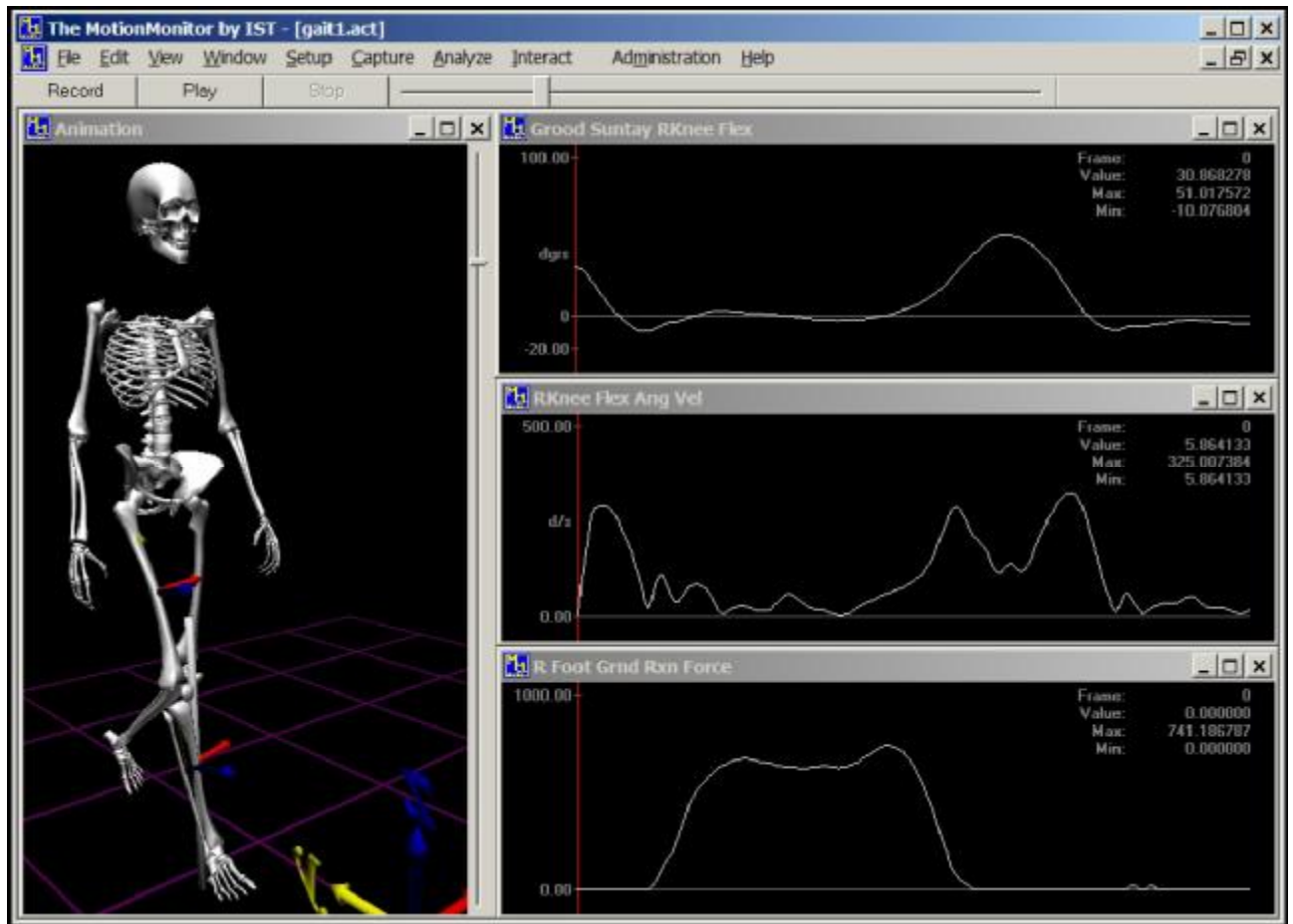
- Jedna video kamera pre 2D analýzu
- Videorekordér (video tape recorder - VCR) pre každú kameru pre 3D analýzu – používajú analógové kamery
- Video kamery s 50 snímkov za sekundu (norma PAL) alebo 60 snímkov za sekundu s použitím genlocku – postačujúce pre chôdzu, avšak použitie viacerých kamier spôsobí dobu expozície väčšiu o 8.3 ms alebo 10 ms (polovica snímku) pre každú kameru.
- „shutter“ (uzávierka) – je použitý pre každú kameru a definuje dobu expozície každého snímku. Bežné hodnoty sú 1/125, 1/250, 1/500, 1/1000 a môže byť aj 1/10000 sekundy.
- Konvertor analógového videozáznamu na digitálny
- Synchronizácia udalosťou – použitá je LED dióda, ktorá je viditeľná z každej kamery.
- Foto bunky na meranie rýchlosti pohybu.
- Svetlá pre každú kameru.
- Kalibračná kocka – pre 3D analýzu (pre 2D stačia štyri kalibračné body)

Nastavenie uzávierky na 1/500 redukuje pohyb značky umiestnenej na členku na 0,78 cm, čo je pomerne uspokojivé.

2.1.2 The motion monitor

Tento softvér vyrobila firma Innovative Sports Training, je určený pre laboratóriá a aplikáciu v medicínskej oblasti. [2] Slúži na presné meranie pohybu ľudského tela, v rôznych aplikáciách, ako autori píšú. Pomocou tohoto softvéru je možno určovať aj silu pôsobiacu nohou, pravdepodobne po zadaní dodatočných parametrov tela osoby. Softvér je komerčného využitia, a je volne neprístupný, preto sme ho teda nemali možnosť odskúšať. Aspoň

uvádzame na obrázku č. 3 vzhľad ich softvéru. Veľkou výhodou softvéru, ako autori tvrdia, je práca real-time bez post-processingu pri analýze pohybu.

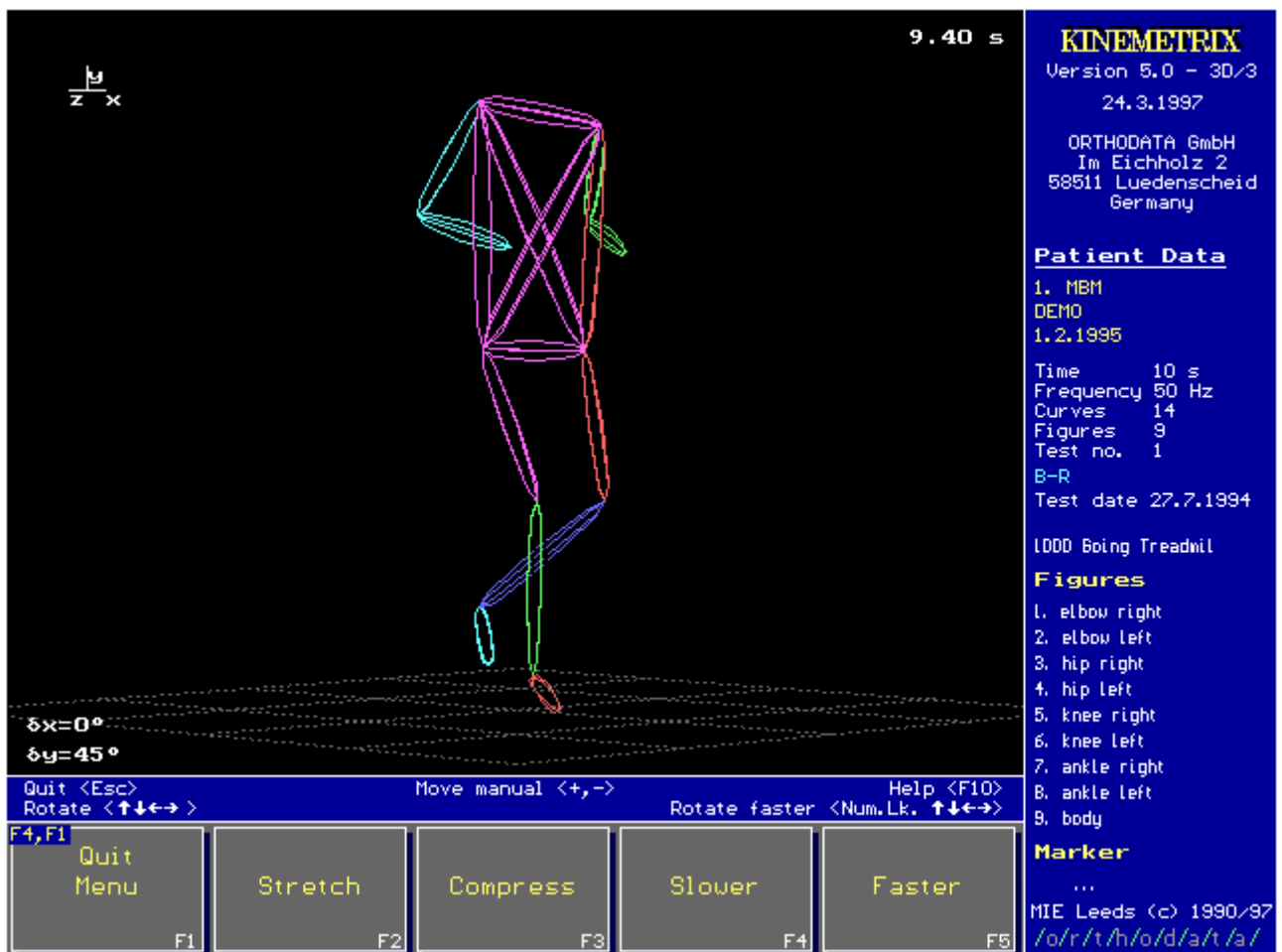


Obr. č. 3 Prostredie softvéru The motion monitor

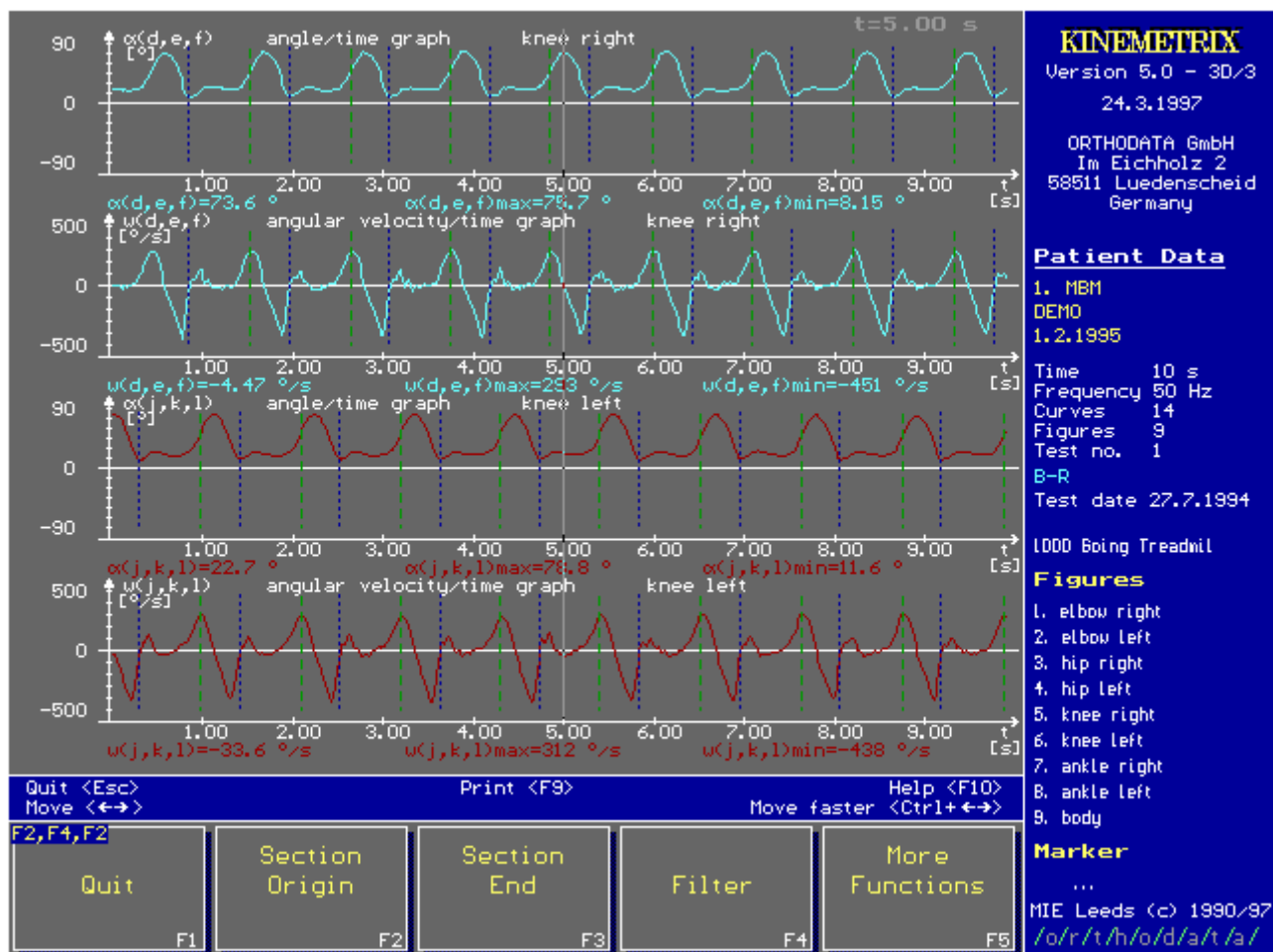
2.1.3 Kinemetrix motion analysis system

Kinemetrix je balíček softvéru, ktorý umožňuje pomocou infračervenej technológie analyzovať pohyby ľudského tela pomocou reflexných značiek.[3] Pre spracovanie obrazu sa využíva špeciálna výkonná karta, ktorá spracováva vstupné údaje. Systém používa na analýzu jednoduché reflexné značky, ktoré sa umiestnia na telo človeka. Tento systém vytvára celkovú analýzu pohybu človeka, a teda sa nezameriava len na spodnú časť tela. Autori na stránkach <http://www.mie-uk.com/kin/> uvádzajú rôzne použitia systému, v oblastiach ako pediatria, šport, ergonomika, medicína a priemysel. Softvér umožňuje nastaviť viacero

konfigurácií značiek, maximálne až 100 značiek. Z analyzovaných dát generuje rôzne grafy, umožňuje prezentovať v grafoch aj jednotlivé značky. Systém umožňuje 2D analýzu pomocou jednej kamery, ako aj analýzu 3D pomocou dvoch a viacerých kamier. Pravdepodobne je možné generovať vzťahy medzi značkami manuálne aj počas behu programu, a program potom podľa týchto informácií vytvorí 3D graf. Pohyb tela je možno ľubovoľne otáčať v softvéri. V strohej technickej špecifikácii uvedenej na webe je uvedené, že systém používa kamery CCIR štandardu, čo je 625 riadkov na 576 stĺpcov pixelov. Je možné použiť kamery s 50 alebo 100 snímkami za sekundu. Veľmi zaujímavé sú hardvérové požiadavky na systém. Uvádza sa použitie počítača s procesorom 486DX, 8 MB pamäte a 80 MB Diskovým priestorom. Teda tento systém je implementovaný pod operačným systémom MS-DOS, a vyžaduje verziu MS-DOS 6.0 alebo vyššiu. Práve preto sme názoru, že už rozlišovanie bodov na tele človeka a prevod na 3D súradnice sa vykonáva práve na prídavnej karte do slotu počítača, čo výrazne uľahčuje činnosť vlastného počítača, ale táto informácia na webe nieje uvedená. Ukážka softvéru je na obr.2 a 3.



Obr. č. 4 Obrazovka Kinematrix motion analysis system



Obr. č. 5 Výstupy vo forme grafov v Kinematrix motion analysis system

2.1.4 Záver analýzy existujúcich riešení

Vo svete a v niektorých tu uvedených systémoch analyzujú viacej informácií, nielen poruchy chôdze. Niektoré tu uvedené systémy robia viacej typov analýz[4]:

- Časová analýza chôdze (Temporal gait analysis). Pri tomto spôsobe analýzy sa vyhodnocuje rýchlosť a spôsob chôdze človeka, dĺžka kroku a pravidelnosť. Pre analýzu sa používajú tzv. spínače nôh (Footswitches) a špeciálne koberčeky na chôdzu (Gait mats).
- Analýza tlaku (Foot pressure analysis) Tu sa analyzuje tlak ktorým pôsobí končatina na podložku, a rozloženie tohoto tlaku na povrchu končatiny. Používajú sa tlakové koberce (Pressure mats), veľkosti niekoľko metrov štvorcových, ktoré obsahujú senzory. Hustota senzorov je od 1 až po asi 3 senzory na centimeter štvorcový,

z čoho vyplýva aj rozlišovacia schopnosť koberca. Tiež sa môžu použiť tlakové podrážky (Pressure insoles), z ktorých pri chôdzi idú dáta priamo do softvéru.

- Analýza pohybu (Motion analysis) Pri analýze pohybu existuje viacero systémov a spôsobov analýzy, najdôležitejšie sú však použitie electrogoniometrov a video analýza pohybu, ktorú prevádzame aj v našom projekte. Pri použití videoanalýzy existujú všeobecne dva spôsoby, a to značková analýza, a bezznačková analýza (Marker a Markerless analysis). Systémy so značkami rozdeľuje na dva druhy, a to
 - Aktívne (Active marker systems). Tu sa ako značky používajú aktívne prvky, napr. IR diódy, ktoré emitujú svetlo.
 - Pasívne (Passive marker systems). Ako značky sa používajú rôzne pasívne materiály rôznych tvarov, najčastejšie reflexné materiály. Používajú sa dve alebo tri kamery, podľa toho, či chceme 2D alebo 3D analýzu, v 3D analýze v praxi ale treba viac než 3 kamery, z dôvodu, že subjekt môže nohu otáčať, a tak nie sú viditeľné všetky značky, a tiež si môže pomáhať, napr. Podopierať palicou. 2D analýzu jednou kamerou ale stránka neodporúča, pretože môže byť nepresná a mala by sa používať len vo vyhradených situáciách.
- Analýza síl (Force analysis) Používa sa na zisťovanie a analýzu síl pôsobiacich pri pohybe človeka. Používajú sa tzv. silové dosky (Force plates), ktoré sa musia špeciálne namontovať do podlahy. Tiež existujú topánky pre meranie síl (Force measurement sandals).
- Elektromyografia (Electromyography, EMG). EMG je spôsob ako zistiť čo svaly robia pri pohybe človeka. Používa sa hlavne v oblasti medicíny na určovanie a analýzu porúch chôdze,

2.2 Analýza súčasného stavu projektu

2.2.1 Analýza vytvorených programov

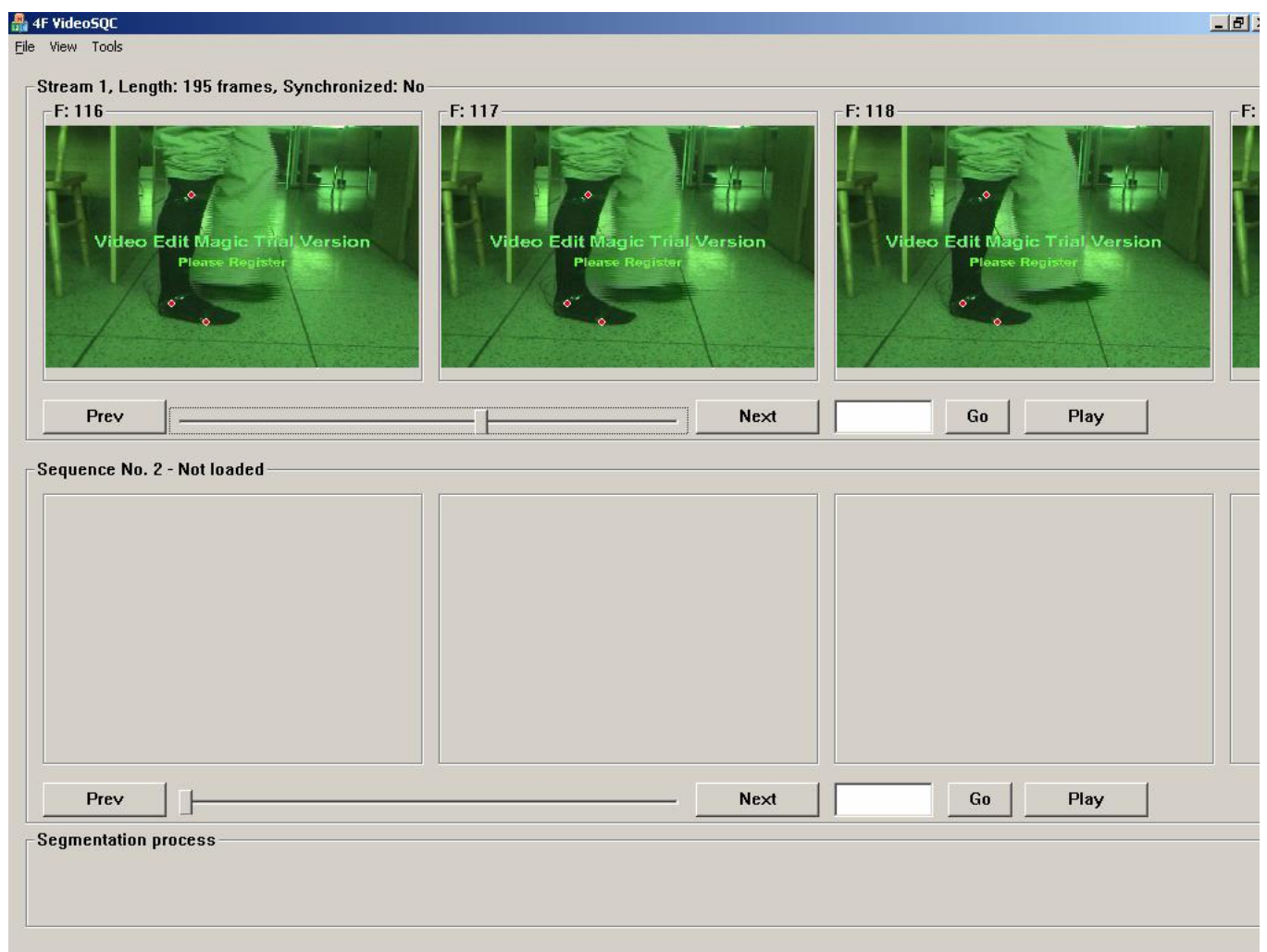
Predchádzajúce dva tímy vytvorili tri programy, ktoré v tejto kapitole zanalyzujeme.

2.2.1.1 VideoSQC

Program slúži na segmentáciu videozáznamu, konkrétne na rozpoznanie značiek a zistenie ich súradníc z jednotlivých snímkov video záznamu. Z jednotlivých snímkov zisťuje len 2D súradnice jednotlivých značiek, čo je nedostatočné. Pre vytvorenie reálnej animácie pohybu ľudskej kostry(jej spodnej časti) v priestore, bude treba doplniť výpočet 3D súradníc zo snímkov z viacerých kamier. Na samotnú segmentáciu je použitý veľmi jednoduchý algoritmus, ktorý nemá dostatočnú funkčnosť a presnosť.

Zdrojové kódy tohto programu sú pomerne čitateľné a dá sa v nich dobre orientovať. Predchádzajúci tím venoval najväčšie úsilie na synchronizáciu videozáznamov, ktorá bola rozpracovaná na dobrej úrovni, takže tejto časti nebude nutné venovať toľko pozornosti.

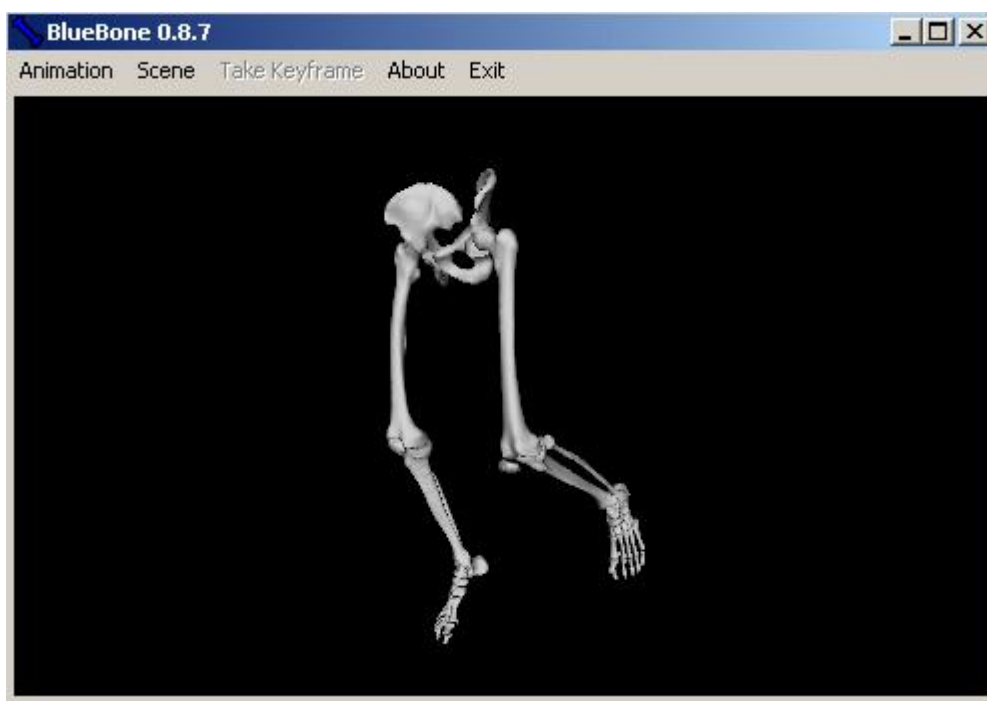
Napriek závažným nedostatkom, program necháme a upravíme ho od základov. Zdokonalíme algoritmus segmentácie, rozšírime jeho funkcie o predspracovanie obrazu a výpočet 3D súradníc. Z programu využijeme získavanie snímkov z video záznamu a synchronizáciu kamier. Tento program považujeme za kľúčový, preto mu budeme venovať najviac pozornosti. Výstupné dáta tohto programu budú slúžiť na animáciu kostry.



Obr. č. 6 Aplikácia VideoSQC

2.2.1.2 BlueBone2

Program zobrazuje zjednodušený model spodnej časti ľudskej kosti. Program umožňuje meniť polohu končatín, otáčať, rotovať, približovať a oddiaľovať pohľad na model. Umožňuje vytvoriť animáciu kostry priamo v programe, alebo načítať zo vstupného súboru. Program dokáže animovať kostru len raz, bude treba doprogramovať možnosť animovania kostry v opakujúcich sa cykloch. Program má drobné chyby, ktoré bude treba opraviť, ale inak je funkčný a vyhovuje potrebám nášho tímu.



Obr. č. 7: Aplikácia BlueBone

2.2.1.3 Analyser

Aplikácia analyser, ktorá bola vytvorená predchádzajúcimi tímami, umožňuje zobrazovať grafy pohybu ľudskej nohy. Aplikácia je prehľadne rozdelená na dve hlavné časti, a to oblasť grafov, ktorá sa nachádza na ľavej strane aplikácie, a oblasť ovládania na pravej strane.

Umožňuje zobrazenie nasledovných grafov zvlášť pre ľavú a pravú nohu:

- Bedrový kĺb
- Koleno
- Členok
- Päta
- Prsty
- Uhl'ová rýchlosť koleno-členok-prsty
- Uhl'ová rýchlosť koleno-členok-päta

Aplikácia zobrazuje dva typy grafov, a to zobrazenie y-ovej súradnice pohybu bodov na nohe v závislosti od času, a zobrazenie z-ovej osi. Aplikácia na hornom grafe v popise grafu v ľavo hore ukazuje, že zobrazované informácie sú z x-ovej osi, ale v dokumentácii k aplikácii je všade spomínaná y-ová os, čo je nepochopiteľný rozdiel, aplikácia asi v skutočnosti zobrazuje hodnoty z y-ovej osi.

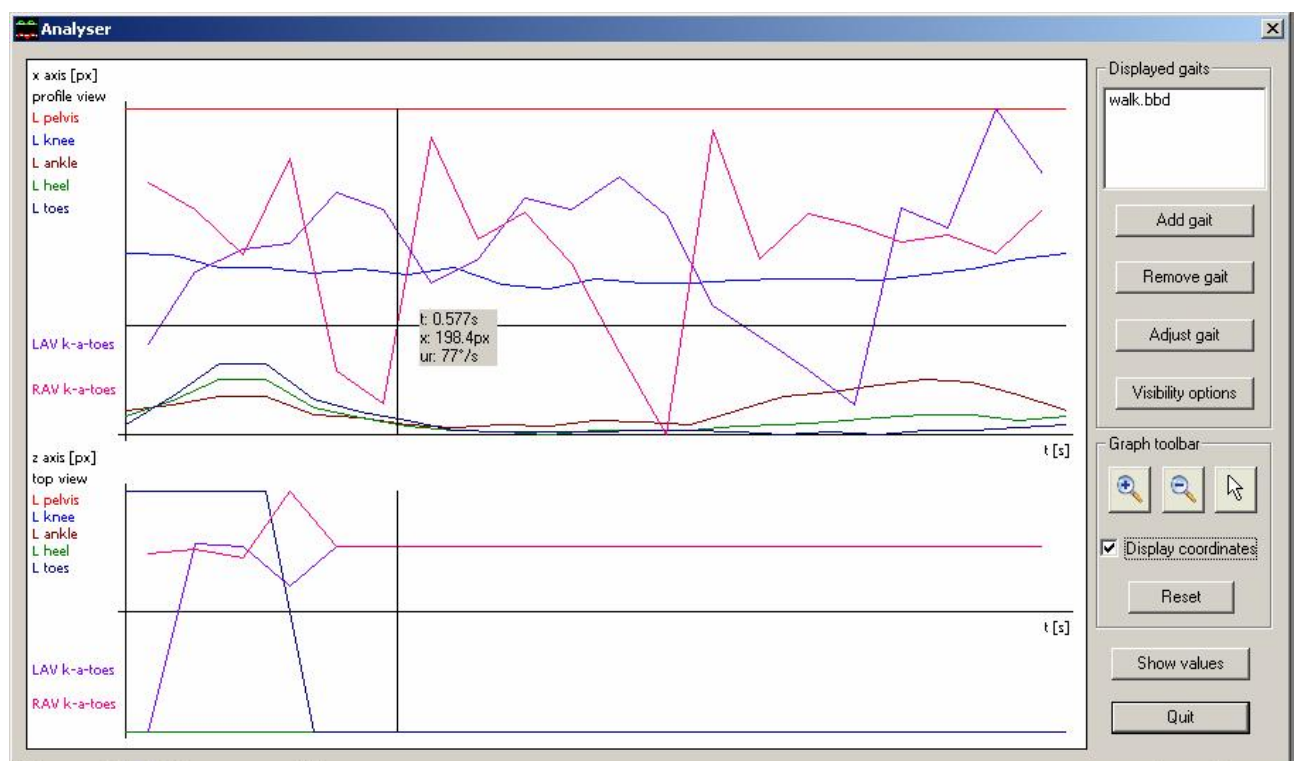
Veľmi vhodná a praktická je možnosť zobrazenia súradníc na aktuálnej polohe kurzora myši, keďže na osiach nie sú zobrazené hodnoty súradníc.

Aplikácia umožňuje zobraziť viacero analýz chôdzí v jednom grafe rozlíšením, pomocou typu čiary.

Grafy aplikácie sa dajú zväčšovať, znižovať a posúvať podľa potreby.

Jazyková verzia aplikácie je mätúca a nie veľmi dobre spracovaná, keďže základné zobrazenie aplikácie je celé v anglickom jazyku, ale nastavovacie modálne okná, ako aj okno tabuľky hodnôt, zobrazenie viditeľnosti grafov, a rôzne výstražné okná sú všetky v slovenskom jazyku, čo je trochu nekonzistentné. Aplikácia má nevhodné nastavenie kláves, lebo pri stlačení Esc alebo Enter sa automaticky zavrie, čo je trochu neštandardne správanie.

Ale celkovo je aplikácia dobre spracovaná a zobrazuje dobré a užitočné informácie pri analýze pohybu nohy.



Obr. č. 8 Aplikácia Analyser

2.2.2 Záver analýzy súčasného stavu projektu

Predchádzajúci tím realizoval zopár experimentov, z ktorých budeme vychádzať. Účelom je dosiahnuť animáciu pohybu oboch dolných končatín, preto dve kamery nebudú postačujúce. Pri bočnom pohľade by totiž nebolo možné vidieť značky na druhej nohe. Bude potrebné upraviť programy tak, aby dokázali spracovať vstupy z viacerých videokamier a dosiahnuť tak animáciu spodnej časti kostry.

2.3 Značková analýza (Marker Analysis)

2.3.1 Aktívne a pasívne značky

Z pohľadu videoanalýzy rozlišujeme dva základné typy značiek:

- aktívne značky
- pasívne značky

Aktívne značky sú spravidla zdrojom elektromagnetického žiarenia, ktoré je schopné rozoznávať záznamová videokamera. Najčastejšie sú používané tzv. vysokosvietiace LED diódy (v oblasti viditeľného svetla) alebo infračervené LED diódy.

K aktívnym značkám sa radia aj napr. ultrazvukové značky, avšak v ich prípade sa nepoužíva videoanalýza. K vyhodnocovaniu informácií získaných za pomoci ultrazvukových značiek je potrebné iné zariadenie ako videokamera.

Pasívne značky svojimi vlastnosťami umožňujú odrážanie svetla a tým zvýšenie intenzity osvetlenia v oblasti, kde je značka umiestnená (reflexné značky), prípadne zabezpečia ostrý farebný kontrast značky s jej okolím (kontrastné značky). V praxi sa môže jednať o útvary jasnejších farieb, napr. ploché telesá pokryté retroreflexnou páskou. Postprocesingom obrazu získaného z kamier (napr. prahovanie) sa značky zvýrazia a zamedzí sa zmäteniu segmentačného algoritmu.

Špeciálne značky sa používajú v oblasti infračerveného svetla. V tomto prípade sa obraz sníma infračervenými kamerami s prípadným použitím infračerveného filtra. Tento prístup

výrazne zjednodušuje algoritmus segmentácie obrazu. Pokým pri obyčajnom svetle vo viditeľnej oblasti vzniká v rámci nasnímanej videosekvencie množstvo farebných odtieňov rôznej svetelnej intenzity, čo zvyšuje obtiažnosť lokalizácie značky, pri infračervenom svetle rámec zachytáva v podstate len značky, od ktorých sa infračervené lúče odrazili vo výraznejšej miere v porovnaní s okolitým materiálom.

2.3.2 Umiestnenie značiek na povrchu tela

Pri rozmiestnení značiek treba mať na zreteli, že z hľadiska kvality trojdimenzionálneho modelovania scény je potrebné, aby každá značka bola snímaná v každom okamihu aspoň dvoma kamerami. Táto požiadavka sa dá zabezpečiť zvýšením počtu kamier, ktoré snímajú sledovaný subjekt.

V prípade, že značky sú umiestnené priamo na povrchu tela subjektu, ich viditeľnosť v priestore je nižšia ako u značiek, ktoré sú umiestnené mimo povrchu tela (napr. sférické značky umiestnené na podporných stĺpkoch), a teda sú viditeľné pod väčším priestorovým uhlom. V druhom prípade sa zvýši pravdepodobnosť, že v danom okamihu je značka snímaná videokamerou. Tento postup je vhodný najmä v prípade, že laboratórium disponuje menším množstvom snímacích videokamier.

2.3.3 Aktívne značky s použitím LED

Systémy so značkami LED zapínajú značky sekvenčne na relatívne krátku dobu, takže sú automaticky schopné identifikovať a rozpoznať konkrétnu značku na základe načasovania svetelného impulzu. Sledovanie značiek nie je v tomto prístupe žiadnym problémom, najmä v prípade dočasnej straty značky z dohľadu alebo vzájomného skríženia trajektórie dvoch značiek. Keďže splývanie dvoch značiek nie je v tomto prípade riskantné, značky je možné umiestniť v relatívne tesnej vzájomnej blízkosti. Tento fakt je nespornou výhodou predovšetkým v prípade snímania pohybu kostí chodidla nohy, a aj konkrétnych prstov, ktoré sú umiestnené v tesnej blízkosti jeden od druhého. Je celkom zrejmé, že použitie pasívnych značiek by bolo v tomto prípade veľkým problémom. Použitie sekvenčného systému LED diód ilustruje obr. č. 9.



Obr. č. 9 Použitie infračervených LED diód pulzujúcich sekvenčne v stanovenom poradí a časovaní

Okrem nespornej výhody v možnosti umiestnenia značiek v tesnej blízkosti nevýhodou tohto systému je nutnosť umiestnenia dodatočného príslušenstva na pacienta. V prípade dlhšie trvajúceho nepretržitého testovania by nevýhodou mohlo byť aj zahrievanie batérií a LED diód.

Pri použití infračervených LED diód sa pred konvenčnými videokamerami v profesionálnych laboratóriách uprednostňujú špeciálne kamery s poľami fotocitlivých diód umiestnenými za tieniacou maskou s pseudonáhodne rozloženým čiarovým kódom. Keď na subjekte blikne LED dióda, tieň masky sa premietne na pole senzorov špeciálnej kamery. Pozícia tieňa je v geometrickom vzťahu k pozícii značky. Priemerujúci efekt príspevku signálu všetkých fotocitlivých prvkov zlepšuje rozlišovaciu schopnosť systému a poskytuje dobrý pomer signál-šum.

2.3.4 Pasívne značky v oblasti infračerveného svetla

Nespornou výhodou pasívnych značiek je ich nízka hmotnosť bez potreby upínania dodatočných batérií a káblov na telo pacienta. Po tele pacienta sú rozmiestnené reflexné značky, ktoré majú vyššiu odrazivosť infračerveného svetla ako ich okolie. Infračervené LED diódy rozmiestnené okolo šošovky kamery vysielaajú pulzy infračerveného žiarenia, ktoré sú značkami odrazené naspäť k šošovke kamery. Na šošovke je použitý jeden alebo viacero infračervených filtrov a prah systému je nastavený tak, aby zvýraznil jasnejšie značky a potlačil menej výrazné objekty v okolí.

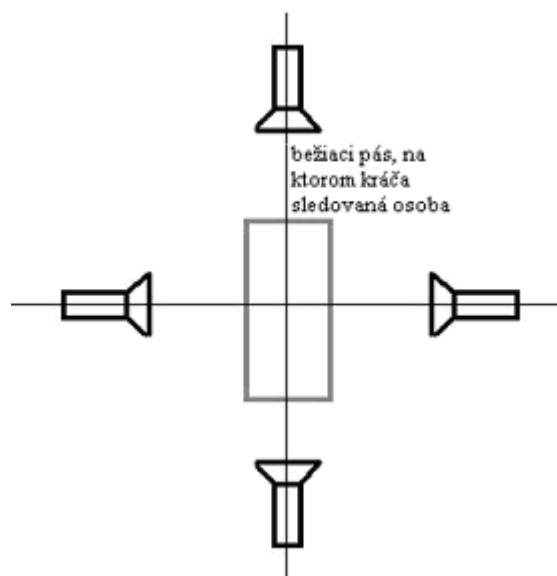
Pasívna povaha značiek vyžaduje, aby každá značka bola identifikovaná štítkom a stopovaná počas experimentu. Ak sa značka stratí z dohľadu kamery alebo dôjde k jej prekrytiu s inou

značkou, môže stratiť svoju správnu identifikáciu v systéme. Nebezpečenstvo prekrytia pasívnych značiek v tomto prípade určuje limity pre minimálnu vzdialenosť značiek.

2.4 Epipolárna geometria

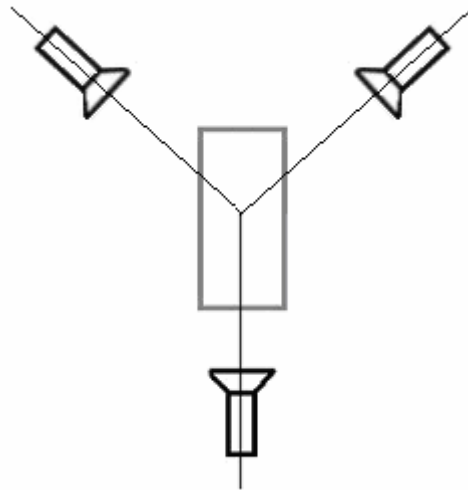
Pri nahrávaní chôdze človeka sa použijú kamery, ktoré budú zaznamenávať pacientov pohyb pri pohľade spredu, zozadu a zboku. Zo záberov kamier budeme v neskorších krokoch rekonštruovať pozíciu jednotlivých značiek vzhľadom na nami zvolenú súradnicovú sústavu. V projektoch predchádzajúcich tímov sa uvažovalo s umiestnením kamier kolmo na seba (tzn. osi kamier zvierajú pravý uhol). To je však obmedzujúce, pretože vychýlenie osi niektorej z kamier vnesie do procesu určovania súradníc určitú chybu. Obmedzenie dané takýmto riešením je možné odstrániť práve použitím epipolárnej geometrie.[5][6][7] Tá nám umožní relatívne presne vypočítať požadované súradnice bez ohľadu na umiestnenie kamier v priestore, takže nie je nutné zaoberať sa presným nastavovaním pozícií kamier pred meraním.

Ďalšou výhodou použitia epipolárnej geometrie je možnosť použitia menšieho počtu kamier pre dosiahnutie rovnakých výsledkov. Aby sme dokázali zaznamenať chôdzu obidvoch nôh, bolo by nutné použiť minimálne 4 kamery (znázornené na obr. č. 10), aby sa zachytili všetky značky rozmiestnené na končatinách.



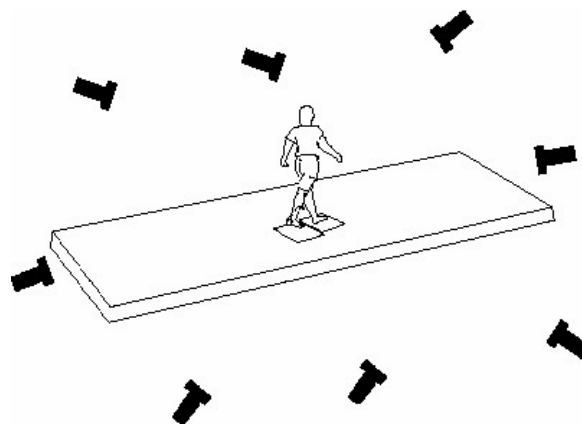
Obr. č. 10 Snímanie chôdze 4 kamerami

Pri efektívnom rozmiestnení kamier okolo pozorovaného objektu nám však na určenie 3D súradníc značiek postačia iba 3 kamery (obr. č. 11), pretože bočné kamery zachytia aj značky umiestnené na prednej strane nôh.



Obr. č. 11 Snímanie chôdze 3 kamerami

Epipolárna geometria predstavuje vo všeobecnosti nástroj, ktorý umožňuje narábať so scénou zachytenou ľubovoľným počtom kamier, pričom ich rozmiestnenie nie je nijako obmedzené (obr. č. 12).



Obr. č. 12 Scéna s 8 kamerami

Hlavným využitím epipolárnej geometrie je identifikácia korešpondujúcich si bodov zachytených na snímkach a rekonštrukcia ich 3D súradníc. Aby sme však vedeli určiť 3D súradnice zachytených bodov, je nutné vedieť správne nastaviť hodnoty interných a externých parametrov popisujúcich danú scénu.

Vnútorne parametre: ohnisková vzdialenosť kamier, pozícia riadiacich bodov

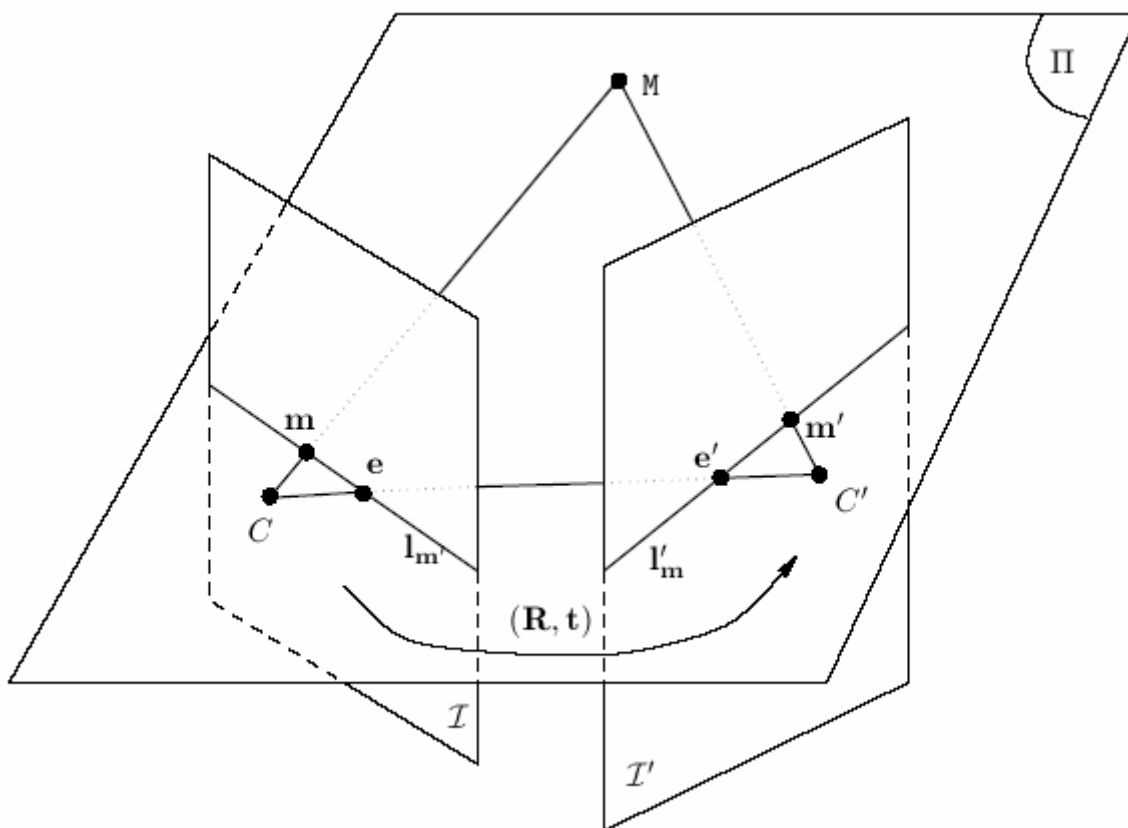
Vonkajšie parametre: posunutie a natočenie snímok voči sebe

Pokiaľ predpokladáme, že parametre sa počas nahrávania chôdze nemenia, nie je nutné použiť na ich určenie špeciálne kalibračné prístroje – dajú sa priamo vypočítať zo snímok scény (tzv. samokalibrácia).

2.4.1 Geometrický model

V tejto časti je uvedená jednoduchá schéma, pomocou ktorej sú vysvetlené základné pojmy z oblasti epipolárnej geometrie. Premenné vzťahujúce sa k druhej kamere sú označené apostrofom.

Na obrázku č. 13 je zobrazený geometrický model scény pozostávajúci z dvoch kamier. Bod $M[x \ y \ z]^T$ predstavuje nejaký bod v priestore. Je zachytený pred obidvomi kamerami a zobrazený do bodov $m[u \ v]^T$ a m' umiestnených v rovinách zobrazenia kamier. Rovina zobrazenia je vlastne obraz scény tak, ako je zachytený videokamerou.



Obr. č. 13 Scéna zložená z dvoch kamier

Body C a C' sú stredy premietania prvej a druhej kamery. Ich vzdialenosť od roviny zobrazenia sa potom nazýva ohnisková vzdialenosť f . Bod m na prvej rovine zobrazenia môže ležať kdekoľvek na priamke l'_m pri pohľade z druhej kamery a naopak. Tieto priamky určujú epipolárnu rovinu Π , ktorá je vlastne množinou všetkých potenciálnych pozícií bodu M v priestore. Prieniky spojnice ohnísk C a C' s rovinami zobrazenia sa nazývajú epipóly e a e' . Označujú miesta, kde na snímke vidíme druhú kameru.

2.4.2 Základné vzťahy v epipolárnej geometrii

Pre vzájomný vzťah medzi skutočným bodom M v priestore a jeho obrazom m na snímke platí vzťah:

$s \cdot m = P \cdot M$, a po dosadení

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

kde s je ľubovoľná mierka a P je matica 3×4 , nazývaná matica projekcie. V prípade, že súradnicová sústava má začiatok v C a jej osi sú rovnobežné s osami roviny zobrazenia, môže byť P rozložená do tvaru $P = K \cdot [I, 0]$, kde K je kalibračná matica 3×3 mapuje súradnice prostredia do súradníc kamery, I je štvorcová identická matica 3×3 je posunutie a 0 je nulový stĺpec.

$$K = \begin{bmatrix} a_x & c & t_x \\ 0 & a_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

a – škálovanie v smeroch osí x, y , súvisiace so zobrazovacím systémom

t – súradnice optického centra – prienik zobrazovacej roviny s optickou osou

c – parameter skreslenia (obvykle 0)

Matica projekcie K sa vo všeobecnosti líši od $K \cdot [I, 0]$ orientáciou a posunutím voči začiatku súradnicovej sústavy. Pridáva sa preto matica transformácie súradnicovej sústavy kamery do sústavy prostredia:

$$P = K[I,0] * \begin{bmatrix} R & -RC' \\ 0 & 1 \end{bmatrix}$$

kde R je matica 3×3 popisujúca rotáciu a C' je vektor posunutia vzhľadom na súradnicovú sústavu prostredia.

Aby sme dokázali popísať vzťah medzi dvoma obrazmi m , m' toho bodu M , potrebujeme určiť maticu F , pre ktorú platí:

$$m'^T \cdot F \cdot m = 0$$

Matica F má rozmery 3×3 a nazýva sa fundamentálna matica. Je algebraickou reprezentáciou projektívnej geometrie medzi dvoma obrazmi – epipolárnej geometrie. Jej hodnoty závisia iba na vzájomnej pozícii kamier a ich kalibrácii (tieto hodnoty sú v maticiach P , P').

2.4.3 Určovanie fundamentálnej matice

Problém hľadania matice projekcie je totožný s problémom hľadania fundamentálnej matice. Na riešenie tohoto problému existuje niekoľko metód:

1. Lineárne algoritmy
2. Algoritmus algebraickej minimalizácie
3. Minimalizácia vzdialenosti

2.4.3.1 Základný lineárny osem-bodový algoritmus

Pre pár korešpondujúcich si bodov $m[u \ v \ 1]^T$ a $m'[u' \ v' \ 1]^T$ platí:

$$m'^T \cdot F \cdot m = 0$$

Teda po rozpísaní:

$$u'uf_{11} + u'vf_{12} + u'f_{13} + v'uf_{21} + v'vf_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

Hodnoty fundamentálnej matice sa dajú usporiadať do vektora:

$$(u'u, u'v, u', v'u, v'v, v', u, v, 1).f = 0.$$

Pre viac korešpondujúcich si bodov sa vytvorí matica A pozostávajúca z takýchto vektorov. Riešenie je možné určiť presne, až na škálovanie. V prípade, že má matica A hodnotu väčšiu ako 8, jej riešenie získame metódou najmenších štvorcov. V takom prípade je f singulárny vektor matice A združený s jej najmenším singulárnym číslom.

2.4.3.2 Algoritmus algebraickej minimalizácie

Dôležitou vlastnosťou fundamentálnych matíc ich singularita. Jednou z možností je konštruovať ju ako singulárnu, ako súčin

$$F = O[e]_x$$

Kde O je regulárna matica, $[e]_x$ je nejaká koso-symetrická matica a e aproximuje epipól v prvom obraze. Aby sa splnili aj ďalšie podmienky fundamentálnej matice, potrebujeme aby sa minimalizovala hodnota matice A pri podmienke, že hodnota $f = I$.

2.4.3.3 Minimalizácia vzdialenosti

Táto metóda je založená na nelineárnej minimalizácii vzdialeností. Lineárnou metódou získame odhad matice F_L . Zvolíme vektor parametrov v , tak aby spĺňal požiadavky kladené na fundamentálnu maticu. Vypočítame parametre v tak, aby minimalizovali $F_L - F(v)$.

S využitím $F(v)$ ako počiatočného kroku minimalizujeme chybovú funkciu vektora v :

$$\min \sum_{i=1}^n d^2(F(v), m_i, m'_i)$$

Kde d je vzdialenosť v metrickom priestore a m, m' je n korešpondujúcich bodov.

2.5 Úprava scény

Predchádzajúce dva tímy analyzovali možnosti úpravy scény pred jej samotným snímaním.

Táto problematika je z ich strany popísaná na dobrej úrovni. V analýze uviedli zásadné vlastnosti, ktoré treba zabezpečiť úpravou scény. Tieto vlastnosti môžeme zhrnúť nasledovne:

1. Dobré osvetlenie scény
2. Minimálny počet snímaných objektov
3. Jednoduché pozadie scény
4. Pozadie scény musí byť v kontraste so snímanými objektami.
5. Vhodné rozmiestnenie kamier

Podľa ich dosiahnutých výsledkov a priložených videozáznamov usudzujeme, že niektoré body neboli z ich strany vôbec splnené. Veľmi dôležité sú hlavne body 1 a 4. Niektoré existujúce systémy (napr. APAS/Gait), ktoré sme analyzovali v predchádzajúcich kapitolách, sa zamerali hlavne na tieto body a dosiahli tak vynikajúce výsledky. Použitím reflexných značiek a vhodným osvetlením scény zvýšili kontrast medzi značkami a okolím. Predchádzajúce tímy síce uviedli použitie reflexných značiek v analýze, avšak pri experimentovaní ich nepoužili.

V nasledujúcich podkapitolách uvedieme možnosti, ako dosiahnuť uvedené vlastnosti.

2.5.1 Rozmiestnenie kamier

Bez použitia epipolárnej geometrie je najvhodnejšie rozmiestnenie kamier pod 90 stupňovým uhlom. Pri použití dvoch kamier na získavanie 3D súradníc je najvhodnejšie rozmiestniť ich tak, že jedna kamera bude snímať sledovaný objekt (človeka) zozadu a jedna kamera z boku (v skutočnosti sa nebude sledovať objekt, ale značky, ktoré sú na ňom umiestnené – viď nasledujúca podkapitola).

V prípade použitia ďalších kamier – jedna spredu a jedna z druhého boku. Použitie viac ako štyroch kamier však nepredpokladáme.

2.5.2 Kontrast medzi objektami a pozadím

Pre zabezpečenie dostatočného kontrastu medzi sledovaným objektom a jeho okolím sa používajú rôzne kontrastné značky. Predchádzajúce tímy používali jednoduché značky malých rozmerov a pomerne nevýrazných farieb. Boli použité jednoduché značky z obyčajných farieb a LED diódy, ktoré svietili. Tiež používali zelený filter pred kamerou,

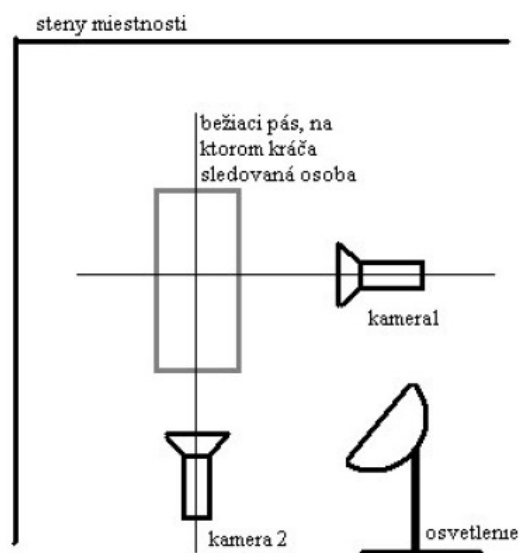
aby tak zvýraznili len svietiace body (LED diódy). Filter však spôsobil, že nebol dosiahnutý dostatočný kontrast medzi značkami a okolím.

Výhodnejšie sa zdá byť použitie kontrastných značiek. Ich použitím dosiahli ľudia, ktorí pracovali na systéme APAS/Gait, vynikajúce výsledky v dosiahnutí takéhoto kontrastu. Pri použití reflexných značiek treba použiť aj vhodné osvetlenie scény. To je podrobne popísané v nasledujúcej podkapitole.

2.5.3 Osvetlenie scény

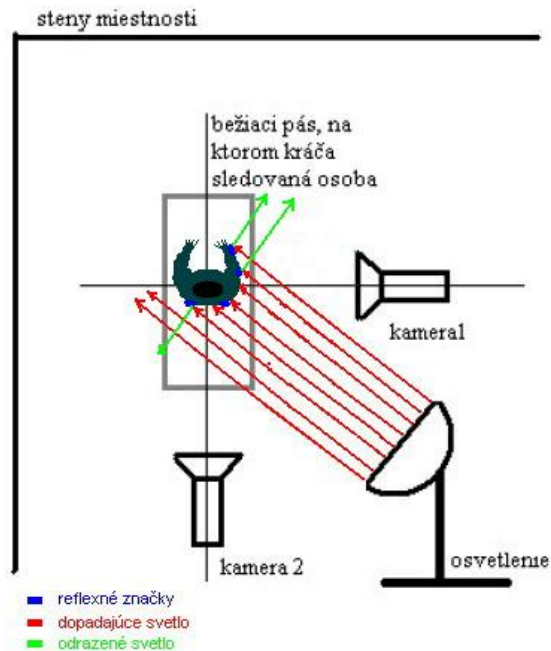
Existuje veľa možností, ako osvetliť scénu. Osvetlenie, ktoré sa bežne nachádza takmer v každej miestnosti nie je postačujúce pre účely tohto projektu. Najčastejšie sa nachádza na strope miestnosti a je všesmerové. Osvetlenie scény pre účely segmentácie je závislé aj na rozmiestnení kamier a na type použitých značiek.

Predchádzajúci tím chcel použiť pri experimentovaní dve kamery a zdroj svetla chceli umiestniť medzi tieto kamery. Jedna kamera snímala objekt zo zadu a jedna z boku. Zvierali teda 90 stupňový uhol a osvetlenie bolo umiestnené medzi týmito kamerami (s každou kamerou zvieralo 45 stupňový uhol). Toto rozmiestnenie je znázornené na nasledujúcom obrázku.



Obr. č. 14 Umiestnenie objektov

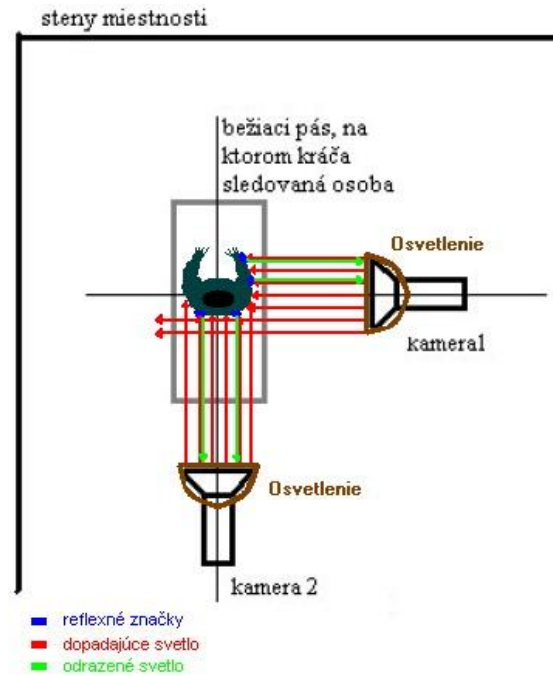
Ak predpokladáme použitie reflexných značiek pri použití takéhoto osvetlenia (umiestnené pod 45 stupňovým uhlom a s rovnomernými lúčmi), vyskytne sa nasledujúci problém, ktorý je znázornený na dolu uvedenom obrázku.



Obr. č. 15 Osvetlenie scény navrhované predchádzajúcimi tímami

Ako je vidno z obrázku, značky nebudú mať dostatočný kontrast s okolím, pretože odrazené svetlo z týchto značiek nesmeruje priamo do objektívu kamery.

Vhodnejšie je teda použiť nasledovné rozmiestnenie kamier, ktoré je znázornené na dolu uvedenom obrázku.



Obr. č. 16 Nami navrhované osvetlenie scény

Pri takomto rozmiestnení svetiel sa niektoré časti lúčov odrážajú zo značiek priamo do objektívu kamery. Tým sú reflexné značky jasnejšie oproti okoliu. Na zabezpečenie ešte väčšieho kontrastu medzi značkami a okolím je vhodné použiť tmavé oblečenie človeka a tmavú farbu stien. Tmavá farba pohlcuje svetlo a okolie značiek tak ostane tmavé.

2.6 Predspracovanie obrazu

Cieľom predspracovania obrazu je zvýrazniť určité črty obrazu, ktoré sú dôležité z hľadiska ďalšieho spracovania.[8] Predspracovanie obrazu tvorí často neoddeliteľnú súčasť spracovania obrazovej informácie. Zlepšuje kvalitu obrazu a tým prispieva k lepším výsledkom spracovania.

Metódy predspracovania obrazu sú založené na nadbytočnosti údajov v obraze. Typickým prípadom je skreslenie obrazovej informácie náhodným šumom. Tento sa dá odstrániť na základe priemeru hodnôt jasu v jeho blízkom okolí, pretože susedné body v obraze majú väčšinou rovnakú alebo podobnú hodnotu jasu.

2.6.1 Ekvalizácia (vyrovnanie) histogramu

Táto metóda slúži napríklad na hľadanie transformačného vzťahu pre modifikáciu jasovej stupnice. Príkladom takejto modifikácie je zvýšenie kontrastu pôvodne nekонтрастného obrazu.

Vo výslednom obraze po ekvalizácii histogramu sú jednotlivé jasové úrovne zastúpené približne v rovnakom množstve. Ekvalizácia zvýši kontrast pre úrovne jasů blízko maximím histogramu a zníži kontrast blízko minimím histogramu.

Vzťah pre ekvalizáciu je nasledovný:

$$q = T(p) = \frac{q_k - q_0}{N^2} \sum_{i=p_0}^p H(i) + q_0$$

pričom $\langle p_0, p_k \rangle$ - interval jasov vo vstupnom obraze

$H(p)$ - histogram vstupného obrazu

N - počet riadkov a stĺpcov

$\langle q_0, q_k \rangle$ - interval jasov vo výstupnom obraze

$q = T(p)$ - je výsledná jasová transformácia

2.6.2 Filtrácia šumu (lokálne predspracovanie)

Lokálne predspracovanie znamená výpočet jasů bodu vo výstupnom obraze na základe lokálneho okolia zodpovedajúceho bodu vo vstupnom obraze.

Rozdelenie lokálneho predspracovania:

- Vyhladzovanie obrazu – slúži na potlačenie vyšších frekvencií obrazovej funkcie. Tým sa chce dosiahnuť potlačenie náhodného šumu. Súčasne však dochádza k potlačeniu iných náhlych zmien jasů, napríklad ostrých hrán a čiar.
- Gradientové operácie – slúžia na zdôraznenie vyšších frekvencií – zvýraznenie hrán. Súčasne sú však zvýraznené aj šumové body.

Filtrácia šumu predstavuje potlačenie rozdielu jasů vo vnútri oblasti, ktorá je zašumená. V prípade, že veľkosť objektov je dostatočne veľká, šum je možné odstrániť spriemerovaním jasových hodnôt v istom okolí.

2.6.3 Obyčajné spriemerovanie

Obyčajné spriemerovanie filtruje obraz takým spôsobom, že ako nový jas bodu sa zvolí aritmetický priemer jasu bodov jeho okolia. Jeho nevýhodou je rozmazávanie hrán, ktoré sa eliminuje ďalšími filtračnými metódami. Filtrácia touto metódou je špeciálnym prípadom diskretnej konvolúcie. Pre okolie rozmeru 3x3 je konvolučná maska pre filtráciu obyčajným spriemerovaním:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Často sa používa väčšia váha stredného bodu masky alebo jeho štyroch susedov:

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Tým sa lepšie aproximujú vlastnosti šumu s Gaussovským rozdelením.

2.6.4 Ďalšie metódy filtrácie

Často sa používajú aj iné metódy filtrácie, napríklad:

- Filtrácia metódou mediánu
- Metóda rotujúcej masky

Týmto sa podrobnejšie nebudeme venovať.

2.7 Segmentácia

Ako bolo uvedené vyššie, segmentácia slúži na rozdelenie obrazu na objekty, resp. na rozoznanie objektov v obrázku. Neexistuje žiadna univerzálna metóda segmentácie, ktorá by bola vhodná na ľubovoľné typy obrazov. Preto je dôležité vyskúšať širokú skupinu techník

a postupov. Napodobniť rozpoznávanie, ktoré vykonáva ľudské oko a mozog, je veľmi zložitá úloha.[9]

Rozdelení metód segmentácie je pomerne veľa. Niektoré sú uvedené v literatúre.

Všeobecne môžeme metódy segmentácie rozdeliť na:

1. metódy založené na obrazových bodoch
2. metódy založené na oblastiach

Medzi metódy založené na obrazových bodoch patrí napríklad prahovanie alebo techniky založené na detekcii hrán.

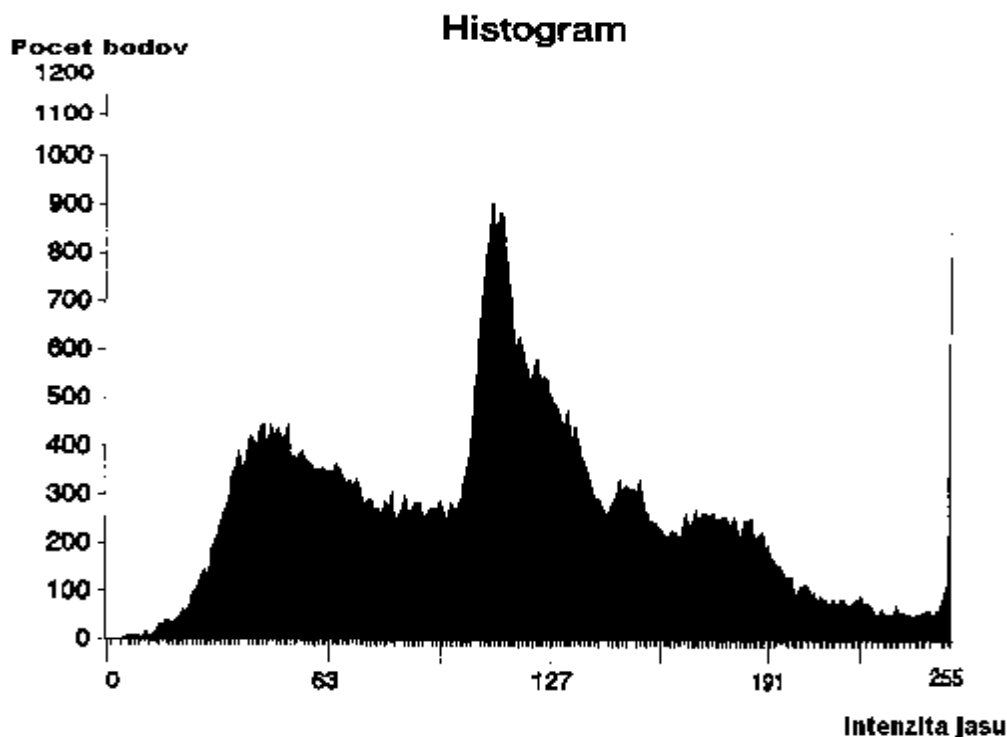
Základným princípom metód založených na oblastiach je hľadanie homogénnych oblastí v obraze.

2.7.1 Histogram

Histogram obrazu slúži na reprezentáciu zastúpenia jednotlivých jasových úrovní v digitálnom obraze. Je veľmi často používaný práve pri segmentácii obrazu (napríklad pre metódy prahovania – vid' nasledovná podkapitola). Histogram jasu je vektor H s počtom zložiek rovným počtu jasových úrovní. Hodnota každej zložky odpovedá početnosti bodov príslušného jasu v obraze.

Histogram digitalizovaného obrazu má niekoľko lokálnych miním a maxím. To môže spôsobiť komplikáciu pri jeho ďalšom spracovaní. Preto sa často histogram jasu vyhladzuje, aby sa potlačili jeho lokálne poruchy – vid' kapitola Predspracovanie obrazu.

Na nasledujúcom obrázku je znázornená ukážka takéhoto histogramu.



Obr. č. 17 Histogram

2.7.2 Prahovanie

Táto metóda používa na rozoznanie objektov v obraze jasovú hodnotu jednotlivých obrazových bodov. Vychádza z predpokladu, že objekty sú charakterizované konštantnou odrazivosťou, či pohltivosťou svetla na svojom povrchu. Ak obraz obsahuje dostatočne kontrastné objekty vzhľadom na pozadie, dá sa použiť istá prahová hodnota k ich vzájomnému oddeleniu.

Prahovanie je najpoužívanejšou a najrýchlejšou segmentačnou metódou.

Metódy prahovania môžeme rozdeliť na:

- Jednoduché prahovanie
 - globálne prahovanie
 - poloprahovanie
 - spektrálne prahovanie
 - multispektrálne prahovanie
- Adaptívne prahovanie

2.7.3 Globálne prahovanie

Princíp tejto metódy spočíva v určení prahovej hodnoty. Na základe tejto hodnoty sa obrazové body rozdeľujú na body objektu a na body pozadia.

Výsledný obraz $g(x, y)$ takto spracovaného pôvodného obrazu $f(x, y)$ získame nasledovne:

$$\begin{aligned} g(x, y) &= 255 & \text{ak } f(x, y) > T \\ g(x, y) &= 0 & \text{ak } f(x, y) \leq T \end{aligned}$$

2.7.4 Poloprahovanie

Poloprahovanie je podobné globálnemu prahovaniu, avšak jasové hodnoty, ktoré presahujú prah T sa nemenia.

$$\begin{aligned} g(x, y) &= f(x, y) & \text{ak } f(x, y) > T \\ g(x, y) &= 0 & \text{ak } f(x, y) \leq T \end{aligned}$$

2.7.5 Spektrálne prahovanie

Pri spektrálnom prahovaní sa namiesto prahovej hodnoty používa interval hodnôt.

$$\begin{aligned} g(x, y) &= f(x, y) & \text{ak } f(x, y) \in I \\ g(x, y) &= 0 & \text{inak} \end{aligned}$$

Kde I je interval jasových hodnôt.

2.7.6 Multispektrálne prahovanie

Len málokedy je možné úspešne prahovať s tým istým prahom na celej ploche obrazu. Je to spôsobené zmenami jasov objektu a pozadia zavinenými napr. nerovnomernosťou osvetlenia či nerovnakými vlastnosťami snímacieho zariadenia na ploche obrazu [1].

Používa sa v prípade, že na obrázku sa nachádza viac objektov s rôznymi jasovými hodnotami. V tom prípade môžeme použiť nasledovný vzorec:

$$g(x, y) = a_1 \quad \text{ak } f(x, y) \in I_1$$

$$g(x, y) = a_2 \quad \text{ak } f(x, y) \in I_2$$

...

$$g(x, y) = a_n \quad \text{ak } f(x, y) \in I_n$$

$$g(x, y) = 0 \quad \text{inak}$$

Kde I_1, I_2, \dots, I_n sú disjunktné intervaly jasových hodnôt

a_1, a_2, \dots, a_n sú rôzne jasové úrovne.

2.7.7 Určenie prahovej hodnoty T

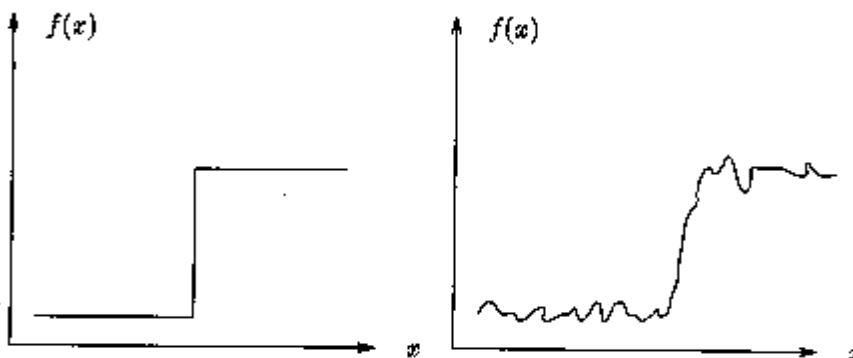
Na vyjadrenie zastúpenia jednotlivých jasových úrovní v obraze sa často používa histogram.

Pre každú jasovú úroveň zastúpenú v obraze udáva celkový počet bodov s týmto jasom [1].

V prípade, že predmety v obrázku majú výrazne odlišnú hodnotu jasu od pozadia (napríklad v prípade použitia reflexných značiek), ako prahovú hodnotu je vhodné použiť lokálne minimum medzi dvoma špičkami histogramu.

2.7.8 Metódy detekcie hrán

Mnoho segmentačných metód je založených na detekcii hrán. Tá je založená na myšlienke hľadania zmeny funkcie jasu $f(x)$ v obraze (Pozri nasledujúci obrázok).



Obr. č. 18 Ideálna hrana (a), reálna hrana (b)

Hrany sa hľadajú pomocou operátorov detekujúcich hrany (napr. prahovanie hodnôt jasu alebo gradientu jasu v obraze) [2]. Hľadajú sa teda hranice medzi oblasťami v obraze.

Problémom hranových segmentačných metód je výskyt hrán v miestach bez prítomnosti skutočnej hranice a súčasne absencia hrán tam, kde sa vyskytujú.

Detekcia hrán v zašumených obrazoch je veľmi problematická. Preto je vhodnejšie použiť iné techniky na segmentáciu – vid' metódy založené na oblastiach.

Obraz získaný hranovým operátorom sa nedá použiť ako segmentovaný obraz, v ktorom sú už rozlíšené jednotlivé objekty. Obraz je ešte nutné použiť ako vstup pre ďalšie spracovanie, ktoré pospája jednotlivé hrany do postupností, reťazcov – tie lepšie popisujú hranice objektov v obraze.

2.7.9 Metódy segmentácie založené na oblastiach

Tieto metódy na rozdiel od hranových operátorov priamo konštruujú oblasti v obraze [2].

Použitie týchto metód segmentácie je vhodné v prípade zašumených obrazov, kde dosahujú lepšie výsledky ako metódy založené na detekcii hrán. Podmienkou pre ich použitie v tomto prípade je homogenita oblastí v obraze.

Používajú sa nasledovné metódy :

- Delenie a spájanie oblastí
- Predelová segmentácia (Watershed segmentation)
- Zhodnosť, porovnávanie, podobnosť (Matching)

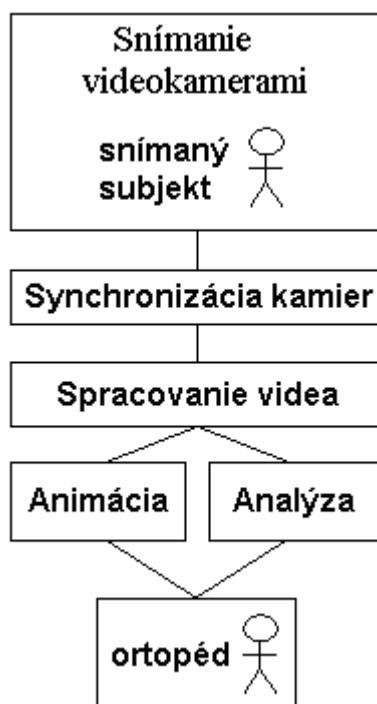
2.8 Záver analýzy

Problémová oblasť je pomerne rozsiahla a existuje viacero možných riešení. So súhlasom vedúceho projektu sme sa rozhodli vytvoriť portál, kde zdokumentujeme a zhrnieme rôzne prístupy a výsledky z tejto oblasti. Výrazne tak uľahčíme hľadanie informácií z tejto oblasti, poprípade aj pre ďalší tím, ktorý sa bude venovať tejto problematike. Portál bude prístupný z web stránky tímu.

3 Hrubý návrh riešenia

Keďže tento projekt má výskumný/experimentálny charakter, nedá sa napísať presný návrh jednotlivých častí systému. Preto tu uvedieme len hrubý návrh systému. Algoritmy pre niektoré časti systému bude treba experimentovaním overiť tak, aby dosahovali čo najlepšiu úspešnosť. Predchádzajúce tímy vyriešili niektoré problémy, ktoré nebude treba znova riešiť, a môžeme sa venovať závažnejším problémom, ktoré vyriešené nie sú.

3.1 Proces analýzy chôdze



Obr. č. 19 Proces analýzy chôdze

3.2 Systém videokamier

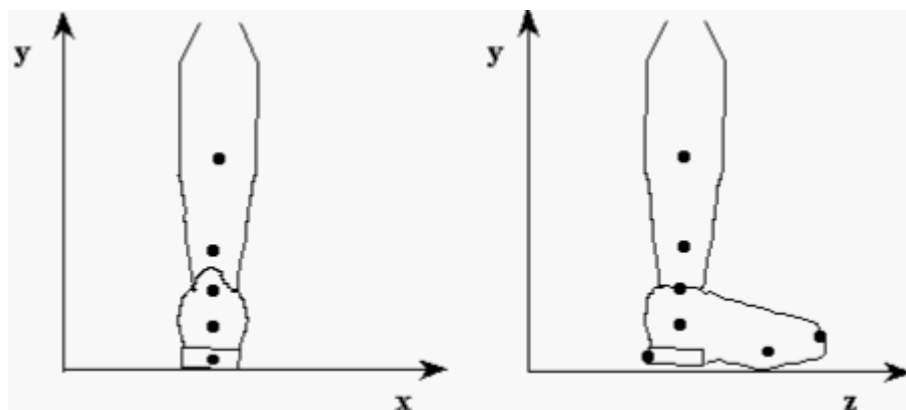
Na nasnímanie chôdze človeka bude potrebné použiť dve alebo viacej kamier. Keďže obraz má byť prezentovaný v 3D súradniciach, musia sa vypočítať zo všetkých kamier. Čím viac kamier sa použije, tým presnejšia bude výsledná animácia, pretože sa dosiahnu presnejšie výpočty. Na začiatku použijeme dve kamery na jednu nohu. Analýza bude prebiehať na oboch nohách, preto bude počet kamier dvojnásobný oproti jednej nohe, teda 4 kamery.

Jednou z požiadaviek je aj variabilný uhol medzi kamerami. V tomto prípade treba využiť epipolárnu geometriu. Uhol medzi kamerami ale určite nemôže byť ľubovoľný, aj vzhľadom na použité osvetlenie. Osvetlenie by nemalo svietiť priamo do kamery.

Chceme využiť reflexné značky, ktoré pri osvetlení sú výrazne viditeľné, a budú sa dať ľahko rozoznať. Preto v scéne musí byť umiestnené dostatočné svetlo, ktoré bude osvetľovať analyzovaný objekt.

Osvetlenie scény musí byť dostatočné, všetky značky musia byť osvetlené dostatočne silne, aby nevznikali chyby pri segmentácii obrazu. Nie je prípustné, aby niektoré značky boli v tieni a neboli osvetlené. Navrhujeme osvetľovať z dvoch zdrojov svetla umiestnených medzi bočnými a zadnými kamerami v strede, alebo budú mať mierne menší uhol k bočnej kamere. Presný vhodný uhol určíme pokusmi, uhol bude výrazne závisieť aj od samotného uhla medzi kamerami.

Umiestnenie značiek bolo riešené predchádzajúcimi tímami spolu s lekárom. Toto umiestnenie predbežne nebudeme meniť, keďže s lekárom sme sa ešte nestretli. Umiestnenie značiek je znázornené na obr. 19.



Obr. č. 20 Umiestnenie značiek

Keďže sa použijú reflexné značky, bude požiadavka na odstránenie všetkých rušivých vplyvov zo scény. Rušivé sú vplyvy iných reflexných predmetov, a predmetov, na ktorých sa odráža svetlo, ako napríklad zrkadlá. Pozadie scény by malo mať čo najmenej rôznych objektov a farieb, malo by byť jednoduché a jednofarebné.

3.3 Synchronizácia kamier

Synchronizáciu kamier riešili minulé tímy. Bolo prezentovaných niekoľko metód synchronizácie, z ktorých ako najvhodnejšie pre naše použitie je metóda blesku. Je veľmi jednoduchá a netreba pri nej špeciálne kamery. Túto metódu preferujeme aj my. Jednoducho sa po spustení všetkých kamier bleskne fotoaparátom, ktorý spôsobí na zázname výrazné zosvetlenie snímok, a programovo potom tieto snímky odstránime. Tiež sme ale rozmýšľali nad synchronizáciou pomocou LED diód alebo rôznych typov bodových svetiel. Tieto diódy alebo svetlá by boli umiestnené v istej časti snímaného poľa kamery a po spustení všetkých kamier sa tieto diódy, prípadne svetlá zapnú. Od tejto chvíle sú kamery synchronizované, a my nemusíme synchronizovať kamery externým bleskom, ktorý zvyšuje náklady na zariadenie. S LED diódami a svetlami budeme experimentovať a skúšať rôzne možnosti v procese implementácie, pretože vyžadujú zmenu kódu.

3.4 Spracovanie videa

Spracovanie videozáznamu je široký pojem, ktorý sa v našom prípade rozdeľuje na viacero podoblastí. Optimálne riešenie každej z nich si bude vyžadovať skúšanie rôznych algoritmov a experimentovanie. Celková schéma spracovania videozáznamu je na obr. XXX.



Obr. č. 21 Spracovanie videozáznamu

3.4.1 Prevod na snímky

Prevod na snímky bol riešený minulým tímom. Je zahrnutý v programe VideoSQC. Túto časť programu môžeme ponechať, je funkčná a nevyžaduje úpravy. Ak bude ale program pre dlhé videosúbory príliš pomalý, môžeme sa pokúsiť túto časť optimalizovať na rýchlosť.

3.4.1.1 Predspracovanie snímkov a detekcia značiek

Minulý tím predspracovanie v programe VideoSQC nepoužil. Priamo detekoval značky pomocou segmentácie obrazu. Keďže ale nepoužívali reflexné značky, ich predspracovanie obsahovalo veľa chýb. Predpokladáme, že pri použití reflexných značiek predspracovanie nebude potrebné, keďže jas osvetlených reflexných značiek je dosť veľký a segmentácia bude bez chýb, resp. s nízkym počtom chýb. Ak by ale segmentácia nebola dostatočne kvalitná, z algoritmov predspracovania chceme použiť algoritmus ekvivalizácie (vyrovnania) histogramu. Tento algoritmus nám zabezpečí zvýšenie kontrastu pre body obrazu, ktoré sú na reflexných značkách. Pre tento algoritmus je potrebné vypracovať histogram každého snímku sekvencie. S použitými algoritmami pre predspracovanie obrazu budeme experimentovať, keďže nevieme, ktorý by sa najlepšie uplatnil v našom projekte.

Pri segmentácii obrazu využijeme algoritmus poloprahovania, ktorý všetky body, ktoré majú väčšiu intenzitu jasu ako prahovú intenzitu, nechá v pôvodnej hodnote intenzity jasu, a ostatným zníži jas na 0. Tiež rozmyšľáme nad použitím globálnej segmentácie, ktorá intenzitu jasu bodov nad prahovou intenzitou jasu zväčší na maximálnu hodnotu 255.

Poloprahovanie chceme využiť, pretože dopredu nepoznáme intenzitu svetla v miestnosti, a svetlá z kamier budú v praxi rôznej intenzity. Po prahovaní musíme zistiť súradnice bodov algoritmom, kedy prechádzame prahovaný snímok a zisťujeme intenzitu bodov snímku. Takto dostaneme body, ktoré patria do značiek, z nich potom musíme určiť stredy značiek. Na určenie konkrétnej súradnice značky musíme určiť stred značky, aby bola detekcia čo najpresnejšia. Na výpočet stredu značky využijeme jednoduchý algoritmus, kedy si určíme pravú, ľavú, hornú a dolnú súradnicu značky a z nich vypočítame stred značky. Tiež môžeme experimentovať aj s týmto algoritmom a využiť pre nájdenie stredu aj viac bodov na značke.

3.4.1.2 Výpočet 3D súradníc

Pokiaľ umožníme ľubovoľné rozmiestnenie kamier v priestore, je nutné do systému zakomponovať algoritmus, ktorý dokáže zo snímok vypočítať pozíciu kamier a určiť všetky potrebné parametre na rekonštrukciu súradníc. Na takýto proces samokalibrácie, t. j. keď budeme mať k dispozícii iba snímky kamier, použijeme základný osem-bodový algoritmus. Tento algoritmus bol spomedzi ostatných zvolený pre jeho rýchlosť a ľahkú implementáciu. Vo väčšine prípadov poskytuje aj pomerne presné výsledky. Ak však bude vyžadovaná väčšia presnosť, môže poslúžiť jeho výstup ako prvý krok v iteračných metódach ako je napr. metóda minimalizácie algebraickej chyby (obidve metódy sú uvedené v analýze).

Algoritmus potrebuje na vykonanie samokalibrácie minimálne 8 bodov viditeľných všetkými kamerami. Za takéto body môžeme považovať reflexné značky umiestnené na nohách. V prípade, že tieto budú na kalibráciu nevhodné, použijú sa vhodne rozmiestnené statické značky s potrebnými vlastnosťami. Snímky potrebné na kalibráciu je možné získať zo záberov, ktoré budú natočené pre účely analýzy a animácie pohybu.

Kalibráciu je nutné vykonať iba na začiatku. Keďže uvažujeme použitie statických kamier, vypočítané parametre sa nebudú v priebehu merania meniť. Vypočítané parametre sa potom môžu používať na transformáciu 2D súradníc značiek (získaných segmentáciou) do 3D súradníc sveta.

3.4.1.3 Transformácia súradníc a zápis do súboru

Transformácia súradníc do formátu súboru analyzátora BlueBone sa bude zabezpečovať v programe VideoSQC po výpočte 3D súradníc bodov animácie. Pri tejto transformácii bude treba korektne previesť 3D súradnice, a bude ich treba prispôbiť do už spraveného formátu súboru.

3.5 Animácia chôdze

Animácia chôdze je riešená pomocou programu BlueBone. Tento program je spravený korektne a poskytuje dobrú funkčnosť. Model kostry bol spravený už prvým tímom, druhý

tím opravil v modeli niektoré chyby. Model je spravený dobre, zlepšiť by sa mohla funkcionálnosť systému a viac možností v menu systému.

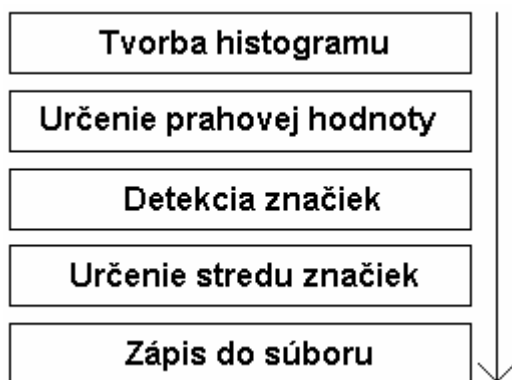
3.6 Analýza chôdze

Analýzu chôdze zabezpečuje program Analyser, ktorý zobrazuje 2D grafy. Tento program sme analyzovali a zistili sme niektoré nedostatky, ktoré sú popísané v časti analýzy existujúcich systémov, preto tento program bude treba mierne upraviť. Tiež možno upravíme množstvo a typy možných grafov.

4 Prototyp

V prototypy sme sa zamerali hlavne na segmentáciu obrazu, čo v podstate znamená, že sme sa zamerali na úpravu aplikácie VideoSQC. Mali sme k dispozícii dve nahrané videá. Jedno sme vytvorili použitím digitálneho fotoaparátu a druhé dvoma videokamerami, ktoré sme nahráli v softvérovom štúdiu.

Algoritmus:



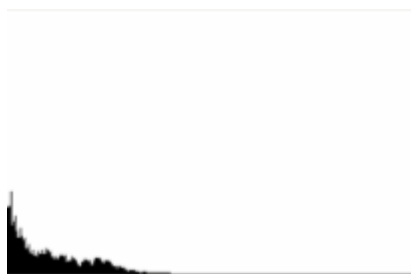
Obr. č.22 – Algoritmus segmentácie

4.1 Tvorba histogramu

Histogram zobrazuje závislosť jasů a počtu bodov.

Opis algoritmu:

1. pre každý bod snímku sa vypočíta jeho jas
2. pre každú hodnotu jasů sa zrátajú body, ktoré majú daný jas



Obr. č. 23 – Histogram videa

x-ová os tvorí hodnota jasů, od 0 po 255

y-ová os zobrazuje počet bodov s daným jasom

Z obrázku je vidieť, že bodov s nulovým jasom (čisto čierne) bolo najviac. A naopak svetlých bodov najmenej. Svetlé body reprezentujú body značiek, keďže sú reflexné a svetlo sa od nich odrážalo.

4.2 Určenie prahovej hodnoty

Prahová hodnota sa určuje z histogramu. Prahovú hodnotu je veľmi dôležité určiť správne. Vo veľkej miere úspešnosť algoritmu závisí práve od zvolenia tejto prahovej hodnoty. Prahová hodnota udáva jas. Všetky body, ktoré majú väčší jas ako prahový, patria značke. Tie body, ktoré majú menší jas, nepatria k značke a teda jas sa im nastaví na nulu. Táto časť bude predmetom ďalšieho experimentovania. Dočasne nastavujeme túto hodnotu ako konštantu priamo v programe. Vybrali sme takú hodnotu, s ktorou algoritmus dosahuje najlepšie výsledky.

4.3 Detekcia značiek

Na vstup sme dostali body s určitým jasom. Väčšina bodov má nulový jas. Tie body, ktoré majú väčší jas, môžu patriť značke, ale nemusia. Okolnosti, pre ktoré môžu mať iné body väčší jas, môžu byť rôzne. Môže to byť vďaka zlému osvetleniu alebo iným svetlým bodom, ktorý sa nachádzal v scéne. Výber scény je veľmi dôležitý, lebo veľmi ovplyvňuje výsledok algoritmu. Podrobnejšie je to zanalyzované v časti Analýza.

Opis algoritmu:

1. postupne prechádzame body snímku po riadkoch. Teda začne sa z ľavého horného rohu a postupuje sa po riadkoch.
2. ak sa nájde bod s väčším jasom, tak sa musí zistiť, či patrí značke. Body, ktoré patria jednej značke sú veľmi blízko seba.
3. preto sa prehľadá okolie toho bodu a ak sa nájde ďalší bod, pridá sa k ďalším, čo patria danej značke. Takto postupne identifikujeme všetky body patriace jednej značke. Okolie, ktoré sa prehľadáva je nastavené na konštantnú hodnotu.
4. algoritmus postupuje ďalej z miesta, kde našiel prvý bod predchádzajúcej značky.
5. algoritmus sa ukončí, ak príde k poslednému bodu snímku, teda k pravému dolnému rohu.

Výstupom tohto algoritmu je zoznam značiek. Pre každú značku sú zapísane zhľuky bodov, teda body, ktoré boli identifikované pre danú značku.

4.4 Určenie stredu značiek

Na vstup sme dostali súradnice bodov, ktoré patria jednej značke.

Opis algoritmu:

1. zo súradníc bodov určíme krajné x-ové a y-ové súradnice
2. vypočítame stred

Algoritmus je pomerne jednoduchý, ale myslíme, že je účinný a dostačujúci. Samozrejme, že ho vykonáme pre každú značku. Výstup algoritmu je konečne zoznam identifikovaných značiek. Teda mal by byť, kvôli rušivým vplyvom okolia, zlej kvalite videa sa mohli identifikovať aj iné body.

4.5 Zápis do súboru

Po identifikovaní všetkých značiek ich stačí zapísať do súboru. Formát súboru sme použili už existujúci, s príponou bbd. Tento súbor potom má slúžiť ako vstup do animačného programu BlueBone. Bohužiaľ tento program už ďalej nepodporuje tento formát. Pravdepodobne predchádzajúci tím ho stadiaľ odstránil. BlueBone podporuje už len formát bba, ktorý je ale komplikovanejší na výpočet. Rozhodli sme sa, že v prototype to nebudeme už implementovať z časových dôvodov.

4.6 VideoSQC

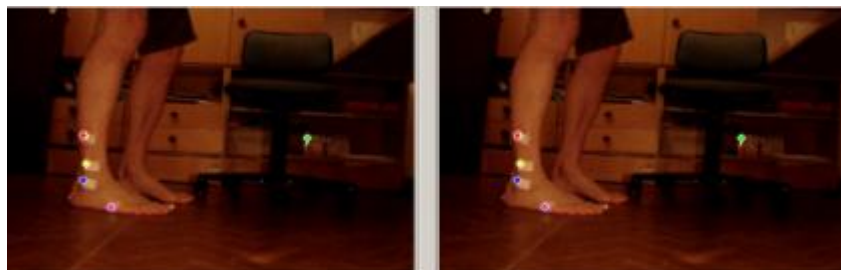
Zmeny sme robili len v tomto programe. Spravili sme ďalší histogram, pridali sme viac možností nastavenia a hlavne vylepšili algoritmus na segmentáciu. Tento program považujeme za najdôležitejší, preto sme mu venovali najväčšiu pozornosť. Od tohto programu závisí úspešnosť projektu. Cieľom je dosiahnuť úspešnosť detekovania značiek blížiacu sa k 100%. Predchádzajúci tím riešil značky, ktoré im ich algoritmus nedetekoval, pridaním funkcionality na manuálne pridanie značky. Tento spôsob sa nám nevidí práve najvhodnejší,

lebo pri malej úspešnosti detekovania značiek a pri veľkom počte snímkov je to pomerne náročná a dlhotrvajúca úloha. My plánujeme v letnom semestri implementovať algoritmus, ktorý by takú značku vypočítal z predchádzajúceho a nasledujúceho snímku, čo považujeme za vhodnejší spôsob. Samozrejme, pôvodnú možnosť manuálneho pridania značky ponecháme.

4.7 Výsledky segmentácie

V tejto časti uvádzame dosiahnuté výsledky segmentácie pre obe animácie, ktoré sme mali k dispozícii. Obrázky sú získané priamo z programu VideoSQC.

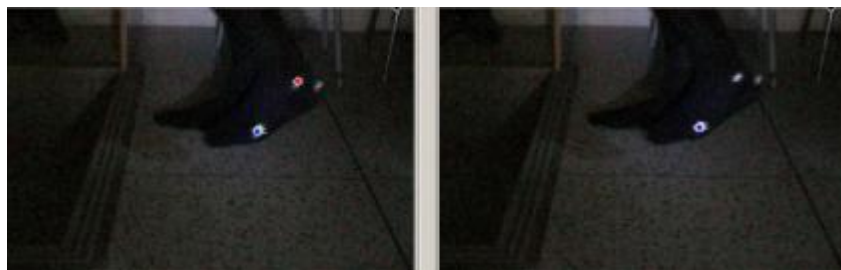
1. animácia, vytvorená digitálnym fotoaparátom



Obr. č. 24 – Segmentácia prvého videozáznamu

Z obrázku je vidieť, že algoritmus identifikoval všetky značky správne. Avšak v pravej časti obrázku identifikoval iný bod ako značku. Je to spôsobené zlým výberom scény. V scéne sa na tom mieste nachádzali veľmi svetlé body, ktoré algoritmus identifikoval ako značku. Problém sa dá odstrániť vhodnejším výberom scény na nahrávanie videozáznamu.

2. animácia, vytvorená dvoma videokamerami v softvérovom štúdiu



Obr. č. 25 – Segmentácia druhého videozáznamu

Na segmentáciu sme v prototypu použili len výstup jednej kamery. Z obrázku je vidieť, že algoritmus identifikoval správne len dve značky z troch. Na mieste, kde je tretia značka je ale

vidieť jeden svetlý bod. Čo je ale nedostatočné, lebo algoritmus ráta s tým, že jednej značke budú patriť viacero bodov. Má to slúžiť na odstránenie rušivých bodov, lenže v tomto prípade to zlyhalo. Nevýhodou tohto videozáznamu je jeho zlá kvalita a hlavne slabé osvetlenie. Pri použití lepšieho osvetlenia by tento problém nenastal a dosiahli by sme väčšiu úspešnosť, blížiacu sa k 100%.

4.8 Záver zimného semestra

V oblasti implementácie prototypu segmentačnej časti badať napriek drobným neúspechom markantné zlepšenie v porovnaní s prácou v minulých rokoch. Za predpokladu vylepšenia technickej stránky realizácie experimentu za účelom vytvorenia kvalitnejšieho vstupu pre segmentačný modul systému sa dá očakávať dosiahnutie veľmi uspokojivých výsledkov. Úsilie venované technickej realizácii reflexných značiek sa badateľne prejavilo v zlepšení výsledkov experimentu poskytnutím validných dát segmentačnému algoritmu.

K významným zmenám došlo v aplikácii VideoSQC, kde došlo k vylepšeniu algoritmu segmentácie. V ďalšej fáze riešenia projektu je potrebné koncentrovať úsilie na doladenie segmentačného algoritmu, implementáciu formúl epipolárnej geometrie, ktorá umožní zmapovať súradnice získaných bodov do 3D, a v neposlednom rade bude potrebné jednak prispôbiť animačný program BlueBone novým podmienkam a upraviť jeho funkcionality pre použitie v praxi, nakoľko jeho súčasný stav je nepostačujúci.

V experimentoch sa v ďalšej dekáde riešenia projektu treba na základe nami získaných skúseností zamerať na zvýšenie kvality osvetlenia reflexných značiek, nakoľko osvetlenie pri experimente v zimnom semestri bolo nedostačujúce.

Výsledky a skúsenosti, ktoré sme doposiaľ pri riešení projektu získali, nám umožnia určiť vhodné priority úloh, na ktoré sa bude treba zamerať v ďalšej etape vývoja systému a jeho aplikácie do praxe.

5 Algoritmy a experimenty

5.1 Lokálny histogram

Pre účely segmentácie bola implementovaná nová funkcia `GetLocalTreshold()` do triedy `C4Fbitmap`. Táto funkcia vracia prahovú hodnotu z lokálneho histogramu.

Samotná funkcia v sebe obsahuje tvorbu tohto lokálneho histogramu na základe argumentov tejto funkcie, ktorými sú:

- Súradnica značky z predchádzajúceho snímku
- Veľkosť strany štvorca, ktorý sa bude brať do úvahy pri tvorbe histogramu

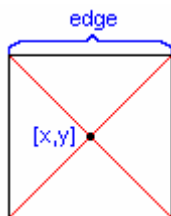
Prvý argument teda predstavuje súradnicu, v ktorej okolí sa bude počítat histogram a prahová hodnota. Druhý argument predstavuje veľkosť tohto okolia.

Segmentačný algoritmus volá túto funkciu pre každú nájdenú značku z predchádzajúceho snímku zvlášť, čím sa zabezpečí urýchlenie celého segmentačného algoritmu, pretože sa neprehľadáva celý frame, ale len jeho malé časti.

Myšlienka celého algoritmu je v tom, že keď vieme súradnice jednotlivých značiek z predchádzajúceho snímku, tak tieto sa v nasledujúcom snímku zmenia len o maličkú hodnotu. Stačí nám teda hľadať súradnicu danej značky v blízkosti značky z predchádzajúceho snímku.

Na to, aby sme vedeli rozpoznať súradnicu značky v nasledujúcom snímku, potrebujeme poznať prahovú hodnotu v lokálnom histograme tejto značky z predchádzajúceho snímku.

Na výpočet tejto prahovej hodnoty slúži práve funkcia `GetLocalTreshold()`, ktorej argumentami sú hodnoty x , y a $edge$ (Pozri obrázok č. 1).



Obr.č.26 - Význam argumentov

Kde:

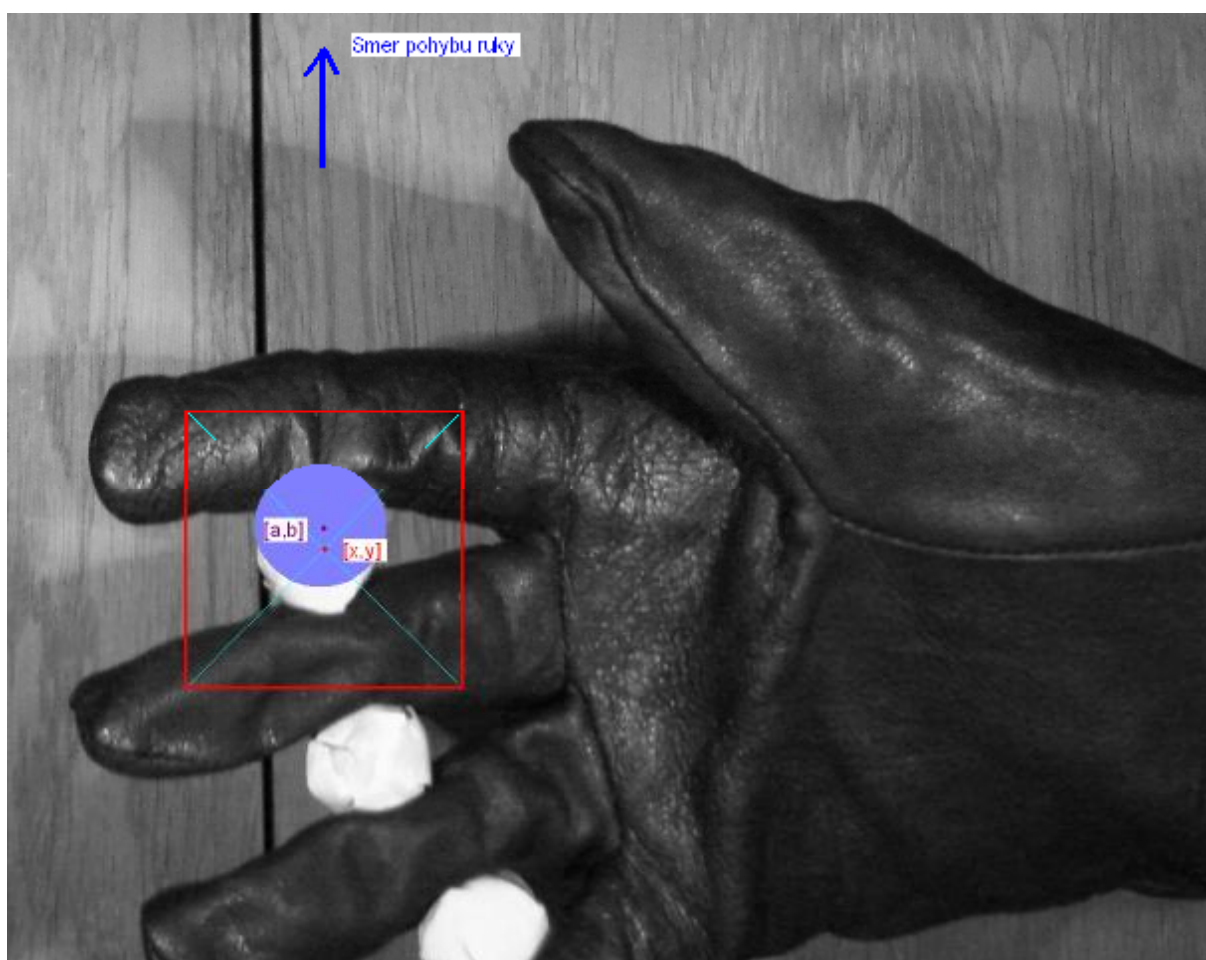
x ... x -ová súradnica značky z predchádzajúceho snímku

y ... y -ová súradnica značky z predchádzajúceho snímku

edge ... veľkosť strany štvorca bitmapy, ktorý sa bude uvažovať do histogramu

Na obrázku číslo 2 je znázornené, ako sa značka pohla pohybom sledovaného objektu.

Predpokladom je, že značka ostane aj pri pohybe objektu v v znázornenom štvorci. Lokálnym histogramom potom vieme jednoznačne určiť, ktorým smerom sa táto značka pohla a aké sú jej aktuálne súradnice v nasledujúcom obrázku.



Obr.č.27 - Pohyb sledovaného objektu

Kde:

$[x,y]$... súradnice stredu značky z predchádzajúceho snímku

$[a,b]$... súradnice stredu značky z nasledujúceho snímku

Súradnice [a,b] sa teda zistia segmentačným algoritmom vo vykreslenom štvorci na základe prahovej hodnoty, ktorú vypočítame vo funkcii GetLocalTreshold().

5.2 Experimenty zamerané na zlepšenie segmentácie

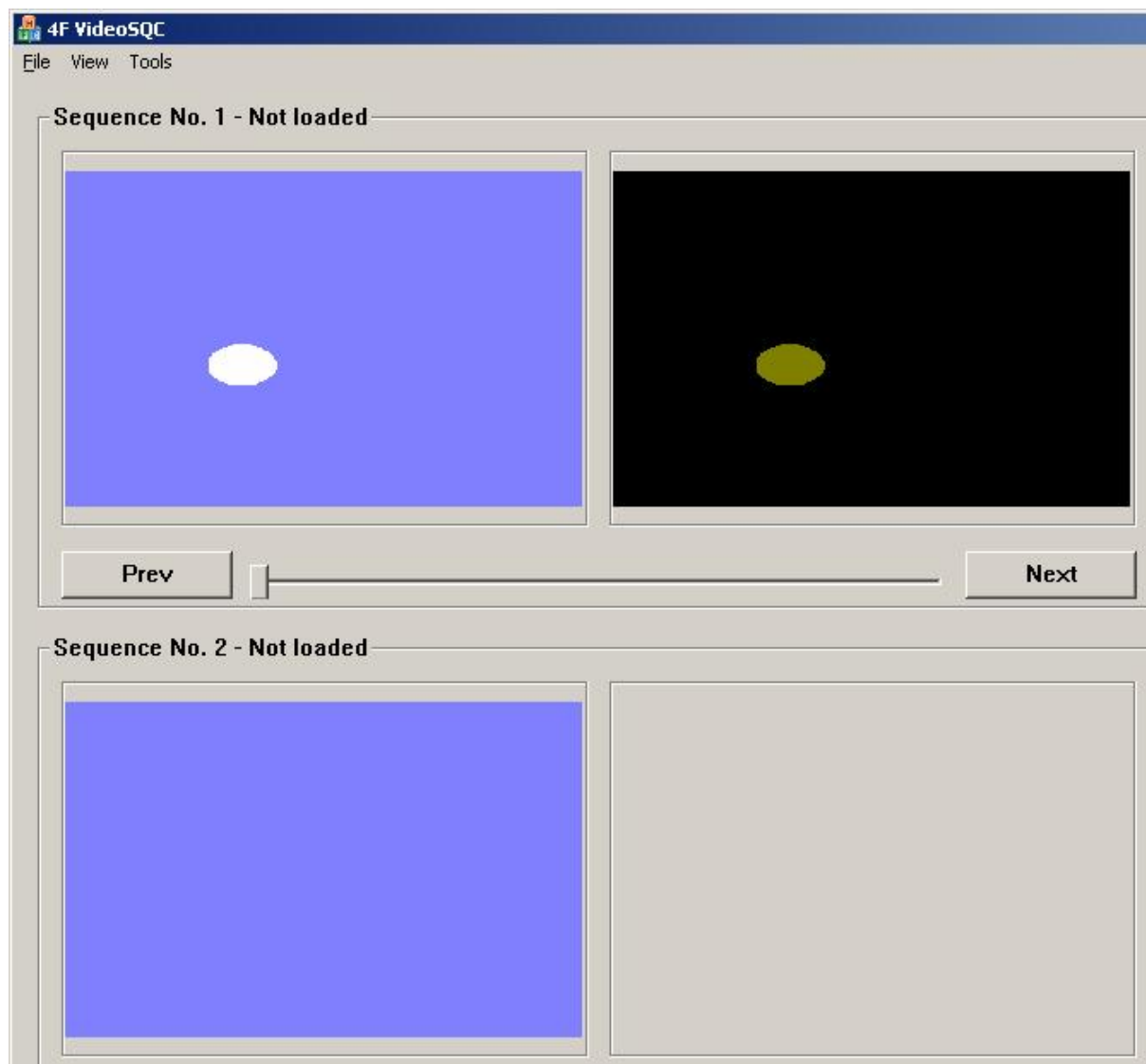
V tejto kapitole sme sa zamerali na experimentovanie s algoritmami zameranými hlavne na zlepšenie procesu segmentácie. Sú to algoritmy na odčítanie scén a grayscale. Uvedieme tu výhody a nevýhody každého z nich a zdôvodnenie, prečo ho použijeme/nepoužijeme v procese segmentácie. Na experimenty sme použili program VideoSQC, do ktorého sme doimplementovali potrebnú funkcionálnu schopnosť na vykonanie experimentu.

Algoritmus na odčítanie scén

Súčasný algoritmus segmentovania je pomerne náročný na prípravu scény. Je potrebné pripraviť veľmi dobré pozadie na jeho úspešnosť. V pozadí by nemali byť žiadne svetlé predmety a iné rušivé vplyvy. Najsvetlejším bodom by mali byť len samotné značky. Takúto scénu je pomerne ťažké pripraviť. Pri našich doterajších experimentoch sme sa stretli zakaždým s nejakým rušivým vplyvom okolia. Na odstránenie tejto závislosti od vhodnosti prostredia sme sa rozhodli vyskúšať práve algoritmus na odčítanie scén.

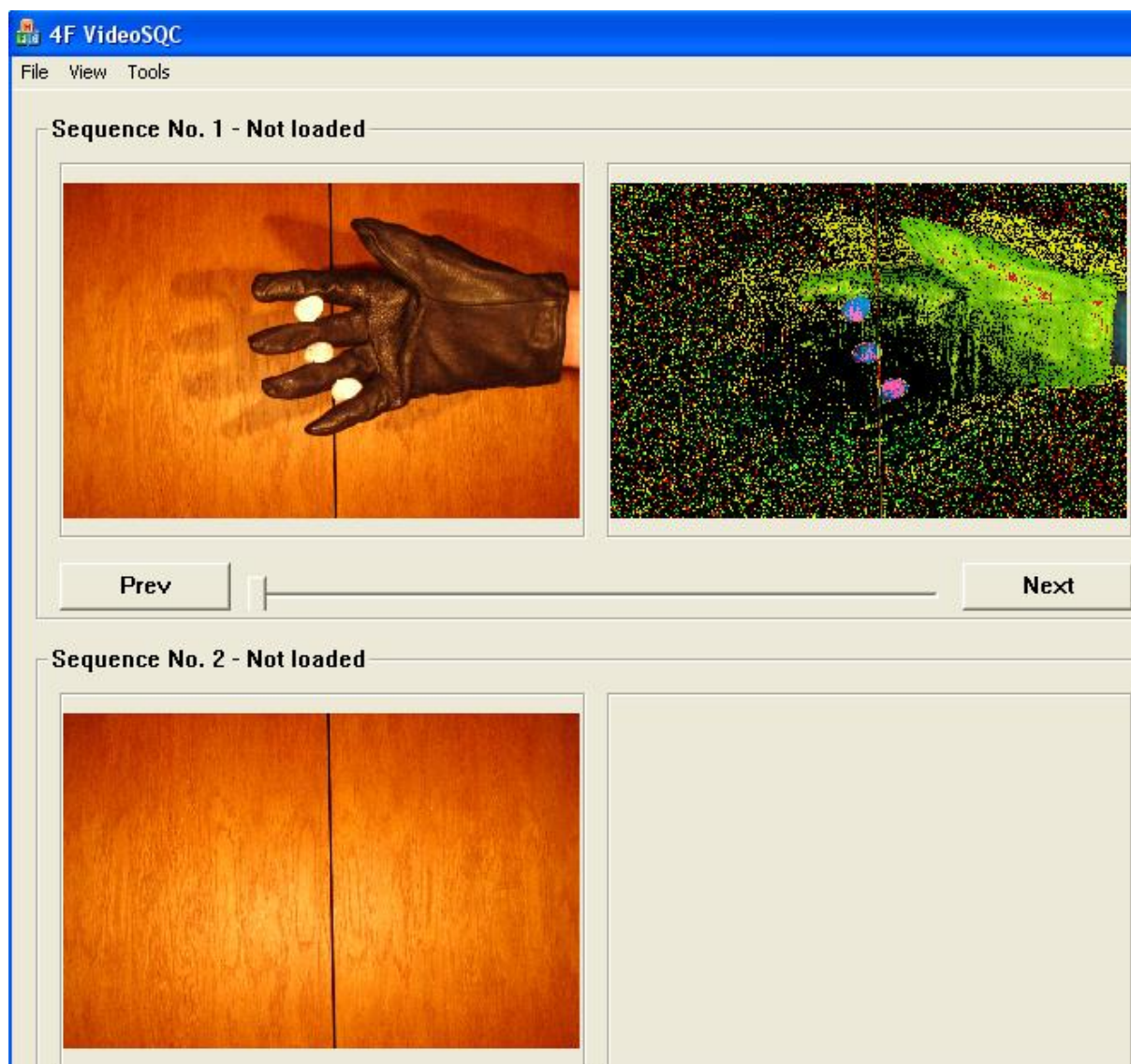
Princíp algoritmu spočíva v tom, že získame najprv obrázok samotného prostredia. Tento snímok bude tzv. podkladový. Od každého natočeného snímku s pokusnou osobou sa bude odčítavať práve snímok podkladový. Tím chceme docieľiť, aby sa rušivé vplyvy okolia odstránili.

Naše pokusy sme začali s ideálnymi a jednoduchými snímkami. Na obrázku č.28 vidíte časť programu VideoSQC s tromi obrázkami. Prvý obrázok reprezentuje pokusný snímok, pod ním je podkladový snímok a úplne vpravo hore je výsledok odčítania snímkov. Na výslednom obrázku zostala samotná elipsa a celé okolie je čierne. Toto je ideálny prípad.



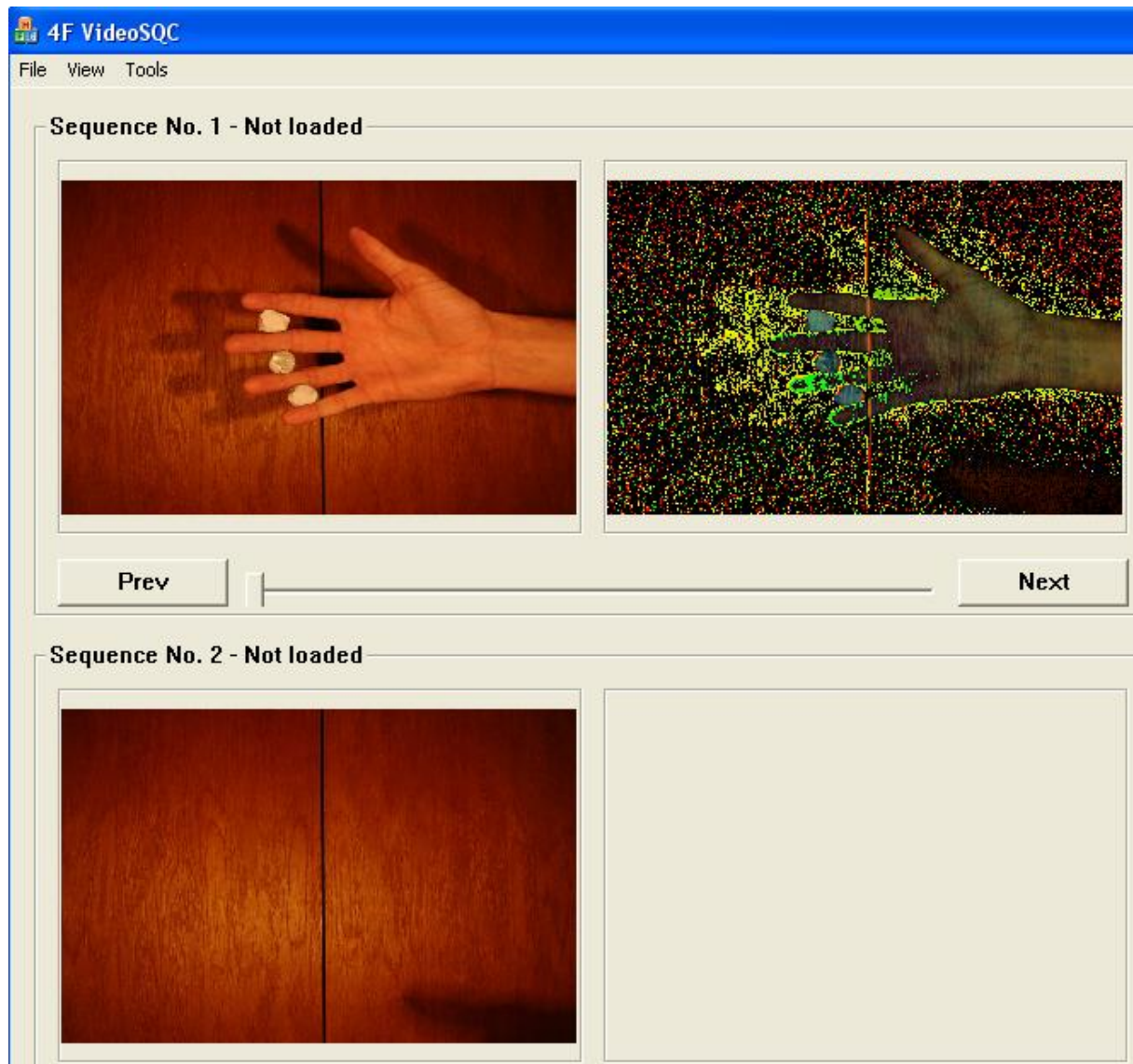
Obr.č.28 – Algoritmu na odčítanie obrázkov, scéna č.1

Na obrázku nám po odčítaní (výsledný obrázok je vpravo hore) zostala len elipsa. Okolie je kompletne čierne, čo je výborné. V prípade zložitejšieho snímku by sme takto odstránili rušivé vplyvy. Elipsa zmenila farbu, znížil sa jej celkový jas. S týmto musíme počítať, lebo celkový jas výsledného snímku sa vždy zníži. Teda musíme znížiť aj hraničnú hodnotu segmentácie. Zatiaľ to vypadá dobre. Na ďalšom obrázku uvedieme príklad odčítania dvoch obrázkov získaných z digitálneho fotoaparátu.



Obr.č.29 - Algoritmu na odčítanie obrázkov, scéna č.2

Na horeuvedenom obrázku vidíme tri reflexné značky. Čierna rukavica je použitá na zvýraznenie kontrastu. Obrázok je veľmi dobre osvetlený, čím sme zvýraznili reflexné značky ešte viac. Výsledok odčítania obrázkov je vpravo hore. Je vidieť, že od ideálneho prípadu je značne vzdialený. Na ďalšom obrázku je ten istý prípad, len teraz bez rukavíc a s horším osvetlením.



Obr.č.30 - Algoritmu na odčítanie obrázkov, scéna č.3

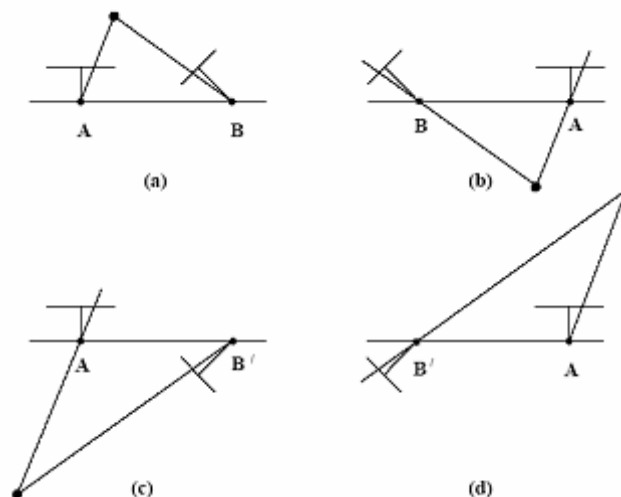
Tieto obrázky sa viac blížia k reálnejším podmienkam. Je horšie osvetlený. Výsledok odčítania ale tiež nespĺnil naše očakávania. Značky sú veľmi nevýrazné. Preto sme sa rozhodli, že tento algoritmus nebudeme používať v procese segmentácie.

5.3 Epipolárna geometria

Pri výpočte 3D súradníc na základe 2D súradníc získaných z dvoch snímok je nutné vykonať nasledujúce kroky:

Treba určiť kalibračné matice K , K' obidvoch kamier použitých pre získanie snímok. Kalibračné matice sú zložené z vnútorných parametrov kamery ako je ohnisková vzdialenosť a umiestnenie stredu premietania. Kalibračné matice budú slúžiť na normalizáciu 2D súradníc. Ak ich nevieme určiť, môžu sa namiesto nich použiť jednotkové matice.

1. Na základe odpovedajúcich bodov na snímkach sa vypočíta fundamentálna matica F , ktorá bude popisovať vzťah medzi týmito bodmi. Pre súradnice x , x' toho istého objektu by malo platiť $x \cdot F \cdot x' = 0$. Existuje viacero algoritmov ako vypočítať F a treba zvoliť ten, ktorý bude dávať najlepšie výsledky. Ak sa pri počítaní používajú normalizované súradnice (pôvodné súradnice sú pre násobené kalibračnou maticou), získame namiesto F esenciálnu maticu E .
2. Ďalším krokom je výpočet projekčných matíc. Spraví sa singulárny rozklad (SVD) fundamentálnej matice F (príp. E), ktorého výsledkom sú tri matice rozmerov 3×3 $F = UDV^T$. Z nich sa dajú zostrojiť 4 projekčné matice: $P' = [UWV^T | u_3]$ alebo $[UW^T V^T | u_3]$ alebo $[UWV^T | -u_3]$ alebo $[UW^T V^T | -u_3]$. Tie sa líšia iba posunutím a orientáciou ich pohľadu na scénu. Projekčné matice treba otestovať na ľubovoľnom bode a vybrať tú správnu, resp. zistiť, či bod leží pred obidvoma kamerami.



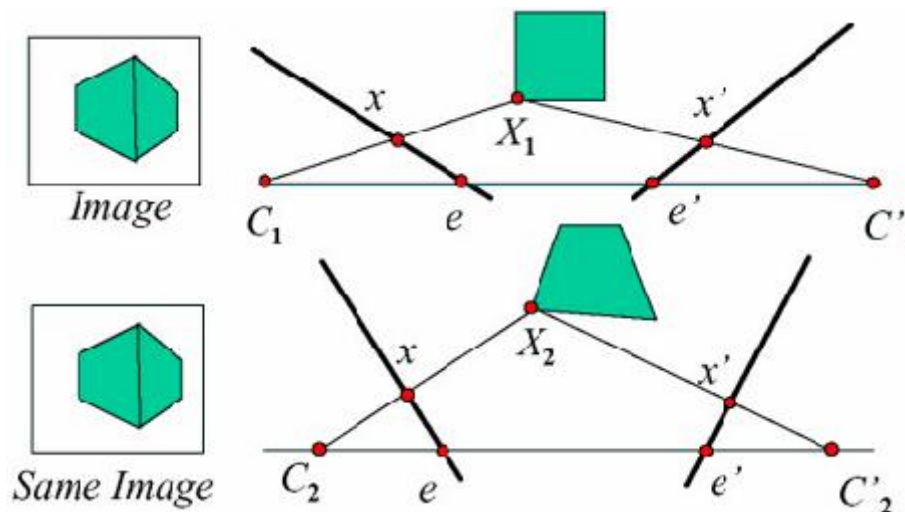
Obr. 31 - 4 možnosti orientácie a polohy kamier.

3. Ak už máme projekčné matice P (používa sa jednotková matica) a P' , môžeme pomocou metódy lineárnej triangulácie zrekonštruovať súradnice bodov. Získame sústavu lineárnych rovníc:

$$\begin{bmatrix} xP^{3T} - p^{1T} \\ yP^{3T} - p^{2T} \\ x'P'^{3T} - p'^{1T} \\ y'P'^{3T} - p'^{2T} \end{bmatrix} X = 0$$

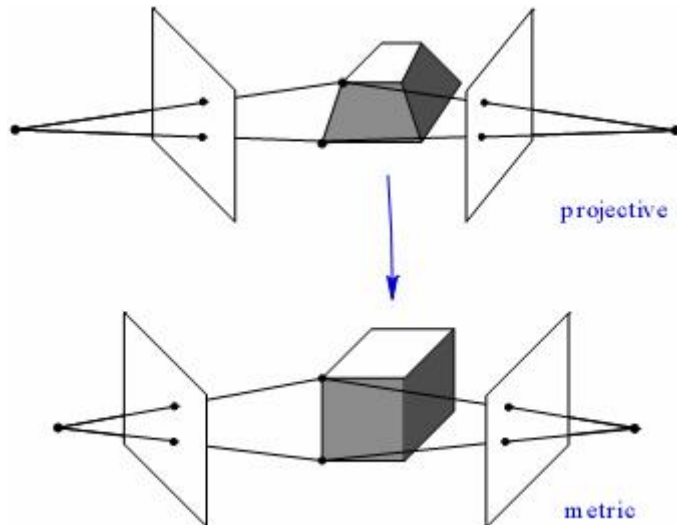
Na ich vyriešenie použijeme metódu najmenších štvorcov a tým dostaneme 3D súradnice bodu X . Premenné x, y predstavujú 2D súradnice, p^i je i -ty riadok matice P .

Pokiaľ sa pri výpočte nepoužívali normalizované súradnice, je nutné vypočítané 3D súradnice ešte upraviť. Ak sa nepoužívajú kalibračné matice, pozície bodov získaných zo snímok môže odpovedať viacerým objektom.



Obr. 32 - Dva rôzne objekty, ktoré majú rovnaký tvar na snímkach.

Na úpravu súradníc je potrebné vypočítať maticu homografie H , ktorá bude transformovať projektívne súradnice do normalizovanej súradnicovej sústavy. K jej vypočítaniu je treba mať k dispozícii 5 a viac 3D súradníc, ktorých polohu na scéne presne poznáme.



Obr. 33 - Matica homografie pretransformuje vypočítané súradnice do normalizovanej súradnicovej sústavy.

Riešenie

Aby sme overili správnosť krokov pre výpočet 3D súradníc a zároveň zistili, s akou presnosťou ich dokážeme vypočítať, rozhodli sme sa vytvoriť testovací program realizujúci celý výpočet. Pri jeho písaní bolo výhodné použiť voľne dostupné knižnice OpenCV a GSL, ktoré nám uľahčili prácu pri rekonštrukcii súradníc zo snímok.

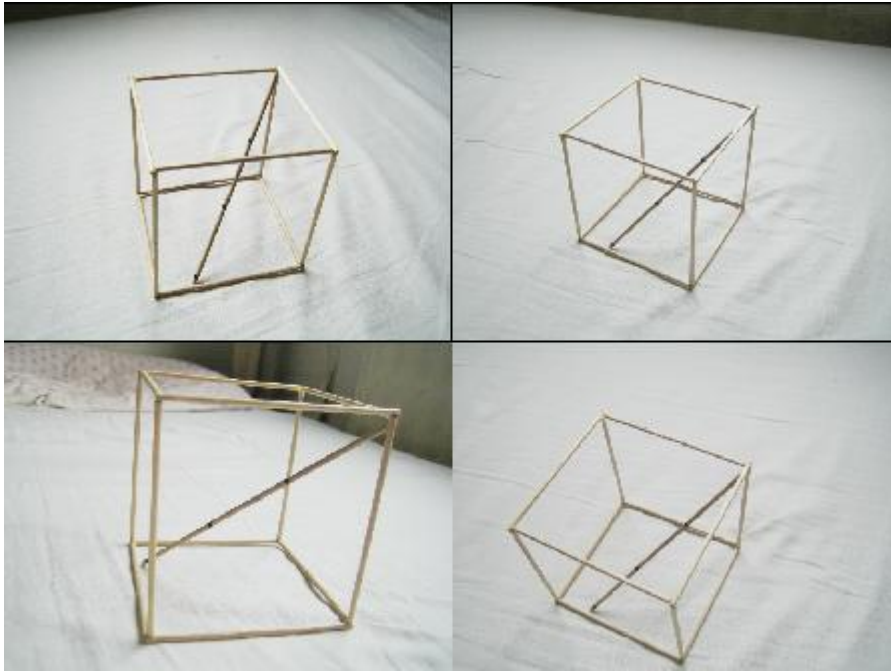
OpenCV je knižnica, ktorá je priamo určená na spracovanie obrazu a poskytuje množstvo funkcií, ktoré by sa dali využiť v rámci nášho projektu. Konkrétne pri výpočte 3D súradníc sa uplatnia funkcie na určenie kalibračných matíc a na výpočet fundamentálnej matice.

Knižnica GNU Scientific Library (GSL) je rozsiahla matematická knižnica pre C a C++ programátorov. Z nej je pre nás zaujímavá hlavne časť venujúca sa práci s maticami, funkcie na výpočet SVD a časť venujúca sa minimalizačným metódam.

Fundamentálna matica

Pri výpočte fundamentálnej matice je potrebné mať k dispozícii niekoľko (minimálne 7) dvojíc súradníc bodov. Pre tento účel sme zostrojili model kocky (na obrázku), pričom jeho rohy sú považované za značky, ktoré budú použité v algoritme.

Súradnice značiek sú určené ako pozícia pixlov na obrázku a sú uložené do súborov v rovnakom poradí (najprv rohy podstavy, potom horného štvorca a nakoniec značky na paličke). Každý takýto súbor predstavuje množinu súradníc získaných z jednej snímky.



Obr.34 - Snímky modelu kocky. Súradnice rohov a značiek na paličke umiestnenej v kocke sú vstupnými údajmi pre výpočet fundamentálnej matice.

Pri určovaní fundamentálnej matice sa používa funkcia z OpenCV *cvfindFundamentalMat*, ktorá umožňuje použiť na jej výpočet viacero metód, konkrétne:

základný sedem-bodový algoritmus

základný osem-bodový algoritmus

algoritmus RANSAC

„least mean square“ algoritmus

Aby sa zistilo, ktorá z metód nám poskytne najpresnejšiu fundamentálnu maticu, boli vykonané testy pre každú z nich. Zároveň bolo testované aký má vplyv počet poskytnutých značiek na výsledok.

Na posúdenie presnosti riešenia bol využitý fakt, že pre F a dvojicu zodpovedajúcich si bodov m platí $x \cdot F \cdot x' = 0$. To však platí iba v ideálnom prípade – v praxi býva výsledkom nejaké nenulové číslo. Pre každú dvojicu snímok sa teda určila priemerná absolútna odchýlka

od nuly $E = \frac{1}{n} \sum_{i=1}^n abs(x_n \cdot F \cdot x'_n)$. V nasledujúcej tabuľke sú uvedené dosiahnuté výsledky.

Metóda	4-5	4-6	4-7	5-6	5-7	6-7	Priemer
7POINT	0,6883	0,7019	0,1743	2,2489	0,1954	0,4179	0,7378

8POINT - 8	0,3100	5,3621	0,3744	0,1924	0,1951	0,2676	1,1169
8POINT - 11	0,2496	0,5471	1,6932	0,2168	0,2955	0,4437	0,5743
RANSAC - 8	1,5556	0,7827	14,9959	0,5202	0,1774	0,4546	3,0811
RANSAC - 11	2,3461	0,3595	10,6615	0,5612	0,3990	0,1302	2,4096
LMEDS - 8	9,5506	1,0217	10,1508	1,6107	0,6301	0,4546	3,9031
LMEDS - 11	6,1261	20,2759	0,8315	1,7515	2,5225	1,9511	5,5764

Testy boli uskutočnené iba na súboroch číslo 4 – 7, pretože tie už obsahovali súradnice 11 bodov (vo vnútri kocky je palička so značkami poskytujúca ďalšie 3 body na porovnávanie). Sedembodový algoritmus dokáže pracovať iba so siedmimi bodmi, zatiaľ čo ostatné metódy (potrebujú 8 a viac) boli vyskúšané na množine 8 a 11 bodov.

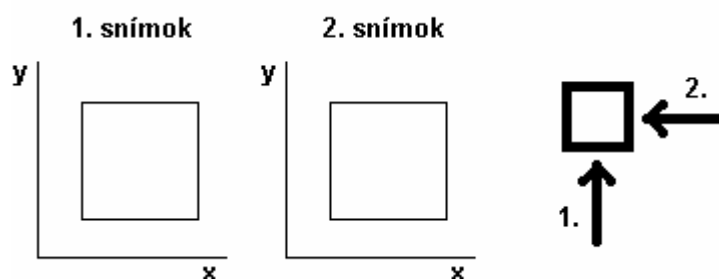
Z dosiahnutých výsledkov vyplýva, že najpresnejšiu fundamentálnu maticu dostaneme použitím základného osembodového algoritmu. Zároveň je vidieť, že pridaním ďalších bodov sa zlepšia dosahované výsledky - v prípade 8-bodového algoritmu už pridanie 3 bodov umožnilo dosiahnuť 50% zlepšenie.

Projekčné matice a výpočet súradníc

Po určení fundamentálnej matice F základným osembodovým algoritmom, sa táto matica rozloží a zostroja sa 4 alternatívy projekčnej matice. Súradnice vypočítané týmito maticami sa medzi sebou líšia znamienkom a vzájomnou polohou, vzdialenosti medzi nimi však zostávajú rovnaké.

V tejto fáze testovania nás skôr zaujímalo, ako presne sú 3D súradnice vypočítané, resp. či výsledný útvar nie príliš zdeformovaný, a nie či je výsledný útvar nejako natočený. Výber správnej projekčnej matice sme odložili na neskôr.

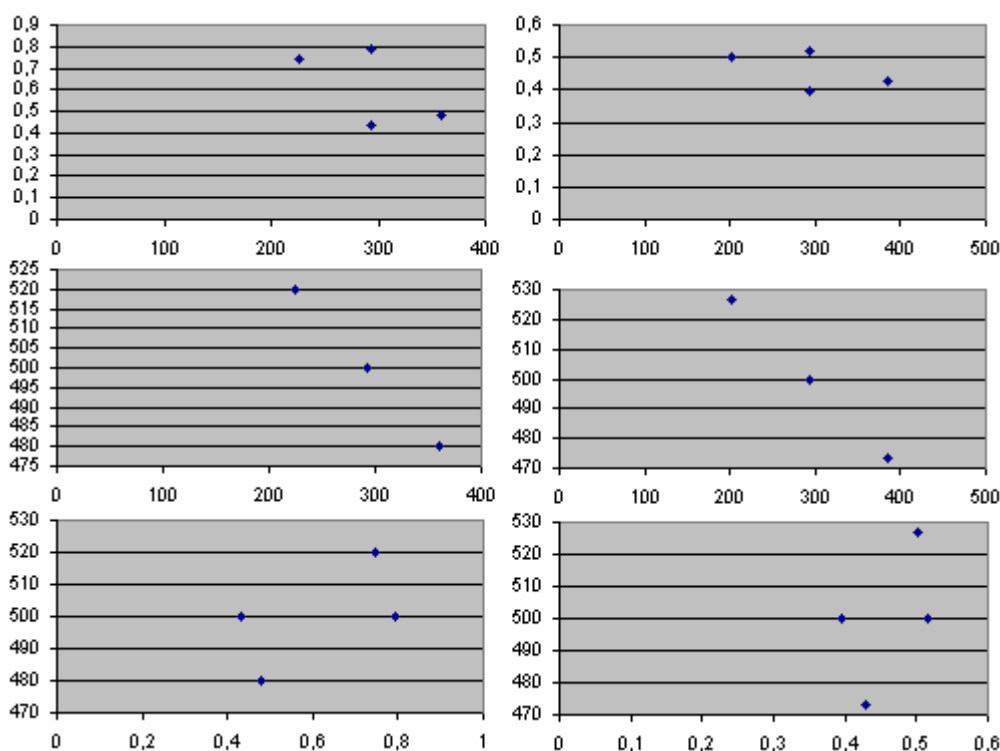
Pri prvom meraní sme použili nami pripravené dáta. Neboli odčítané zo snímok, ale boli určené spôsobom, ako je zobrazené na obr. 5. Chceli sme tak otestovať algoritmus na jednoduchom prípade ešte pred tým, ako prejdeme na reálne snímky.



Obr. 35 - Ideálna situácia. Predné body kocky zakrývajú zadné. Kamery sú umiestnené v rovnakej vzdialenosti a kolmo na seba

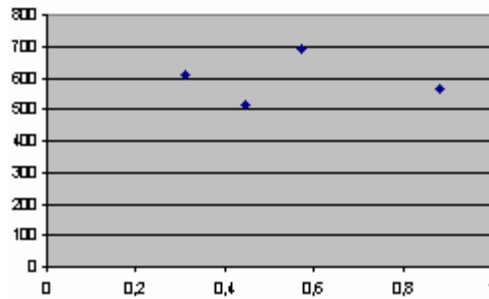
V nasledujúcej tabuľke sú uvedené vstupné hodnoty a hodnoty vypočítané našim algoritmom. Ako je z výsledkov vidieť (Obr. 6), vypočítané súradnice sú oproti pôvodným posunuté a pootočené, takže výsledný útvar je skosený. Je to spôsobené tým, že sme nepoužívali kalibračné kamery a chýba aj matica homografie, ktorá by súradnice pretransformovala do pôvodných súradníc. Súradnice z-ovej osi sú navyše o tri rády nižšie, nie je však problém ich upraviť, aby rádo sedeli s ostatnými.

Bod	Vstupný súbor		Vypočítané súradnice		
	x	y	x	y	z
1	500	500	292,584	499,761	0,792431
2	1000	500	359,757	480,008	0,479817
3	1000	500	292,634	499,975	0,43421
4	500	500	225,465	519,728	0,745532
5	500	1000	292,64	499,92	0,516989
6	1000	1000	384,377	473,067	0,42872
7	1000	1000	292,652	499,993	0,396344
8	500	1000	200,914	526,836	0,501857



Obr. 36 - Súradnice vypočítaného útvaru. Vľavo je zobrazená spodná stena kocky, vpravo vrchná, pričom prvá dvojica predstavuje pohľad zhora, nasleduje pohľad spredu a z boku.

Výsledný útvar je síce deformovaný, ale stále pripomína skosenú kocku. Pri použití reálnych snímok však dochádza ešte k väčšiemu skresleniu a navyše sa niektoré body nachádzajú v značnej vzdialenosti od ostatných (Obr. 37).



Obr. 37 - Ukážka zdeformovania podstavy kocky pri použití dát z reálnych snímok. Aj pri použití rôznych kombinácií snímok zostáva deformácia na tej istej úrovni.

Do akej miery je deformácia 3D súradníc spôsobená používaním nenormalizovaných súradníc v tejto chvíli nedokážeme povedať. Z časových dôvodov sa nám nepodarilo implementovať výpočet kalibračných matic (poprípade maticu homografie), čo by bol posledný zostávajúci krok pri výpočte súradníc. Zároveň by nám to umožnilo porovnať skutočné a vypočítané súradnice a určiť celkovú presnosť.

Výsledky získané pomocou nenormalizovaných súradníc sú však porovnateľné s ukázkami v študijných materiáloch. Tie sa od našich nejako výrazne nelíšia a môžeme predpokladať, že doteraz vykonané kroky nevnášajú do výpočtu súradníc žiadnu veľkú chybu.

Aj napriek tomu, že momentálne nevieme určiť presnosť, s akou budú 3D súradnice zodpovedať skutočnosti, bolo by výhodné v budúcnosti používať epipolárnu geometriu. Kvalita dosahovaných výsledkov by mala závisieť čisto iba na presnosti vstupných údajov, takže vypočítané 3D súradnice by mali byť zaťažené rovnakou chybou, aká bude aj pri použití iných metód výpočtu.

Použitie kamier umiestnených kolmých na seba (zvažované v predchádzajúcich prácach) prináša so sebou niekoľko problémov. Asi najväčším je nutnosť dôkladnej prípravy scény. Kamery musia byť v rovnakej výške a vzdialenosti od snímaného objektu a ich osi musia byť na seba kolmé. Natočené snímky by tak mali predstavovať pohľad spredu a z boku na rovnaké body. V praxi však nie je dosiahnuteľné, aby súradnice bodov z oboch snímok dokonale sedeli (napr. bude mierne posunutá y-ová súradnica) a budú sa musieť nejakým spôsobom upravovať, čím sa môžu do výpočtu súradníc vniesť chyby.

Hlavnou výhodou epipolárnej geometrie je schopnosť pracovať so snímkami z ľubovoľných pozícií a teda vyššie opísané problémy odpadajú. Aj pri použití kolmých kamier sa budú musieť súradnice nejakým spôsobom transformovať, takže z tohto pohľadu je situácia podobná. Otázna zostáva iba výsledná presnosť – tu by mohol mať miernu výhodu klasický prístup, aj keď za cenu nevyhnutnej prípravy scény.

5.4 Segmentácia pomocou programu VideoSQC

V programe VideoSQC sme vykonali najväčšie úpravy. Drvivá väčšina týchto úprav bola výskumného charakteru, kedy sme zisťovali možnosti použitia rôznych segmentačných algoritmov pre snímky. Postupne sme dospeli k nasledovnému algoritmu, ktorý vykazoval celkom dobré vlastnosti:

Algoritmus segmentácie pomocou rekurzívnej záplavovej metódy hľadania značiek:

Všetky použité algoritmy si obrazovú sekvenciu snímok rozdelili do jednotlivých snímok. Tento algoritmus postupne prechádzal cez snímky, pričom hľadal značky nasledovným spôsobom: Na začiatku sa určila prahová hodnota intenzity (threshold), pričom body na obrázku, ktoré sú jasnejšie ako daná intenzita sa označili, že môžu patriť do značky, a body, ktoré mali intenzitu nižšiu, sa označili, že nemôžu patriť do značky. Takýto postup sme volili pri oboch použitých algoritmoch. Postupne sa pre všetky body na obrázku (od prvého, až k poslednému) začali hľadať susedné body, ktoré mohli tiež patriť do značky, tj. Mali intenzitu väčšiu ako threshold. Ak algoritmus našiel všetky body, ktoré môžu patriť do značky, ktoré boli v okolí prehládavanej značky, a už žiadne ďalšie neexistovali (v okolí), tak túto sadu bodov mohol označiť ako jednu značku. Označil ju len v tom prípade, ak spĺňala ďalšie kritérium, a tým bol minimálny počet bodov, ktorý mohol byť v značke. Určením minimálneho počtu bodov v značke sme chceli odstrániť prípadný šum s vysokou intenzitou, a okraje rozmazaných značiek, čo sa aj podarilo. Ak bol počet bodov väčší ako prahová hodnota, systém si uchoval všetky body, pre ďalšie použitie, a pokračoval ďalším bodom. Samozrejme sa počas celého vykonávania algoritmu každý bod, ktorý už bol prehládaný, označil do hashovacej tabuľky, aby už ďalej nebol prehládaný znova. Po prejdení všetkých bodov mal algoritmus uchované v poli tie zoznamy bodov, ktoré mohli tvoriť každý jednu značku. Ďalej sa muselo určiť, či niektoré z bodov, ktoré sú rozoznané ako rôzne značky, nie sú v skutočnosti jednou značkou. Pre toto sa využila ďalšia konštanta, a to minimálna

vzdialenosť v pixeloch medzi značkami. Postupne sa všetky vybrané body prehľadávali a zisťovalo sa, či nepatria jednej značke. Ak systém našiel body v dvoch značkách, ktoré boli blízko seba, spojil ich do jednej značky.

Nakoniec, algoritmus zistil stredy jednotlivých značiek. Toto zistil jednoduchým algoritmom, kedy našiel pre x aj y -ovú súradnicu minimálny a maximálny bod, a spravil z oboch priemer.

Algoritmus takto vrátil sadu bodov, ktoré boli stredmi značiek.

Algoritmus segmentácie s využitím lokálnych selekcií a výberu lekárom:

Základom pre tento algoritmus bola možnosť nechať si upresniť polohy značiek lekárom, a tiež znalosť (získaná pokusmi) že v praxi rýchlosť nohy nieje veľká, a preto môže byť hľadanie značiek usmernené na konkrétne oblasti, a nemusí sa hľadať globálne na celom obrázku. a preto mohol byť ďalšie hľadanie jednoduchšie. Na začiatku tento algoritmus segmentuje jednu snímku, a to predchádzajúcim popísaným algoritmom. Táto snímka sa potom zobrazí používateľovi. Tento má možnosť posunúť body tak, aby boli čo najideálnejšie, alebo si ich len prezrieť a prekontrolovať. Ak je všetko v poriadku, používateľ znova stlačí tlačítko segmentácie, a segmentácia bude pokračovať od druhého snímku, ale už iným, novým algoritmom. Tento algoritmus berie ako základ body z predchádzajúceho snímku, tj. v prípade druhého snímu, výsledky (stredy značiek) v prvom snímku. Značky nehľadá na celom snímku, ale len na časti. Postupne, pre každú značku z predchádzajúceho snímu, prehľadá jej okolie, a zistí kde sa značka nachádza. Toto hľadanie je umožnené, lebo značky sa nepohybujú veľmi rýchlo medzi snímkami a maximálny rozdiel vzdialenosti značiek na dvoch susedných snímkach je niekoľko pixelov. Takto sa výrazne znižuje celkový čas na trvanie algoritmu, ktorý je dosť dlhý. Vyhľadávanie na oblasti je omnoho rýchlejšie, keďže napr. oblasť o veľkosti $30 * 30$ pixelov, tj. 900 pixelov, je len asi 0.3% z celého snímku, tj. $640 * 480$, čo je 307200 pixelov. V prípade, že značka nieje dostatočne výrazná, môže sa použiť pôvodná hodnota, tj. predchádzajúca značka. Takto sa zabezpečí aj náhodné vypadnutie niektorej značky.

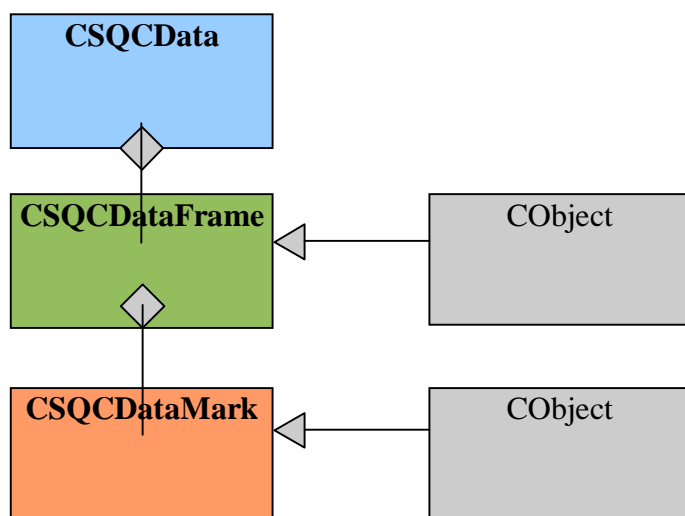
5.5 Integrácia BlueBone s VideoSQC

Implementácia rozhrania pre podporu VideoSQC formátu

Dôležitou časťou funkcionality celého systému analýzy ľudskej chôdze je integrácia animačnej aplikácie BlueBone so segmentačnou aplikáciou VideoSQC. V praxi to bude

znamená, že po vytvorení snímkov viacerými kamerami sa tieto dajú na vstup aplikácii VideoSQC, ktorá zabezpečí novú segmentačnú politiku v súlade s navrhnutým princípom *lokálnej segmentácie* a tvorbu výstupného súboru v textovom formáte s príponou *bbs*. Tento súbor obsahuje záhlavie a jednotlivé framy so zoznamom značiek a príslušnými súradnicami. Bolo potrebné, aby program BlueBone bol schopný čítať tento súbor značne odlišný od jemu vlastných vstupných súborov (*bba*, *bbd*) samozrejme spolu s naplnením dátových štruktúr a zabezpečením animovania vstupných dát. Inými slovami, BlueBone musí vedieť prečítať dáta z nasnímaného videa, nielen animačné dáta, ktoré boli vytvorené v jeho rozhraní.

Pre fungovanie tohto mechanizmu boli do aplikácie BlueBone pridané tri nové triedy a niekoľko členských funkcií k už existujúcim triedam. Diagram navrhnutých a implementovaných tried je na nasledovnom obrázku.



Obr. 38 - Diagram tried pre dáta načítané z aplikácie VideoSQC.

Triedy od seba navzájom nededia, ale sú v sebe zahrnuté kompozíciou ako polia objektov. Základná trieda **CSQCData** obsahuje nasledovné privátne dátové členy:

```

class CSQCData
{
private:
    UINT m_nSEQ;
    UINT m_n1; // FPS
    UINT m_nFrames;
    UINT m_n2; // number of loops
    CObArray *m_aFrames;
}
  
```

public:

```
CSQCData();
CSQCData(UINT,UINT,UINT,UINT,CObArray *);
virtual ~CSQCData();

void SetSEQ (UINT s) {m_nSEQ = s;};
UINT GetSEQ () {return m_nSEQ;};
void SetN1 (UINT n) {m_n1 = n;};
UINT GetN1 () {return m_n1;};
void SetNFrames (UINT f) {m_nFrames = f;};
UINT GetNFrames () {return m_nFrames;};
void SetN2 (UINT n) {m_n2 = n;};
UINT GetN2 () {return m_n2;};
void SetArrFrames (CObArray *a) {m_aFrames = a;};
CObArray *GetArrFrames () {return m_aFrames;};

void LoadSQCDData(const char* acFileName);
};
```

Trieda **CSQCData** obsahuje metódu **LoadSQCDData()**, ktorá zabezpečuje vytvorenie inštancií potrebných objektov na načítanie dát zo súboru aplikácie VideoSQC. Ako je vidieť, uchováva dáta zo záhlavia *bbs* – súboru, ako číslo sekvencie, rýchlosť animácie vo framoch za sekundu, počet framov, počet opakovaní animácie, a nakoniec pole referencií na objekty triedy **CSQCDataFrame**. Táto trieda vyzerá nasledovne:

```
class CSQCDataFrame : public CObject
{
private:
    UINT m_nFR; // frame ID
    UINT m_nLC; // number of marks
    CObArray *m_aSQCDDataMarks; // array of marks
public:
    CSQCDataFrame();
    CSQCDataFrame(UINT,UINT,CObArray *);
    virtual ~CSQCDataFrame();

    void SetFR (UINT fr) {m_nFR = fr;};
    UINT GetFR () {return m_nFR;};
    void SetLC (UINT lc) {m_nLC = lc;};
    UINT GetLC () {return m_nLC;};
    void SetSQCDDataMarks (CObArray *dm) {m_aSQCDDataMarks = dm;};
```

```

    CObArray *GetSQCDataMarks () {return m_aSQCDataMarks;};
};

```

Z jej štruktúry je badateľné, že trieda dedí od **CObject**, takže je možné ju vložiť do poľa objektov typu **CObArray**. Logicky reprezentuje jeden frame a uchováva jeho ID, počet značiek, ktoré na ňom boli zachytené, a pole ukazovateľov typu **CObject** na objekty triedy **CSQCDataMark**, ktorá reprezentuje konkrétny objekt značky:

```

class CSQCDataMark : public CObject
{
private:
    UINT m_nID;
    int m_x;
    int m_y;
    int m_z;

public:
    CSQCDataMark();
    CSQCDataMark(UINT,int,int,int);
    virtual ~CSQCDataMark();

    void SetID (UINT id) {m_nID = id;};
    UINT GetID () {return m_nID;};
    void SetX (int x) {m_x = x;};
    int GetX () {return m_x;};
    void SetY (int y) {m_y = y;};
    int GetY () {return m_y;};
    void SetZ (int z) {m_z = z;};
    int GetZ () {return m_z;};
};

```

Ako dátové členy obsahuje súradnice značky (tretia súradnica je zatiaľ nulová) a ID značky. S takto pripravenými dátami je možné v aplikácii BlueBone realizovať animáciu súradníc načítaných segmentačnou aplikáciou VideoSQC. Integrácia zahŕňa aj pridanie položky do menu aplikácie BlueBone, a síce „LoadSQC“, ktorá zabezpečuje načítanie súboru typu *bbs* aplikácie VideoSQC.

Samotné načítanie súboru je realizované cez metódu **CSQCData::LoadSQCData()**, ktorá zároveň naplňa všetky potrebné štruktúry. Vytvorenie objektu triedy **CSQCData** sa deje v **CBlueBoneDlg::OnAnimationLoadSQC()** cez základný štandardný konštruktor. Nahratie dát získaných z VideoSQC zabezpečuje metóda **CAnimation::LoadSQC()**.

Úskalia tvorby experimentálnej animácie

Segmentácia experimentálneho videa pripravila výstupný súbor s piatimi značkami, v tomto poradí *bedro*, *koleno*, *členok*, *záprstie* a *prsty*. Z uvedeného vyplýva, že ako vstup do animačného programu bude uvažovaná iba jedna noha, a to pri zázname iba z jednej kamery bez použitia epipolárnej geometrie, čiže animácia bude prebiehať v 2D. Na vykonanie 3D animácie by bolo potrebné, aby každá značka bola snímaná v každom čase aspoň dvoma kamerami, čo sa nám nepodarilo z dôvodu nepostačujúcich technických podmienok experimentu zrealizovať. O tomto bližšie pojednáva záverečná kapitola.

Keďže animačný objekt aplikácie BlueBone požaduje sériu desiatich trojíc súradníc značiek (kĺbov), pre druhú nohu sme dočasne použili súradnice prvej nohy na prvom snímku pre každý ďalší snímok. Takto sme dosiahli požadovaný počet súradníc desiatich kĺbov pre každý frame.

Požiadavky na vstupný súbor *.bbs

Hlavnou požiadavkou je, že súbor značiek musí byť kompletný pre každý uvažovaný frame. Toto musí zabezpečiť segmentačný algoritmus. V prípade, že dôjde k výpadku značky, je potrebné, aby bol frame dosyntetizovaný interpolačne, alebo prípadne úplne vypustený zo sekvencie (Pri rýchlosti 25 fps to nie je až taká tragédia.). Je veľmi dôležité, aby číslo v záhlaví *bbs* súboru zodpovedalo skutočnému počtu framov v sekvencii, inak pri načítavaní súboru aplikáciou BlueBone dôjde k výnimke.

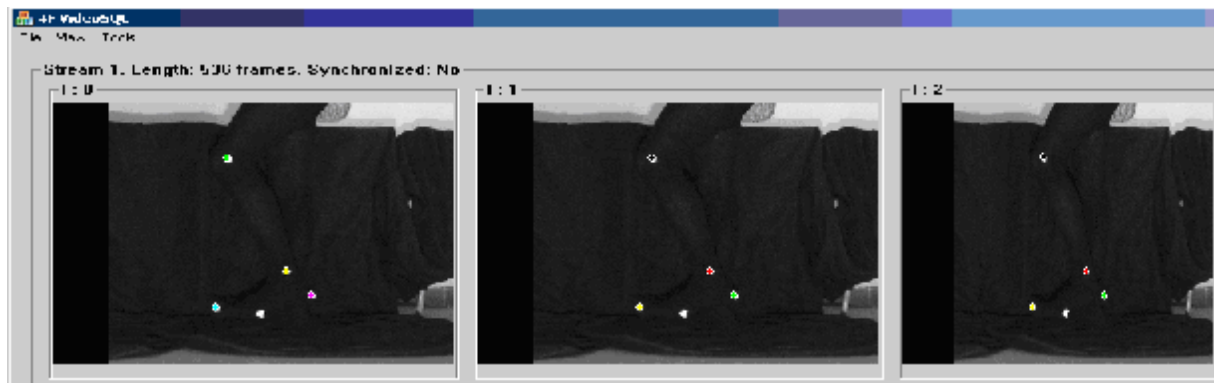
Druhou nemenej dôležitou požiadavkou je, aby značky boli v súbore v správnom vopred určenom poradí, ako je uvedené vyššie. V prípade, že to nebude dodržané, BlueBone priradí súradnice nesprávneho kĺbu a animácia bude obsahovať chybný frame.

Zhrnutie funkčnosti integrácie sekvencií do animačného programu

Modul, ktorý spracováva VideoSQC súbory bol otestovaný na experimentálnom videu natočenom v softvérovom štúdiu. Boli použité zábery z bočnej kamery. Animácia v programe BlueBone očividne kopírovala pohyby nohy z experimentálneho videozáznamu, čo sa dá zhodotiť ako úspech.

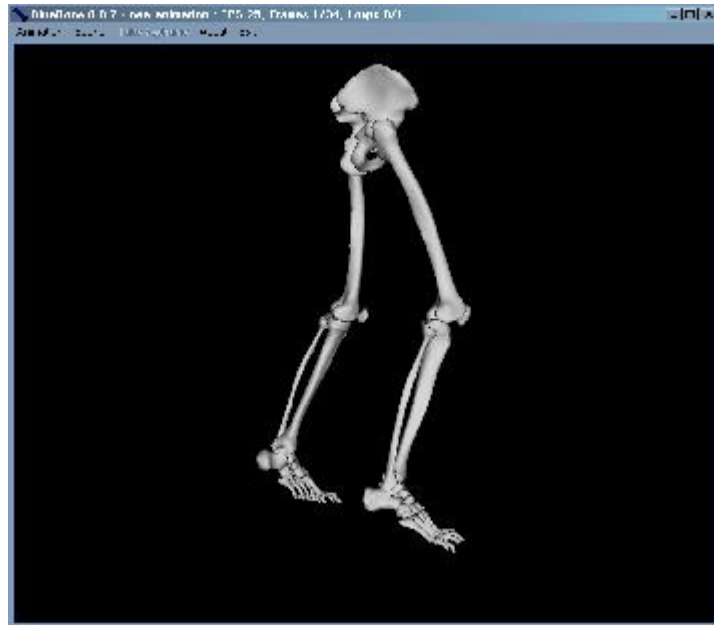
5.6 Výsledky

V tejto kapitole popíšeme dosiahnuté výsledky. Na segmentáciu sme použili posledné video, ktoré sme natočili. Bolo natočené s bežnou pomalou kamerou a dobre osvetlené jedným svetlom spoza kamery. Snímaný objekt sa pohyboval veľmi pomaly, tak aby to kamera stihla snímať bez rozmazania značiek. Získané video sme dekomprimovali, lebo program VideoSQC vie robiť len s dekomprimovanými videami. Na obrázku č.39 vidíme program VideoSQC po fáze segmentácie.



Obr. č. 39 – Segmentácia v programe VideoSQC

Segmentácia sa podarila na 100%, čo znamená, že rozoznala všetky značky. Nakoniec sme exportovali uložené polohy značiek do externého súboru. Tento súbor sme dali na vstup do animačného programu BlueBone. Na obrázku č. 40 vidíme snímky programu Bluebone vo fáze animácie.



Obr. č. 40 – Animácia v programe BlueBone

Z obrázku to nie je vidieť, ale Bluebone animoval obrázky presne podľa vstupného videa. Keďže sme dosiahli 100% úspešnosť segmentovania a animácia bola tiež úspešná, tak s výsledkom projektu sme veľmi spokojní.

6 Vysokorýchlostné kamery

V rámci tohoto projektu sme sa snažili zabezpečiť najvhodnejšie podmienky, aby sme bez problémov dokázali segmentovať nasnímané videozáznamy. Okrem samotnej úpravy scény, použitia reflexných značiek a vhodného osvetlenia scény sme sa snažili získať lepšie hardvérové vybavenie pre tento projekt. Sústredili sme sa najmä na vysokorýchlostné kamery, ktoré by nám umožnili nasnímať kvalitné videozáznamy bez rozmazaných značiek, pričom by bolo možné snímať aj rýchle pohyby. Bežné videokamery so vzorkovacou frekvenciou 25 fps neposkytujú dostatočne kvalitné snímky, pretože pri rýchlejších pohyboch snímaného objektu nestíhajú zachytávať tieto objekty a obraz je potom rozmazaný.

Preto sme kontaktovali firmu KVANT spol. s r.o., ktorá sa zaoberá laserovou a snímacou technikou. Túto spoločnosť zastupuje pán Ing. Martin Anetta. Keďže ceny vysokorýchlostných kamier sa pohybujú od 15 000 EUR, pán Ing. Martin Anetta nám poskytol informácie o možnosti zapožičania vysokorýchlostných kamier priamo od ich výrobcu.

Firma KVANT spol. s r.o. by nám za poplatok sprostredkovala zapožičanie hi-speed kamery, avšak tieto poplatky sú pre nás veľmi vysoké.

Cenová ponuka ohľadne zápožičky hi-speed kamery zahŕňa nasledovné položky:

- jednorázové prepravné a režijné náklady 300 EUR
- cena za požičanie 100 EUR / den
- školenie a asistenčné služby 20 EUR / hod

Keďže v rámci tímového projektu nemáme k dispozícii finančné prostriedky na zapožičanie, museli sme si vystačiť s bežnými 25 fps videokamerami.

Preto sme sa sústredili na použitie vhodných algoritmov, ktoré by nám umožnili segmentovať aj takéto rozmazané videozáznamy. Výsledky experimentovania s týmito algoritmi sú popísané v kapitole venovanej experimentom.

Hoci teraz nemáme k dispozícii finančné prostriedky pre zapožičanie hi-speed kamery, v nasledujúcej podkapitole uvádzame prehľad výrobcov takýchto kamier a ich webové stránky. Tieto uvádzame pre prípad, že by sa v budúcnosti našiel sponzor na ich zapožičanie. Na týchto stránkach je možné nájsť technické parametre jednotlivých hi-speed kamier.

Prehľad výrobcov

Firma KVANT spol. s.r.o. dokáže sprostredkovať zapožičanie hi-speed kamier od nasledovných výrobcov:

1.) **pco.imaging**

- ☐ Možnosť zapožičania na 5 dní.
- ☐ Technické parametre kamier je možné nájsť na adresách:
sensitive cameras: http://www.pco.de/pco/php/products/index_1-en_01030101.html
speed cameras: http://www.pco.de/pco/php/products/index_1-en_01030303.html

2.) **Hema electronic**

- ☐ Možnosť zapožičania na 5 dní.
- ☐ Technické parametre kamier je možné nájsť na adrese
http://www.hema.de/english/pdf/icam_e.pdf

3.) **Redlake**

- ☐ Technické parametre kamier je možné nájsť na adresách:
<http://www.redlake.com/>
<http://www.kamery.sk/typo3/kamery.sk/index.php?id=11>

Kontakt na firmu KVANT

Firmu KVANT spol. s.r.o. je možné nájsť na nasledovnej adrese:

KVANT spol.sr.o.
FMFI UK - Mlynská dolina
842 48 Bratislava, SLOVAKIA
telefón: +421-2-65411344
fax: +421-2-65411353
mobil: +421-905-276587
e-mail: Martin.Anetta@kvant.sk

7 Záver

Tento dokument je výsledkom práce nášho tímu počas zimného a letného semestra. Našou úlohou bolo analyzovať problematiku animácie pohybu dolných končatín človeka pri chôdzi z nasnímaných polôh kĺbov pomocou videokamery. Mali sme sa hlavne sústrediť na samotné snímanie pohybu a transformáciu súradníc kĺbov do animačného programu. Samotný projekt tvoria tri programy: Analyser, BlueBone a VideoSQC. Aplikácia analyser slúži na analýzu zosnímaných údajov v grafoch a poskytuje prostredie pre porovnávanie a skúmanie priebehov rotácií a posunov jednotlivých kĺbov. Aplikácia VideoSQC slúži na získanie polôh kĺbov z videosekvencií. Podrobný postup získania týchto polôh sme popísali v dokumentácii. Posledná aplikácia slúži na zobrazenie animácie pomocou dolnej časti ľudskej kostry. Najdôležitejším z týchto programov je program VideoSQC, ktorým sme sa my najviac zaoberali.

Tomuto projektu sa venovali pred nami až dva tímy. Z procesu snímania obrazu mali vyriešenú synchronizáciu viacerých kamier. Ich hlavným nedostatkom bol algoritmus segmentácie, ktorý dosahoval veľmi malú úspešnosť. Ďalším problémom bolo, že snímali pohyb len jednej nohy. Na snímanie pohybu oboch nôh je potrebné implementovať epipolárnu geometriu.

Ako sme už spomínali, našou úlohou bolo hlavne vylepšiť segmentáciu, čo je metóda určená na získanie polohy kĺbov. Výrazne sme vylepšili samotný algoritmus segmentácie aj predspracovania obrazu. Za určitých špecifických podmienok pre nasnímanie obrazu sme dosiahli 100% úspešnosť segmentácie. Ako sa ukázalo, hlavnými podmienkami úspešnosti segmentácie sú kvalitné rýchle kamery a veľmi dobré osvetlenie. My sme experimentovali s bežnými pomalými kamerami, ktoré sú postačujúce len na snímanie pomalých pohybov. Pri rýchlejších pohybov sú značky vo videosekvenciách rozmazané a algoritmus často zlyháva pri ich identifikácii, pretože rozmazaním sa výrazne zníži aj intenzita týchto bodov. Získané polohy kĺbov sme úspešne transformovali do animačného programu BlueBone.

Nepodarilo sa nám však úplne implementovať epipolárnu geometriu z dôvodu nedostatku času. Tento proces zostal vo fáze analýzy a implementácie jednoduchého prototypu. Kvôli tomuto nedostatku môžeme snímať a analyzovať pohyb len jednej nohy.

Naším hlavným cieľom bolo vylepšiť algoritmus segmentácie, čo sa nám aj podarilo. Pre nadväzujúcich riešiteľov odporúčame vylepšiť a dokončiť implementáciu epipolárnej geometrie, bez ktorej nie je možné použiť tento projekt v reálnom prostredí.