VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Rozpoznávání akcí

Action Recognition

2019 Tomáš Dedek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Tomáš Dedek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Rozpoznávání akcí
Action Recognition

Jazyk vypracování: čeština

Zásady pro vypracování:

Rozpoznávání akcí a činností je v poslední době věnována značná pozornost výzkumu. Úspěšné metody mohou být využity v mnoha aplikacích, jako je bezpečnost, analýza lidského chování apod. Je proto potřeba vyvíjet spolehlivé metody pracující v reálném čase.

1. Seznamte se se základními technikami detekce pohybu ve videosekvencích.
2. Seznamte se s metodami detekce a klasifikace akcí (SSM[1] a DMM[2]).
3. Tyto metody popište.
4. Experimentálně ověřte funkčnost algoritmů. Případně navrhněte a implementujte vylepšení.
5. Zjištěné poznatky řádně zdokumentujte v textu práce.

Seznam doporučené odborné literatury:

[1] JUNEJO, I N, E DEXTER, I LAPTEV a Patrick PEREZ. View-Independent Action Recognition from Temporal Self-Similarities. IEEE Transactions on Pattern Analysis and Machine Intelligence [online]. 2011, 33(1), 172-185 [cit. 2017-10-12]. DOI: 10.1109/TPAMI.2010.68. ISSN 0162-8828. Dostupné z: http://ieeexplore.ieee.org/document/5432213/
[2] CHEN, Chen, Kui LIU a Nasser KEHTARNAVAZ. Real-time human action recognition based on depth motion maps. Journal of Real-Time Image Processing [online]. 2016, 12(1), 155-163 [cit. 2017-10-12]. DOI: 10.1007/s11554-013-0370-1. ISSN 1861-8200. Dostupné z: http://link.springer.com/10.1007/s11554-013-0370-1
[3] Ronald Poppe: A survey on vision-based human action recognition, Image and Vision Computing, Volume 28, Issue 6, June 2010, Pages 976-990, ISSN 0262-8856

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radek Simkanič, DiS**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.	prof. Ing. Pavel Brandštetter, CSc.
vedoucí katedry	*děkan fakulty*

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2019 .....

Na tomto místě bych rád poděkoval mému vedoucímu Ing. Radku Simkaniči, DiS za věnovaný čas, konzultace a věcné připomínky, kterými přispěl k vypracovávání této bakalářské práce. Dále bych chtěl poděkovat všem, kteří mě podporovali po celou dobu studia.

Abstrakt

Tématem této bakalářské práce je rozpoznávání akcí. V této práci jsem se zaměřil na rozpoznávání a klasifikaci akcí pomocí dvou metod. Konkrétně na DMM a SSM. Tyto metody jsou v práci popsány a implementovány. K implementací byl použit programovací jazyk C++a knihovna pro zpracování obrazu OpenCV.

Klíčová slova: Rozpoznávání akcí, klasifikace akcí, dataset, OpenCV, C++, Kinect

Abstract

Theme of this bachelor's thesis is action recognition. In this thesis I focused on action recognition and classification by using two methods. Specifically on DMM and SSM. These methods are in this thesis described and implemented. The work is based on programming language C++and computer vision library OpenCV.

Key Words: Action recognition, action classification, dataset, OpenCV, C++, Kinect

Obsah

Seznam použitých zkratek a symbolů

SDK	–	Software Development Kit – sada vývojových nástrojů
CMOS	–	Complementary Metal-Oxide Semiconductor – technologie obrazových čipů
IR	–	Infrared – infračervený
RGB	–	Red Green Blue – barevný model
JPG, JPEG	–	Joint Photographic Experts Group – formát obrázků
XML	–	Extensible Markup Language – značkovací jazyk
BIN	–	Binary – binární
GMG	–	Geometric MultiGrid – metoda na segmentaci pozadí
KNN	–	K-Nearest Neighbors – metoda na segmentaci pozadí
MOG	–	Mixture Of Gaussian – metoda na segmentaci pozadí
MOG2	–	Mixture Of Gaussian 2 – metoda na segmentaci pozadí
ROI	–	Region Of Interest – oblast zájmu
SVM	–	Support Vector Machine – metoda strojového učení
HOG	–	Histogram of Oriented Gradients – histogram orientovaných gradientů
HMAX	–	Hierarchical MAX – metoda rozpoznávání objektů
2D	–	Dvoudimenzionální
3D	–	Trojdimenzionální
4D	–	Čtyřdimenzionální
LOP	–	Local Occupancy Patterns
ROP	–	Random Occupancy Pattern
STIP	–	Space-Time Interest Point
STOP	–	Space-Time Occupancy Pattern
DMM	–	Depth Motion Maps
LBP	–	Local Binary Pattern
SSM	–	Self-Similarity Matrix
RAM	–	Random-access memory – druh paměti

Seznam obrázků

Seznam tabulek

Seznam tabulek

1 Úvod

Ropoznávání akcí je v poslední době důležitým tématem počítačového vidění a je mu věnována velká část výzkumu. Uspěšně fungující metody mohou najít uplatnění ve velkém spektru odvětví, jako je robotika, analýza lidského chování, bezpečnost, interakce mezi strojem a člověkem a dalších. Současné metody často využívají zjednodušené scénáře s jednoduchým pozadím, několika jednoduchými akcemi a statickými kamerami. Proto je potřeba vyvíjet spolehlivé a v reálném čase pracující metody.

Tato bakalářská práce je rozdělena do několika částí. V první části jsou popsány základní pojmy, zejména knihovna OpenCV, která patří k nejpoužívanějším knihovnám pro zpracování obrazu, hloubkové mapy, snímače pohybu (Microsoft Kinect, ASUS Xtion 2, Intel RealSense D435 a Orbbec Astra) a datasety (UTKinect-Action3D Dataset, MSR Action3D Dataset).

Následující část se zabývá nejběžnějšími možnostmi snímání a zpracování obrazu pro účely počítačového vidění, metodami pro zvýšení efektivity algoritmů pro rozpoznávání akcí (odečítání pozadí, prahování, detekce hran a specifikace oblasti zájmu). Cílem je rozpoznat a správně klasifikovat akce prováděné člověkem. K tomuto účelu jsou určeny metody a algoritmy založené na modelu lidského těla, obrazových modelech nebo prostorové statistice.

Poslední část je zaměřena na popis a testování vybraných metod, a to Depth Motion Maps a Temporal Self-Similarities. Cílem je implementace, ověření funkčnosti a srovnání těchto dvou metod rozpoznávání akcí.

2 Základní informace a pojmy

2.1 Programovací jazyk C++

C++ je multiparadigmatický programovací jazyk, který vyvinul Bjarne Stroustrup a další v Bellových laboratořích AT&T rozšířením jazyka C. C++ podporuje několik programovacích stylů neboli paradigmat jako je procedurální programování, objektově orientované programování a generické programování, není tedy jazykem čistě objektovým.[1] V současné době patří podle TIOBE Indexu C++ mezi čtvrtý nejrozšířenější programovací jazyk. Nejrozšířenějším programovacím jazykem je pak Java následována jazyky C, Python a pátým nejrozšiřenějším programovácím jazykem je Visual Basic .NET.[2]

2.2 Knihovna OpenCV

OpenCV (Open Source Computer Vision Library) je open source knihovna pro počítačové vidění a strojové učení. Tato knihovna byla vytvořena tak, aby poskytovala společnou infrastrukturu pro aplikace počítačového vidění a urychlila tak využití strojového vnímání v komerečních produktech. Tato knihovna obsahuje přes 2500 optimalizovaných algoritmů určených pro strojové učení. Tyto algoritmy mohou být použity například pro detekci a rozpoznání obličejů, identifikaci objektů, klasifikaci lidských akcí ve videosekvencích a další. OpenCV poskytuje rozhraní pro C++, Python, Java, MATLAB a funguje na různých platformách, jako jsou Windows, Linux, Android a Mac OS.[3]

2.3 Hloubková mapa

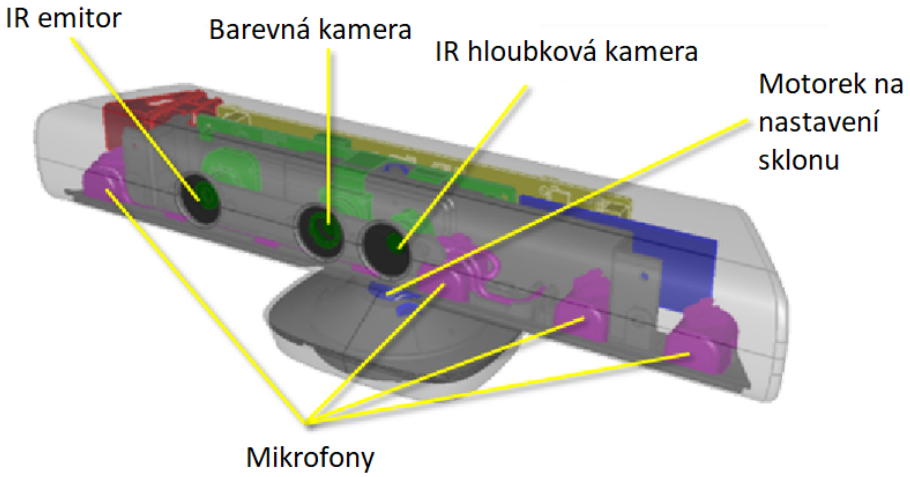
Hloubková mapa znamená v 3D grafice snímek, který obsahuje informace o vzdálenosti objektu od senzoru.[4] U často používaného Kinectu je rozsah snímané vzdálenosti 80 až 400cm ve výchozím režimu a 40 až 300cm v near mode.[5] Microsoft Kinect v2 se přestal v roce 2017 prodávat.[6] V tabulce 1 jsou další zařízení pro snímání pohybu (ASUS Xtion 2, Intel RealSense D435, Orbbec Astra) a jejich srovnání.

	Kinect v2	ASUS Xtion 2	Intel RealSense D435	Orbbec Astra
Cena	$160	$270	$180	$160
Hloubkové zorné pole (vertikálně)	60°	52°	65.5° (pouze 42° pro RGB)	45°
Rozlišení (hloubka)	512 × 424	640 × 480 (použitelné 320 × 240)	1280 × 720 (ale nepřesné s mnoha artefakty)	640 × 480 (použitelné 320 × 240)
FPS (hloubka)	30	30	až 90	30
Možnost více senzorů na 1 PC	Ne	Ne	Ano	Ano

Tabulka 1: Porovnání zařízení pro snímání pohybu.[11]

2.4 Microsoft Kinect

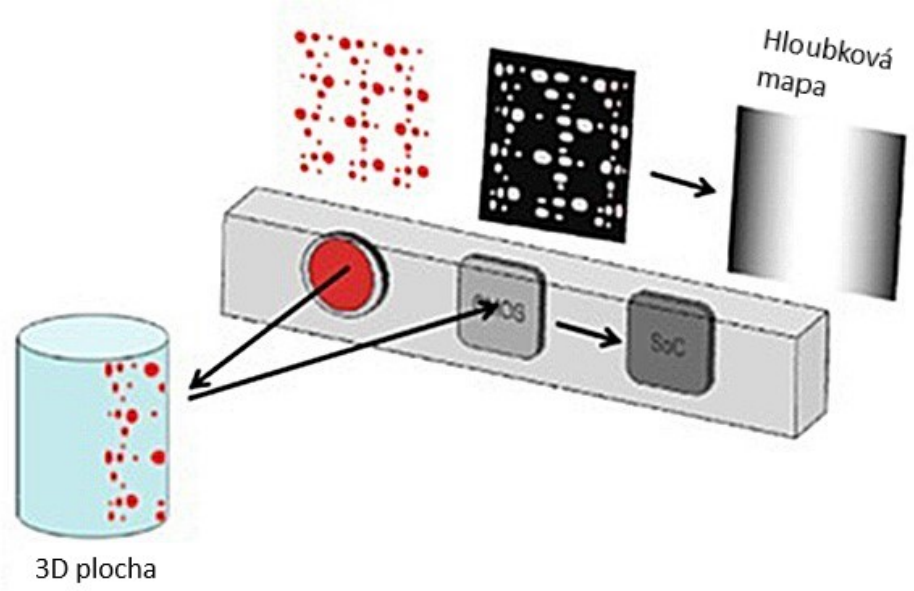
Kinect je zařízení pro snímání pohybu vyrobený Microsoftem pro Xbox360, Xbox One a počítače s operačním systémem Windows. Toto zařízení se stalo oblíbeným, jelikož umožnilo uživateli ovládat herní konzoli nebo počítač pomocí gest, pohybů nebo hlasových příkazů. První generace byla představena v roce 2010 jako herní zařízení pro konzoli Xbox 360. V roce 2011 vydala společnost Microsoft předběžnou verzi SDK pro vývoj aplikací v operačním systému Windows 7, stále ovšem nebylo možné Kinect připojit přímo k počítači. Toto se změnilo s příchodem Kinectu pro Windows v roce 2012. Kinect v2 pro Windows byl představen v roce 2014 a přinesl řadu vylepšení. V roce 2017 došlo k ukončení výroby Kinectu. Popis zařízení Kinect je na obrázku 1.[6]



Obrázek 1: Popis částí Kinectu.[8]

Barevná CMOS kamera se už stává standardem v mnoha zařízeníchch, jako jsou mobilní telefony, tablety a podobně. Kamera v Kinectu neslouží přímo pro výpočet vzdáleností a co se obrazové kvality týče, je spíše průměrné až podprůměrné kvality. Kinect v1 umí snímat v rozlišení 640×480 pixelů při 30 fps.[7][9] Kinect v2 již umí snímat v rozlišení 1920×1080 pixelů při 30 fps.[7]

IR podsystém je klíčový pro vznik hloubkového obrazu. Skládá se z IR emitoru, který promítá na objekt mračno bodů, které je pro lidské oko neviditelné a monochromatické CMOS kamery citlivé jen na IR pásmo, která toto mračno bodů detekuje viz obrázek 2. Na bližších objektech je mračno bodů hustší a na vzdálenějších objekty se mračno bodů stává řidší. Tato kamera u Kinectu v1 umí snímat v rozlišení 320×240 pixelů, zorné pole kamery je 57° horizontálně a 43° vertikálně.[7][9] Kinect v2 opět umí snímat ve vyšším rozlišení, a to 512×424 pixelů, zorné pole je tentokrát 70° horizontálně a 60° vertikálně. Po sejmutí tohoto mračna bodů dokáže Kinect určit vzdálenost objektu od kamery.[7]



Obrázek 2: Vytvoření hloubkové mapy pomocí Kinectu.[10]

2.5 Dataset

V této práci se používají jako vstup do vybraných metod pro rozpoznávání akcí dva datasety, a to UTKinect-Action3D dataset [12] a MSR Action3D dataset [14].

UTKinect-Action3D dataset byl zaznamenán zařízením Microsoft Kinect (viz kapitola 2.4) s Kinect pro Windows SDK Beta. V tomto datasetu se nachází 10 typů akcí: chůze, sednutí, postavení, zvednutí předmětu ze země, nesení, hození, tlačení, tažení, mávání, tleskání. Tyto akce vykonává 10 subjektů a každý subjekt danou akci vykonává dvakrát (toto opakování se označuje také jako epizoda). Tento dataset se skládá ze čtyř částí:

- RGB snímků ve formátu jpg s rozlišením 480×640.
- Hloubkových map s rozlišením 320×240 viz obrázek 3. Tyto mapy byly uloženy do XML souborů pomocí knihovny OpenCV.
- Souborů s pozicemi jednotlivých kloubů. Každý řádek obsahuje data jednoho snímku, kde první číslo je číslo snímku a následující čísla jsou hodnoty umístění kloubů na souřdnicích x, y, z.
- Popisů jednotlivých sekvencí v textovém souboru.[12]

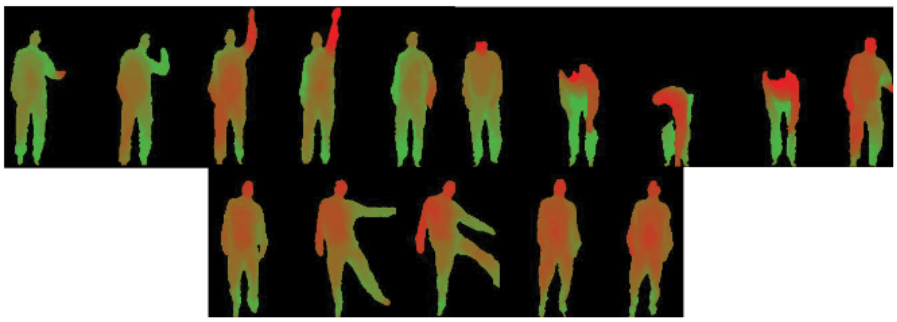
Na tomto datasetu se provádí testování dvěma různými způsoby. První způsob využívá k trénování první epizodu a k testování druhou epizodu. Druhý způsob využívá na trénování první polovinu subjektů a na testování druhou polovinu subjektů.[64]

Obrázek 3: Příklad snímků z UTKinect-Action3D Dataset.[13]

MSR Action3D dataset obsahuje 20 typů akcí. Tyto akce vykonává 10 subjektů a každý subjekt danou akci vykonává dvakrát nebo třikrát (toto opakování se označuje také jako epizoda). Rozlišení snímků je dle webových stránek 640×240, ovšem ve skutečnosti rozlišení činí 320×240. Tento dataset byl zaznamenán zařízením podobným Microsoft Kinect a skládá se ze tří částí:

- Hloubkových map v souborech bin viz obrázek 4.
- Souborů s pozicemi jednotlivých kloubů. Každý kloub má dvě hodnoty udávající pozici vůči senzoru, hodnotu hloubky a hodnotu spolehlivosti.
- Souborů odpovidajícím souborům z předchozího bodu. Odlišností jsou hodnoty udávající pozice jednotlivých kloubů v realném světě.[14]

Na tomto datasetu se provádí testování třemi různými způsoby. První způsob využívá k trénování první epizodu a k testování druhou a třetí epizodu. Druhý způsob využívá k trénování druhou a třetí epizodu a k testování první epizodu. Poslední způsob využívá na trénování první polovinu subjektů a na testování druhou polovinu subjektů.[64]



Obrázek 4: Příklad snímků z MSR Action3D Dataset.[15]

Oba datasety využívají křížového ověření. Křížové ověření neboli testování mimo vzorek je jeden z několika postupů odhadu, jak dobře bude model (klasifikátor), který je natrénovaný pomocí dat z trénovací množiny, pracovat na datech z testovací množiny. Je několik metod křížové validace.[16]

Jednou z nich je **k-násobná křížová validace** (k-fold cross-validation). Jejím cílem je náhodně rozdělit originální vzorek do k stejně velkých dílčích vzorků. Z těchto vzorků se jeden vzorek zachová jako validační vzorek pro testování modelu a zbylých $k - 1$ vzorků je použito jako trénovací data. Proces křížové validace je potom opakován k-krát, kde je každý z k subsetů použit právě jednou pro testování. Výhodou této metody oproti opakovanému náhodnému podvzorkování (random sub-sampling) je ta, že všechny data jsou použity jak pro trénování, tak pro testování. V praxi se většinou využívá 10násobná křížová validace, tedy k=10, ale většinou zůstává k jako proměnný parametr.[16]

Další metodou je **holdout metoda**. V této metodě jsou datové body náhodně přiřazené ke dvěma sadám, obvykle nazývaným tréninkový soubor a testovací soubor. Velikost každé sady je libovolná, i když obvykle je testovací sada menší než tréninková sada. V typickém křížovém ověřování jsou výsledky vícenásobných testů modelu zprůměrovány dohromady, ovšem metoda holdout, zahrnuje jeden průběh. Měla by tedy být používána s opatrností, protože bez takového zprůměrování více cyklů lze dosáhnout vysoce zavádějících výsledků.[16]

Jednou z dalších metod je **opakované ověření náhodného dílčího vzorkování** (repeated random sub-sampling validation). Tato metoda, také označována jako Monte Carlo cross validation, náhodně rozděluje datovou sadu do tréninkových a validačních dat. Pro každé takové rozdělení je model přizpůsoben tréninkovým datům a pomocí validačních dat je vyhodnocena prediktivní přesnost. Výhodou této metody (oproti k-násobné křížové validaci) je, že poměr mezi tréninkovými a validačními sety není závislý na počtu iterací. Nevýhodou této metody je, že některá data nemusí být vybrána vůbec, zatímco jiné mohou být vybrány vícekrát.[16]

3 Počítačové vidění

Počítačové vidění představuje jednu z disciplín počítačové vědy a vývoje softwaru, která řeší problematiku pochopení vícerozměrných dat. Jedná se tedy o technologii, která se snaží napodobit lidské vnímání okolí. Součástí tohoto oboru je lokalizace, rozpoznávání a klasifikace akcí. K tomu, aby bylo možné rozpoznávat akce, je třeba nejdřív detekovat pohyb ve videosekvencích. Rozvoj oboru je přímo spojen s rozvojem počítačových zařízení, jejichž výpočetní výkon se využívá v maximální možné míře.[17] Každý systém počítačového vidění obsahuje znalosti z různých oblastí:

- Fyzikální principy – geometrie, světlo
- Hardware – pořízení dat, diskretizace, digitalizace
- Algoritmy – zpracování signálu
- Interpretace výsledků
- Velké množství kombinací pro výběr řešení [17]

Pro úpravu obrazu se mimo jiné používají morfologické operace. Nečastěji se aplikují na binární obrazy, ale lze je snadno zobecnit i na šedotónové a barevné obrazy. Morfologické operace se používají pro předzpracování (odstranění šumu, zjednodušení tvaru objektů) a zdůraznění struktury objektů (kostra, ztenčování, zesilování, konvexní obal, označování objektů). Jsou potřeba dva vstupy. Jeden je originální obrázek a druhý se nazývá strukturační prvek nebo jádro (obdélníkové, eliptické a křížové). Mezi dvě základní morfologické operace patří dilatace a eroze. Ukázka po aplikování těchto dvou morfologických operací je na obrázku 5. Dalšími variantami jsou pak otevírání, zavírání a přechod.

Dilatace [69] skládá body dvou množin pomocí vektorového součtu. Objekty v obrazu jsou po aplikaci dilatace zvětšené o jednu "slupku"na úkor pozadí. Dilatace se používá k zaplnění děr, případně zálivů.

Druhou základní morfologickou operací je **eroze** [69]. Eroze skládá dvě bodové množiny s využitím vektorového rozdílu. Je duální transformací k dilataci a používá se pro zjednodušení struktury objektů. Objekty jednotkové tloušťky zmizí a složité objekty spojené čárami jednotkové tloušťky se rozloží na několik jednodušších objektů.[71]

(a) (b) (c)

Obrázek 5: Ukázka dilatace a eroze. Na obrázku (a) je originální obrázek. Na obrázku (b) je obrázek po aplikování dilatace a na obrázku (c) je obrázek po aplikování eroze.[69]

3.1 Detekce pohybu

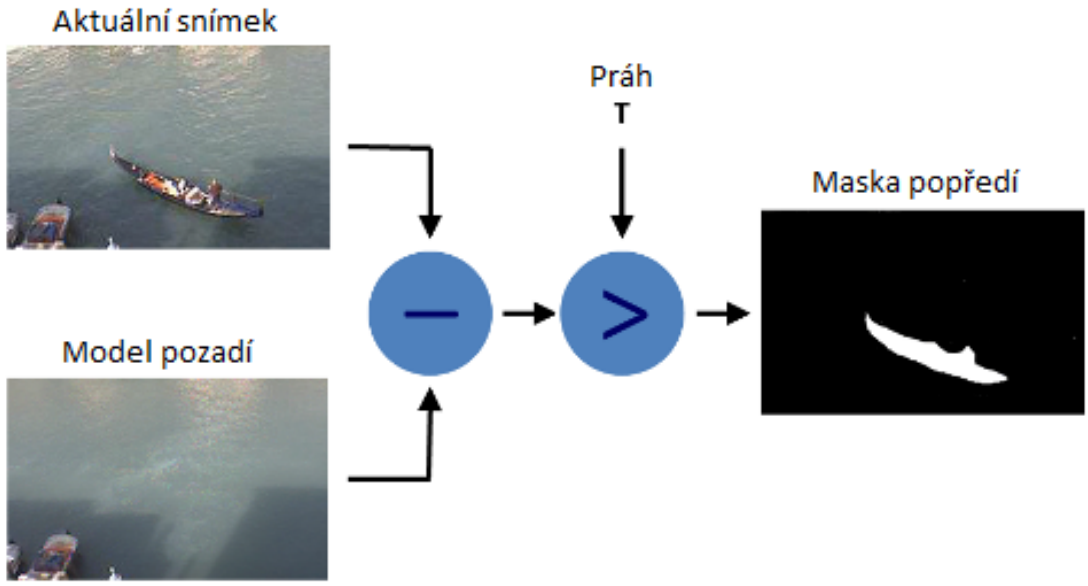
V rámci zvýšení efektivnosti je vhodné ještě před použitím algoritmů na rozpoznávání akcí využít některý z algoritmů na detekování pohybu v sekvencích. K detekci pohybů ve videosekvencích se používají následující metody:

- Odečítání pozadí
- Prahování
- Detekce hran
- Specifikace oblasti zájmu

3.1.1 Odečítání pozadí

Jedná se o jednu z nejčastěji používaných metod pro detekci pohyblivých objektů ve videosekvenci. Cílem je detekce pohybujících se objektů z rozdílu mezi aktuálním snímkem a referenčním snímkem, který se často nazývá jako obrázek pozadí nebo model pozadí. Jako model pozadí se používa statický snímek bez pohyblivých předmětů. Příklad odečtu pozadí je na obrázku 6. Existuje mnoho algoritmů, kterými lze docílit detekce pohybu pomocí odečítání pozadí. Nejvíce využáváné algoritmy jsou založeny na modelech Gaussovských směsí (Gaussian mixture models) [68], mezi ně patří GMG, KNN, MOG, MOG2.[19] Jednotlivé algoritmy se pak liší rychlostí, požadavky na paměť a přesností.[18]

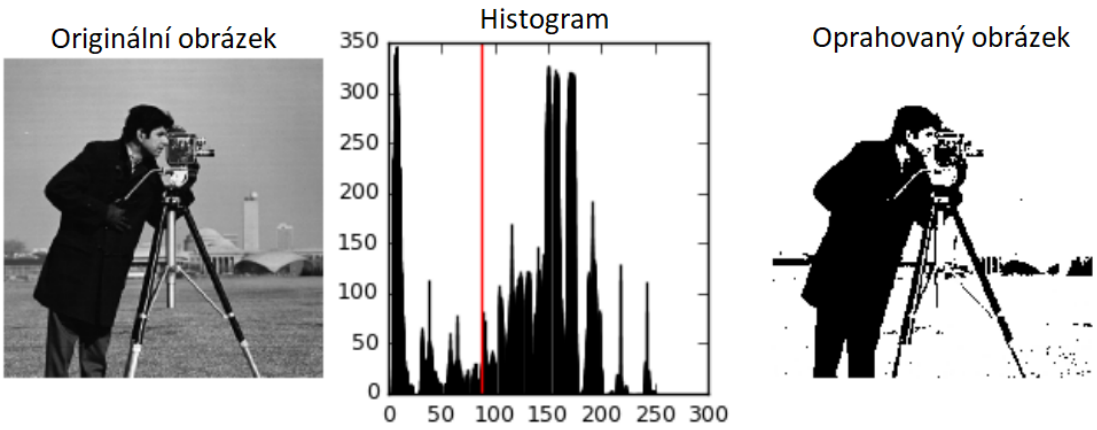
MOG2 je algoritmus na segmentaci pozadí nebo popředí založený na modelech Gaussovských směsí [68]. Tato metoda je vylepšená metoda MOG, která je založena na modelování každého pixelu z obrazu na směs K Gaussových rozdělení ($K = 3$ až 5). Dále jsou v metodě MOG váhy, které reprezentují čas a značí, jak dlouho zůstávají v obraze pixely bez změny. Tudíž se jako pozadí jeví ty pixely, které jsou ve scéně déle a jsou statické. Jednou z hlavních vlastností MOG2 algoritmu je to, že vybere vhodný počet gaussovského rozdělení pro každý pixel. Díky tomu se lépe přizpůsobuje scénám, které se mění v důsledku změn osvětlení.[67]



Obrázek 6: Příklad funkce algoritmu odečtu pozadí.[20]

3.1.2 Prahování

Jedná se o jednu z metod segmentace obrazu. Jejím principem je nalezení takové hodnoty (prahu) v histogramu pro kterou bude platit, že všechny hodnoty jasu nižší než práh odpovídají pozadí, zatímco všechny hodnoty vyšší než práh odpovídají popředí viz obrázek 7.[21] Histogram je grafické znázornění distribuce dat pomocí sloupcového grafu se sloupci stejné šířky, vyjadřující šířku intervalů (tříd), přičemž výška sloupců vyjadřuje četnost sledované veličiny v daném intervalu.[56]

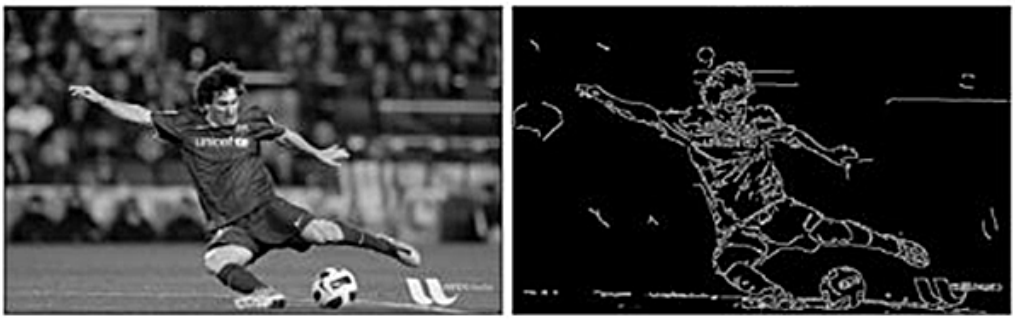


Obrázek 7: Příklad prahování. Na prvním obrázku je originální obrázek. Na druhém je umístěn práh v histogramu a na posledním obrázku je odstranění pozadí na základě prahu.[22]

3.1.3 Detekce hran

Detekce hran patří spolu s redukcí šumu, jasovým a geometrickým transformacím k základním operacím s obrazem. Hrana je místo v obrazu, které vykazuje vysokou prostorovou frekvenci tj. skokovou změnu jasové funkce. Samotné detektory hran jsou sestaveny tak, aby byly citlivé na libovolnou skokovou změnu jasové funkce obrazu, tedy i na změny způsobené šumem. Aby se snížil počet falešně detekovaných hran je vhodné obraz před samotnou hranovou filtrací vhodně upravit, například rozmazáním.[23] Jeden z nejpoužívanějších postupů z této kategorie popsal John Canny v roce 1986 [70].

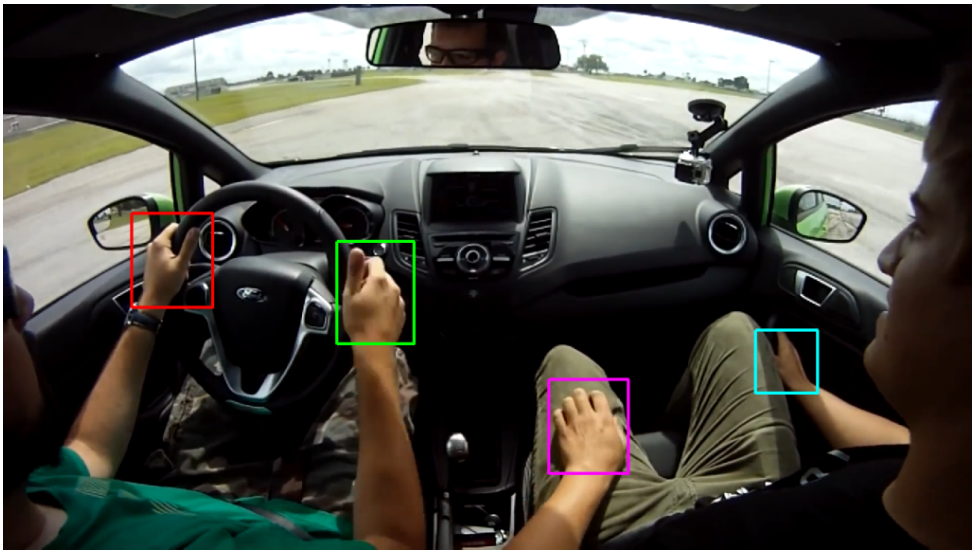
Cannyho detektor hran je populární algoritmus detekce hran. Byl vyvinut John F. Cannym v roce 1986 [70]. Jedná se o vícestupňový algoritmus. prvním krokem je odstranění šumu v obraze pomocí Gaussova filtru, protože je detekce hran citlivá na šum v obraze. V druhém kroku je rozmazaný obraz filtrován s aplikovaným jádrem Sobel v horizontálním i vertikálním směru. Po získání velikosti a směru gradientu se provádí úplné skenování obrazu, aby se odstranily nežádoucí pixely, které nemusí představovat hranu. Poslední fází je Hysterezní prahování. Tato fáze rozhoduje, zda všechny hrany jsou skutečně hrany nebo ne. K tomu jsou potřeba dvě prahové hodnoty, dolní práh a horní práh. Všechny hrany s gradientem intenzity větším než horní práh jsou hrany a ty, které se nachází pod dolním prahem hrany nejsou, takže jsou vyřazeny. Hrany, které leží mezi těmito dvěma prahovými hodnotami, jsou klasifikovanými na základě jejich konektivity. Pokud jsou připojeny k hranovým pixelům, považují se za součást hran. V opačném případě jsou také vyřazeny. Ukázka aplikace Cannyho detektoru je na obrázku 8.[24]



Obrázek 8: Ukázka aplikace Cannyho detektoru hran.[25]

3.1.4 Specifikace oblasti zájmu

Tato technika, která je známá pod zkratkou ROI, slouží pro zúžení oblasti pro zpracování v obraze na menší a konkrétnější oblast. V počítačové grafice se používá k identifikaci objektů v obraze, které mají určitý význam pro daný úkol viz obrázek 9. Často se využívá pro zpracování oblastí zájmu ve větší kvalitě, než jiné oblasti. ROI může být definováno prakticky jakýmkoliv geometrickým tvarem.[26]



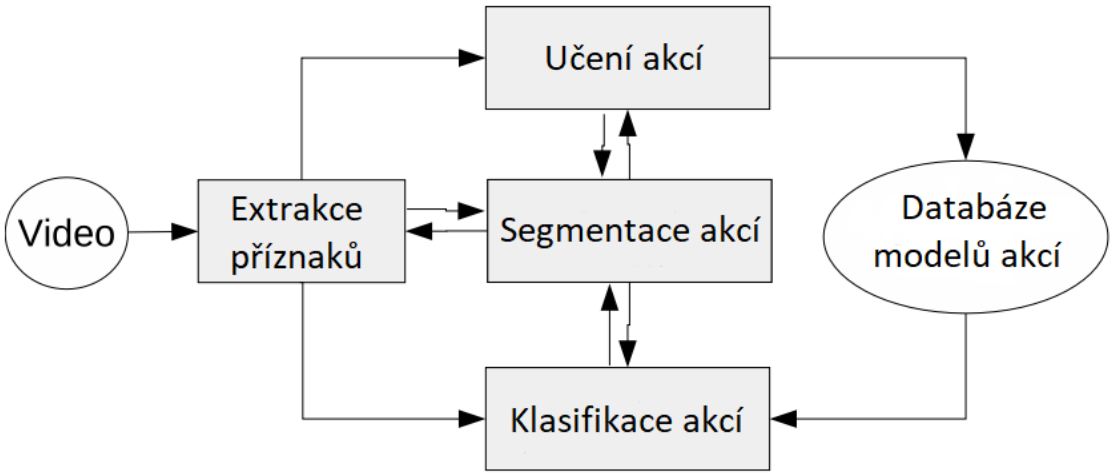
Obrázek 9: Příklad specifikace oblasti zájmu. V tomto případě je potřeba detekovat pohyb dlaní. Pro potřebu detailního pohledu na dlaně je vhodné využití ROI. [27]

4 Metody rozpoznávání akcí

Rozpoznávání akcí se během posledních let stalo velmi důležitým tématem v počítačovém vidění a dostalo se mu tak velkého využití v oblasti robotiky, interakci člověka se strojem, či analýzy lidského chování. Účelem rozpoznávání akcí je identifikovat a pojmenovat akce prováděné člověkem pomocí různých algoritmů.

Při vývoji algoritmů a metod rozpoznávání akcí se řeší řada problémů, jelikož by měly být v ideálním případě univerzálně použitelné pro jakoukoliv osobu a v jakémkoliv místě. Tyto metody musí být tedy nezávislé na různých rysech osob (postavě, způsobu provedení akce apod.), případně na umístění kamery vůči osobě.

Systém rozpoznávání akcí probíhá v několika krocích zobrazených v obrázku 10. Jako první dochází k extrakci příznaků z videosekvence. Tyto extrahované příznaky jsou dále využity buď k učení, klasifikaci nebo segmentaci akcí.



Obrázek 10: Schéma datového toku pro generický systém rozpoznávání akcí.[42]

4.1 Extrakce příznaků

Příznakem lze nazvat cokoliv, co má ve videosekvenci nějakou užitečnou informaci pro rozpoznání nebo klasifikaci. Může se jednat třeba o určitý tvar nebo také siluetu člověka. Hlavním úkolem extrakce příznaků je získávání pozic a pohybových podnětů, které nesouvisí s lidskými činnostmi. Výsledkem je tedy redukce velkého množství informací obsažených ve videosekvencích do kompaktnější snadno zpracovatelné formy, která stále dostatečně popisuje původní soubor dat.[42][43]

4.2 Učení a klasifikace akcí

Hlavními kroky učení a klasifakce akcí je učení vzorových akcí z extrahovaných příznaků a pomocí těchto vzorů dochází ke klasifikaci právě probíhající akce. Kategorie akcí, jako je mávání, chození a další, se mohou jevit jako jasně definované. Ve skutečnosti musí být variabilní, jelikož každá osoba může mít jinou postavu, jiné oblečení, danou akci může provést jinak rychle i samotné provedení akce může být odlišné.[42]

Pro učení se využívájí tří základní techniky, a to učení s učitelem, učení bez učitele a kombinovaného učení. **učení s učitelem** je jednou z nejrozšířenějších technik učení v oblasti rozpoznávání akcí. Na začátku systém příjmá jak vstupní tak i výstupní data. Jeho úkolem je pak vytvořit odpovidající pravidla, které mapují vstup na výstup. Trénink by měl probíhat tak dlouho dokud nebude výkonnost dostatečně dobrá. Systém je naučen na předem definovaných akcích, které jsou poté porovnávány s právě probíhající akcí. Ve většině případů je tato technika velmi rychlá a přesná.[44]

V případě **učení bez učitele** není žádný předem připravený vzor. Jde tedy o to naučit stroj rozpoznávat akce, aniž by měl předem připravený vzor akce. Cílem je odhalit skryté struktury, jako například najít skupiny snímků s podobnými osobami. Toto je ovšem v praxi obtížně realizovatelné.[44]

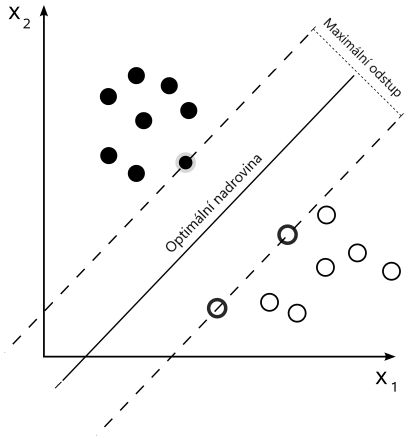
Kombinované učení je kombinací obou předchozích technik, přičemž je malé množství dat s výcvikovými vzory a velké množství dat bez předem připravených vzorů. Označená data slouží jednotlivým algoritmům jako tzv. násada, podle níž je natrénován počáteční klasifikátor, který se následně snaží označit zbylá neoznačená data zcela samostatně.[66]

4.2.1 Algoritmy strojového učení

Jedním z nejmodernějších algoritmů strojového učení je **hluboké učení** (deep learning). Hluboké učení je specifický přístup pro budování a trénování neuronových sítí. Neuronové sítě jsou napodobením lidského mozku. Algoritmus je považován za hluboký, jestliže vstupní data než se dostanou do výstupu, procházejí řadou nelinearit. Naproti tomu většina moderních algoritmů strojového učení je považována za "mělkou", protože vstup může probíhat pouze v několika úrovních volání podprogramu. Hluboké učení se spoléhá na jakýkoliv tréninkový proces, který objeví užitečné vzory ve vstupních příkladech. Hluboké učení je ideální pro práci s velkými daty a rozpoznávání hlasu.[39]

Mezi nejpoužívanější metodu strojového učení patří **metoda podpůrných vektorů** (support vector machine – SVM) [40], což je metoda strojového učení s učitelem, sloužící zejména pro klasifikaci a také pro regresní analýzu. Základem metody SVM je lineární klasifikátor. Cílem je nalézt nadrovinu, která prostor příznaků optimálně rozdělí na podprostory. Optimální nadrovina je taková, že hodnota minima vzdáleností bodů od roviny je co největší. To znamená, že okolo nadroviny je na obě strany co nejširší pruh bez bodů. Česky je tento pruh někdy nazýván jako pásmo necitlivosti nebo hraniční pásmo. Na popis nadroviny stačí pouze body ležící na okraji

tohoto pásma a těch je obvykle málo. Tyto body se nazývají podpůrné vektory. Ukázka je na obrázku 11. SVM jsou dobré při rozpoznávání obličejů a rukopisů.[40]



Obrázek 11: Rozdělující nadrovina a hraniční pásmo pro lineární SVM.[40]

Dalším příkladem jsou **pravděpodobnostní modely**, které se obvykle pokouší předvídat nejlepší odpověď vytvořením modelu s distribucí pravděpodobnosti. Jedná z výhod tohoto modelu spočívá v tom, že se vrací jak předpověď, tak stupeň jistoty. Pravděpodobnostní model je určen k distribuci možných výsledků. Může popisovat všechny předpokládané výsledky a předpovědět pravděpodobnost každého z nich. Často se používá k poskytnutí „relevance“ k výsledkům vyhledávače.[41]

Jako další lze uvést **techniky pro kombinaci více modelů** (ensemble learning). Tyto algoritmy slouží ke kombinování výstupů z různých Predictive Analytics modelů a vytvoření jediného výstupu. Jsou navrženy tak, aby pomohly při učení jiných programů pro strojové učení. Mezi první efektivní Ensemble Learning algoritmy patří Bootstrap agregační algoritmy. Bootstrap agregace je algoritmus pro strojové učení vytvořený pro podporu přesnosti a stability programů používaných v regresi a statistické klasifikaci.[41]

4.3 Segmentace akcí

Cílem segmentace akcí je rozdělit videosekvenci na jednotlivé části, a tím zvýšit přesnost rozpoznávání akcí. Většina metod sloužících k rozpoznávání akcí z hloubkových map se zaměřuje na použití ručně segmentovaných videoklipů. Tyto videoklipy obvykle obsahují pouze jednu úplnou akci. To sice zaručí docela uspokjivou přesnost při učení a testování na těchto videoklipech, ale v reálném čase je tento způsob téměř neudržitelný.[42]

4.4 Prostorová reprezentace akcí

Prvním krokem v reprezentaci akcí je extrakce příznaků, které rozlišují hlavně držení a pohyb těla. Existuje však více způsobů jak reprezentovat lidské tělo. Proto je lze rozdělit na následující

metody: metody založené na modelu lidského těla, metody založené na obrazových modelech a metody využívající prostorové statistiky.[42]

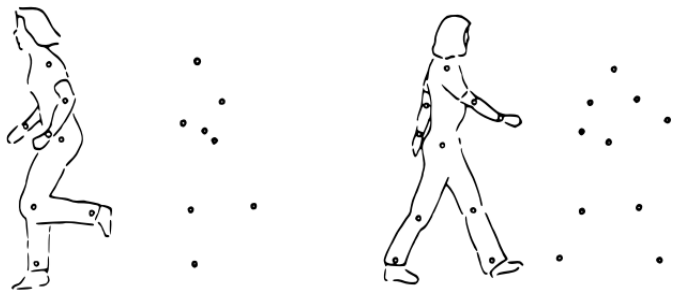
4.4.1 Metody založené na modelu lidského těla

Tyto metody reprezentují prostorovou strukturu akcí vůči lidskému tělu. V každém snímku videosekvence se získává za pomoci různých příznaků pozice lidského těla. Na základě těchto modelů lidského těla dochází poté k samotnému rozpoznávání akcí. Jedná se tedy o biologicky příjatelný přístup rozpoznávání akcí, který je podpořen psychofyzikální prací na vizuální interpretaci pohybu. Vědec Gunnar Johansson zjistil, že člověk je schopen rozeznat lidskou akci pouze z několika málo bodů na lidském těle. G. Johansson tento experiment demonstroval na osobě, která chodí v temné místnosti a má na hlavních kloubech přidělané světelné body.[28] Ukázka tohoto experimentu je na obrázku 12. Obrázek 13 pak znázorňuje rozmístění bodů u Microsoft Kinect. Tento experiment vedl k diskuzím o tom, zda lidé rozpoznávájí akce z 2D vzorů pohybu, nebo zda si nejprve vypočtou 3D rekonstrukci vzorů pohybu a až poté rozpoznají akci. Toto nakonec vedlo k tomu, že se strojové vidění rozdělilo na tyto dvě třídy: [42]

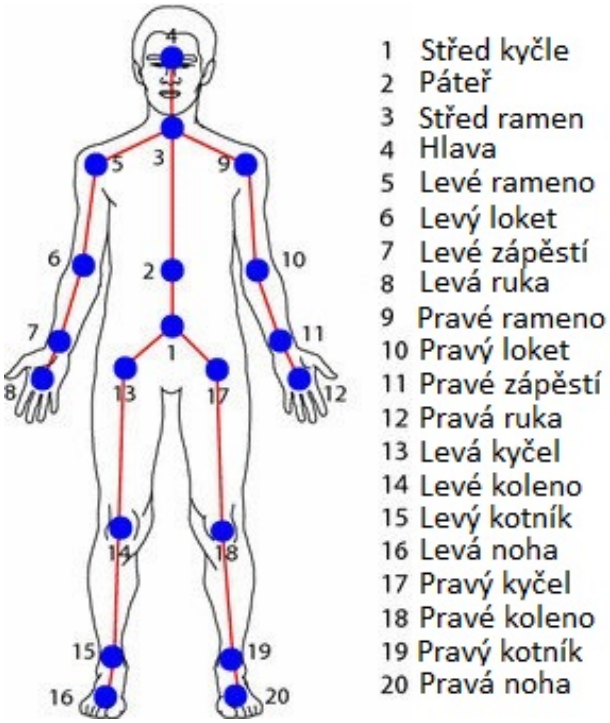
- rozpoznávání pomocí rekonstrukcí 3D modelů lidského těla
- přímé rozpoznávání z 2D modelů těla člověka

Rozpoznávání pomocí rekonstrukcí 3D modelů lidského těla se skládá ze dvou kroků. První krok se zabývá vytvořením 3D modelu lidského těla, který je reprezentovaný jako model složený z kloubů. Druhý krok se pak zabývá samotným rozpoznáváním akcí založeným na trajektoriích pohybů kloubů. Tento přístup má ovšem dva problémy. Těmi jsou velký počet pohybů, kterými lze provést danou akci a vysoká variabilita tvarů pohybů.

Přímé rozpoznávání nevyžaduje vytvoření 3D modelu, a tak jsou akce rozpoznávány přímo z 2D modelu těla člověka. Výsledkem jsou tak ploché postavy a 2D anatomické orientační body podobné Johanssonovým světelným bodům. Další přímé rozpoznávací přístupy používají prosté 2D reprezentace těla založené na sledovaných částech daného objektu, jako je např. trajektorie rukou a hlavy.[42]



Obrázek 12: Johanssonova ukázka jak lidé mohou rozpoznat akce pouze z pohybu několika světelných bodů umístěných na hlavních kloubech.[42]



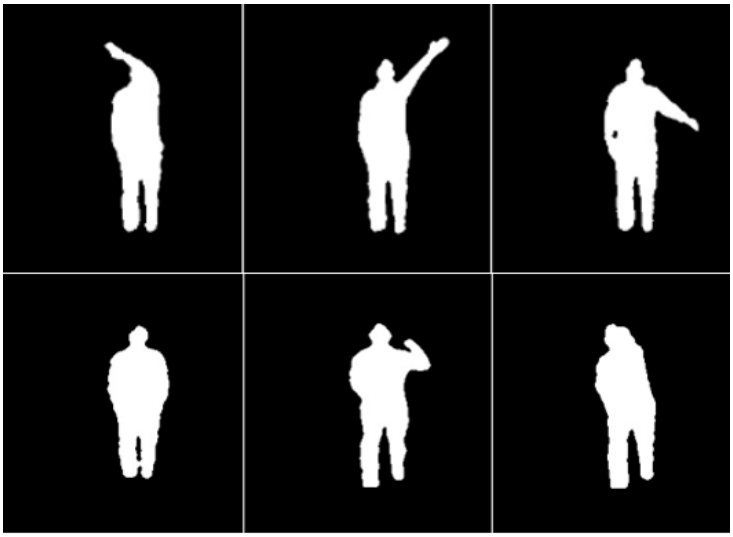
Obrázek 13: Rozmístění sledovaných bodů použitím Microsoft Kinect.[31]

4.4.2 Metody založené na obrazových modelech

Metodám rozpoznávání akcí využívající obrazové modely se někdy také říká holistické reprezentace [42]. Na rozdíl od předchozích metod vužívajících modely lidského těla nevyžadují detekci a označování jednotlivých částí těla. Stačí detekovat pouze oblast zájmu (viz kapitola 3.1.4) okolo sledované osoby, přičemž ve většině případů jsou příznaky vypočteny na základě tohoto ohraničení detekované oblasti. Modely obrazů proto mohou být mnohem jednodušší než modely založené na těle člověka, důsledkem toho mohou být prováděny mnohem efektivněji a robustněji.

Paradaxoně se však ukázalo, že metody vužívající modelu lidského těla jsou schopny dosahovat podobných výsledků jako metody využívající obrazové modely.[42]

Jak bylo zmíněno výše, jednou z důležitých tříd při použití těchto metod je extrakce oblasti zájmu nebo siluety těla. Nejjednodušším případem je použití sekvence, kde se nachází za snímaným člověkem statické pozadí viz obrázek 14. V ostatních případech je potřeba přistoupit k extrakci pozadí, což může zejména u členitých pozadí být problémem. Takto extrahované snímky lze již použít jako vstup pro rozpoznávání akcí. Výhodou je necitlivost na barvy, textury a změny kontrastu. Na druhou stranu úspěšnost rozpoznávání akcí ve velké míře závisí na kvalitním a robustním odstranění pozadí ze snímku.[42]

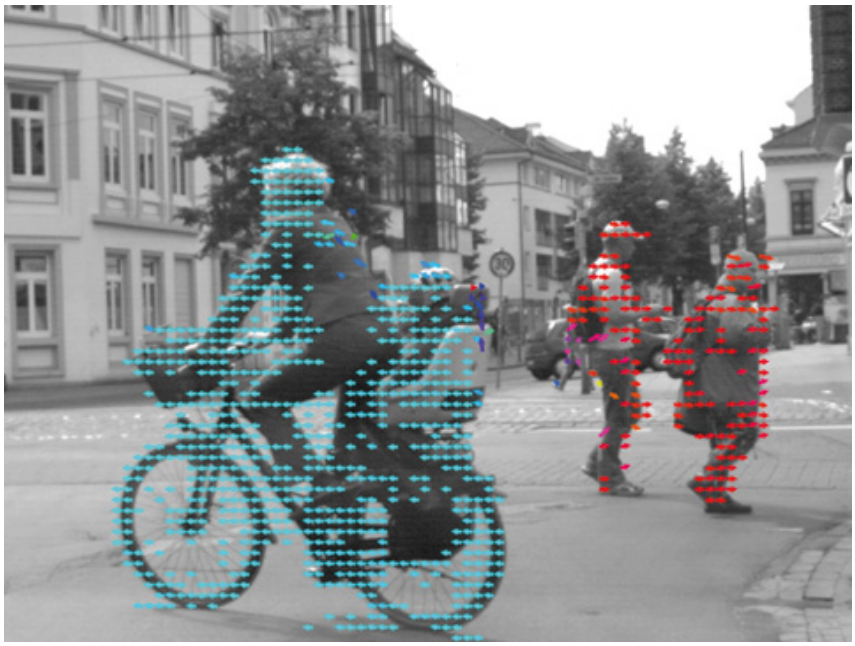


Obrázek 14: Obrazové modely z datasetu MSR Action3D Dataset.[38]

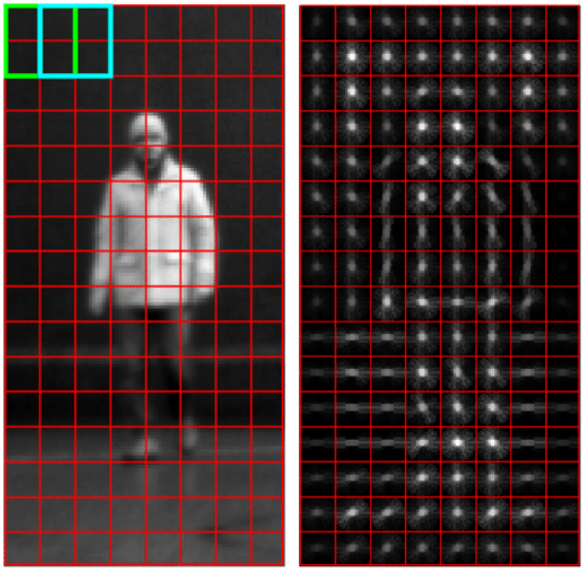
Další třídou obrazových modelů je využití optického toku extrahovaného z po sobě jdoucích snímků viz obrázek 15. Tato reprezentace je nezávislá na odstranění pozadí snímků, což zajišťuje v mnoha případech lepší použitelnost. Tyto metody jsou založeny na předpokladu, že změny mezi jednotlivými snímky jsou způsobeny důsledkem pohybu. Nereagují tak na změny textur a osvětlení.[42]

Jednou z dalších tříd obrazových modelů je extrakce příznaků na základě gradientu. Gradient v oblasti počítačového vidění je chápán jako změna intenzity v nějakém směru.[29] Každý snímek je tak reprezentován histogramem těchto gradientů. Jako zástupce tohoto přístupu lze uvést **HOG**[34] (histogram orientovaných gradientů) deskriptor. Tento deskriptor nepočítá histogram z celého snímku, ale tento snímek si rozdělí do několika vzajemně se přesahujících bloků a vypočítá histogram uvnitř každého bloku. Tento přístup sdílí mnoho vlastností jako při využití optického toku a není tak závislý na odstranění pozadí. Ukázka je na obrázku 16 a příklad algoritmu je na obrázku 17. Na rozdíl od optického toku lze pomocí gradientu popsat i nepohyblivé scény.[42]

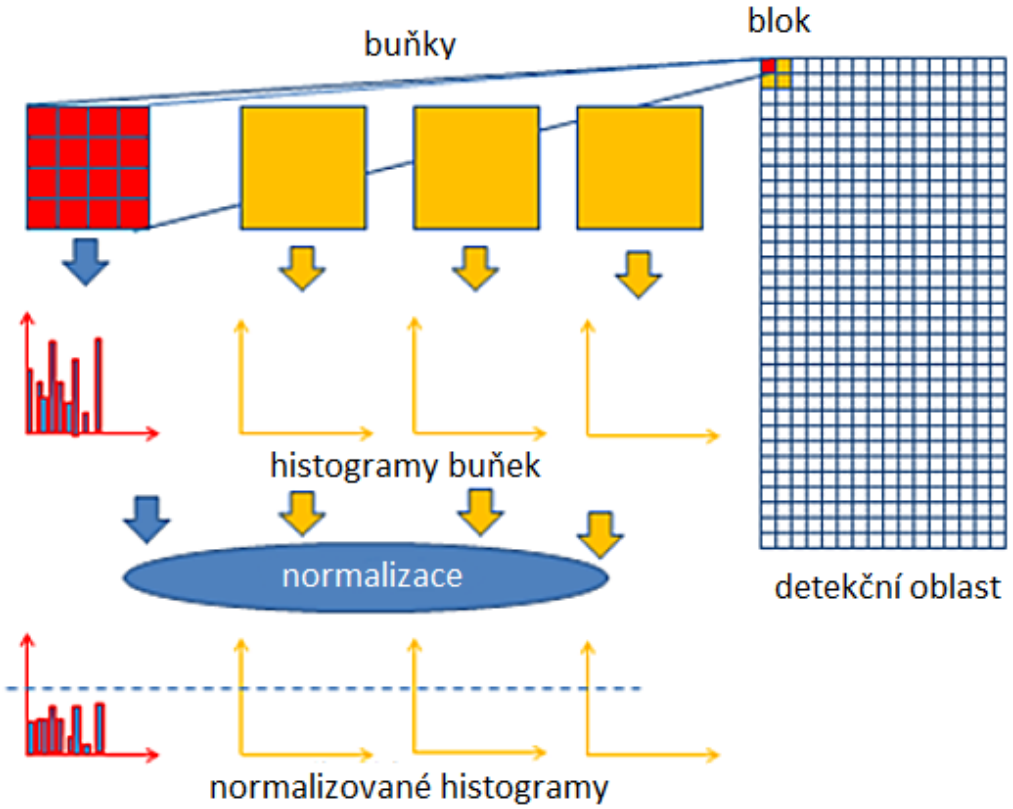
Poslední třída obrazových modelů je založena na neurovědecky inspirovaném přístupu HMAX[35], který kombinuje Gaborovy filtry[36] s optickým tokem v max pooling systému[37].



Obrázek 15: Zobrazení optického toku v obraze.[30]



Obrázek 16: (a) původní snímek, (b) snímek po aplikaci HOG deskriptoru.[32]



Obrázek 17: Příklad HOG algoritmu. Celá detekční oblast se rozdělí na malé spojené oblasti zvané buňky. Skupiny sousedních buněk jsou považovány za prostorové oblasti zvané bloky. Seskupování buněk do bloku je základem pro seskupování a normalizaci histogramů. Pro pixely v každé buňce se poté vypočte histogram směrů gradientů nebo histogram hranové orientace. Normalizovaná skupina těchto histogramů představuje blokový histogram a soubor těchto blokových histogramů představuje deskriptor. [33]

4.4.3 Metody využívající prostorové statistiky

Metody využívající prostorové statistiky jsou založeny na rozkladu snímků, videosekvencí do menších oblastí, které nejsou závislé na částech lidského těla nebo obrazovém systému souřadnic. K rozpoznávání akcí pak dochází na základě získávání statistik lokálních příznaků z těchto oblastí. Hlavní výhodou této metody je nezávislost na pojmenování částí lidkého těla, jeho detekci a lokalizaci. Tento přístup je založen na strategii zdola nahoru, kde jsou nejdřív detekovány zájmové body v obraze (zejména v strukturách jako jsou rohy) a poté je každé oblasti přiřazena předpřipravená množina příznaků.[42]

4.5 Přehled konkrétních metod

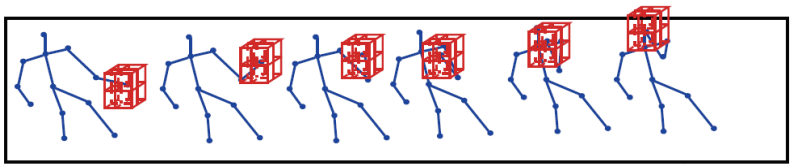
Většína metod rozpoznávání akcí je založena na modelech lidského těla nebo na obrazových modelech. Mezi metody založené na modelech lidského těla patří například Temporal Self-

Similarities[51], Matrix Descriptor of Changes[63], mezi metody založené na obrazových modelech lze zařadit Random Occupancy Patterns[46], Space-Time Interest Points[48], Space-Time Occupancy Patterns[50], Depth Motion Maps[52] a mezi metody využivající obou přístupu patří Local Occupancy Patterns[45].

4.5.1 LOP – Local Occupancy Patterns

Použití pouze 3D pozic lidských kloubů nemusí být zcela dostatečné k reprezentaci akcí, obzvláště když akce zahrnuje interakce mezi předmětem a jinými objekty. Proto je nutné navrhnout takový příznak, který dokáže přesně popsat hloubku pro jednotlivé klouby. Na obrázku 18 je osoba, která pije z hrnku. Když uchopí hrnek je prostor okolo ruky zakryt hrnkem. Při zvednutí hrnku dojde k zakrytí prostoru jak okolo ruky, tak i okolo hlavy. Tyto informace mohou být užitečné pro charakterizaci této interakce a odlišit pití od jiných činností. Právě tyto interakce mezi člověkem a předmětem jsou charakterizovány jako LOP neboli místní vzory obsazení.

V každém snímku je pro LOP příznak vypočítána informace o místním obsazení, která je založena na mračnu bodů (point cloud) v 3D uspořádání okolo konkretního bodu. Toto mračno bodů je generováno z vstupní hloubkové mapy. V okolí každého z kloubů na lidském těle je pak vytvořena mřížka, která obsahuje tzv. biny, což jsou buňky dané mřížky. Poté dojde k sečtení bodů z mračna bodů, které spadají do binů mřížky. Nasledně je pak aplikována sigmoidní normalizační funkce k získání informace o místním obsazení. LOP příznak je tedy vektor obsahující z příznaků všech binů v prostorové mřížce okolo vybraného kloubu. [45]

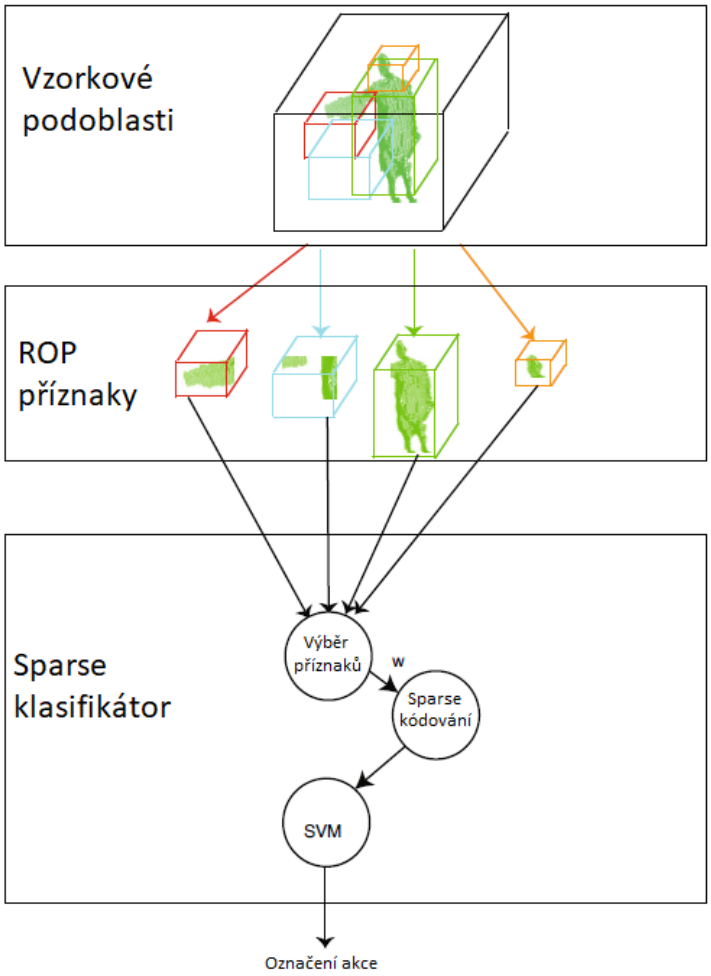


Obrázek 18: Vyobrazení vzoru obsazenosti okolo zápěstí a hlavy.[45]

4.5.2 ROP – Random Occupancy Patterns

ROP vzniklo z důvodu nedostatečné spolehlivosti algoritmů založených na sledování kostry v případě, když nastane několikanásobná okluze. Okluzí se rozumí stav, kdy nejsou vidět jednotlivé klouby, které jsou potřebné ke konstrukci kostry člověka. ROP proto využívá jako vstup hloubkové mapy. Při aplikaci těchto příznaků se zachází s trojrozměrným snímkem sekvence jako s 4D objektem, přičemž jednotlivé příznaky jsou extrahovány z náhodně vybraných 4D podoblastí různých velikostí a na různých místech. Vzhledem k tomu, že příznaky ROP jsou extrahovány ve větším měřítku jsou robustní vůči šumu a zároveň jsou méně citlivé na okluzi, protože získávají pouze informace z oblastí, které jsou pro danou akci nejvíce popisující.

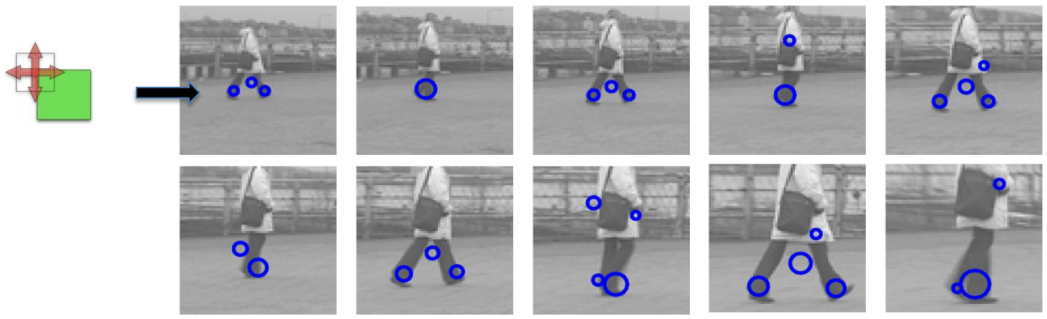
Za použití regularizační metody Elastic-Net[47] jsou poté vybrány příznaky, které jsou pro klasifikaci nejužitečnější. Celý algoritmus je znázorněn na obrázku 19. Výhodou tohoto přístupu je vyšší rychlost klasifikování, jelikož počet příznaků je menší.[46]



Obrázek 19: Ukázka průběhu rozpoznávání akcí metodou ROP. Nejdřív dojde k vybrání podoblastí, poté následuji ROP příznaky a nakonec dochází ke klasifikaci akcí.[46]

4.5.3 STIP – Space-Time Interest Points

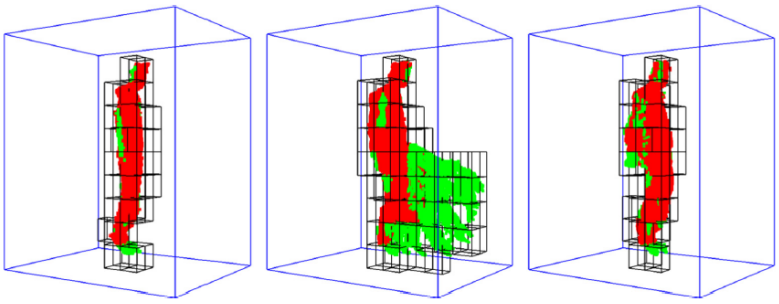
STIP je metoda založena na extrakci a detekci zájmových bodů, zejména rohů v obraze a jejich následným sledováním v čase. Tato metoda byla navrhuta vědcem I. Laptevem a je rozšířením 2D Harrisnova detektoru rohů.[48] Na obrázku 20 je ukázka detekce zájmových bodů na chůzi osoby. Těmito body jsou tak rohy na nohách a rukách, které se sledují v čase vykonávání akce.



Obrázek 20: Znázornění STIP při chůzi.[49]

4.5.4 STOP – Space-Time Occupancy Patterns

Tato metoda slouží k rozpoznávání akcí v 3D prostoru, kdy vstupem jsou hloubkové mapy bez dodatečné extrakce pozic kloubů v prostoru. STOP pak reprezentuje tyto hloubkové mapy v 4D časoprostorové mřížce (viz obrázek 21), kde osy času a prostoru jsou rozděleny do několika segmentů. Výhodou STOP je zachování jak prostorové, tak i časové informace mezi jednotlivými buňkami v časoprostoru a zároveň jsou dostatečně flexibilní k přizpůsobení různých variací akcí. Samotné buňky se pak obvykle skládají ze siluety člověka nebo pohybujících se částí těla. Tyto buňky tudíž obsahují důležité informace pro samotné rozpoznávání akcí. Vypovídající hodnotou je tak jejich míra obsazenosti v prostoru.[50]



Obrázek 21: Znázornění STOP v hloubkové mapě v časoprostoru.[50]

5 Vybrané metody a jejich implementace

5.1 Depth Motion Maps

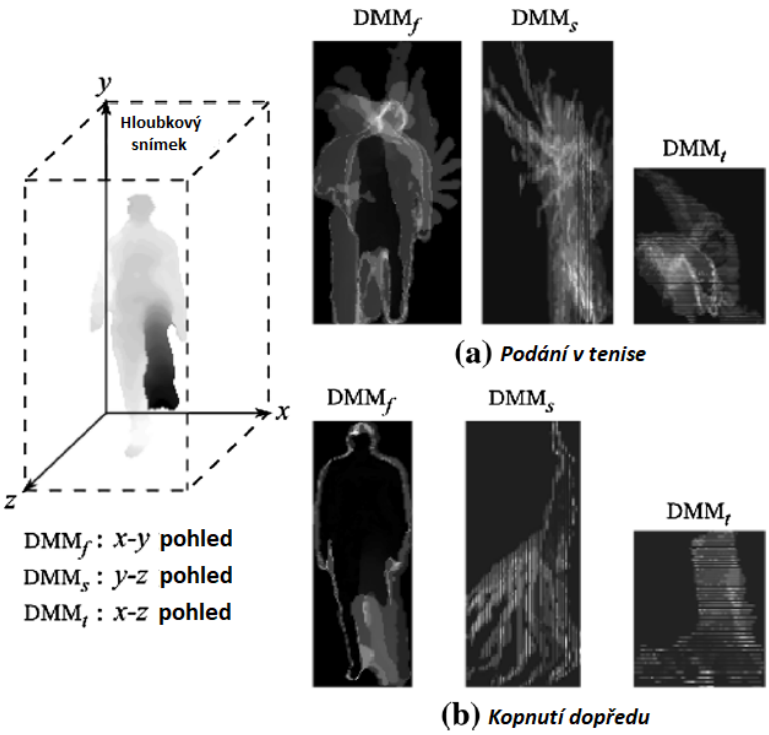
Jedná se o metodu rozpoznávání akcí využivající jako vstup hloubkové mapy. Jako příznaky jsou pak použity hloubkové pohybové mapy (Depth Motion Maps – DMM), které jsou generovány akumulací energie pohybu z hloubkových map promitaných ze tří projekčních pohledů (čelního pohledu, bočního pohledu a horního pohledu). V porovnání s 3D hloubkovými mapami jsou hloubkové pohybové mapy 2D obrazy, které poskytují kódování pohybových charakteristik akcí.

Hloubková mapa může být použita pro zachycení 3D struktury a informace o tvaru. Pro DMM se používá každá hloubková mapa ke generování tří 2D projekčních map, které odpovidají přednímu, bočnímu a hornímu pohledu viz obrázek 22. Tyto mapy označujeme jako $mapa_v$, kde $v \in \{f(front), s(side), t(top)\}$. Pro každou projekční mapu se pak energie pohybu vypočítá jako absolutní rozdíl dvou po sobě jdoucích map bez prahování. Pro sekvenci hloubkových map s N snímky je výpočet následující:

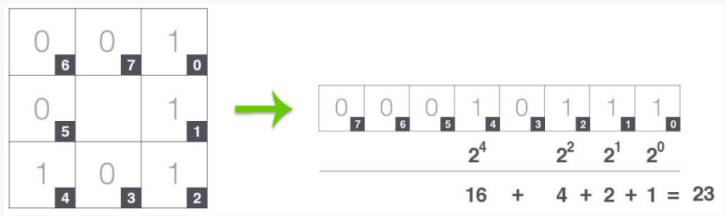
$$DMM_v = \sum_{i=a}^{b} \mid map_v^i - map_v^{i-1} \mid, \tag{1}$$

kde i značí index snímku, map_v^i je projekční mapa i-tého snímku z projekčního pohledu v, $a \in \{2, \ldots, N\}$ a $b \in \{2, \ldots, N\}$ označují rozsah snímků.[52] Rozšířením metody Depth Motion Maps je metoda Edge Enhanced Depth Motion Map (E2DMM)[65], která rozpoznává dynamická gesta rukou na základě hloubkového videa. K zachycení více informací o časové struktuře gest se v této metodě používá dynamická časová pyramida (DTP).[65]

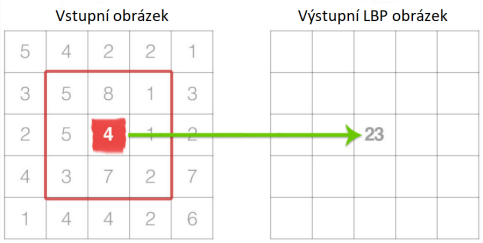
Pro zvýšení efektivity rozpoznávaní akcí je možné aplikovat histogram orintovaných gradientů (HOG)[34][55] nebo lokální binární vzory (LBP)[53][54]. Histogram orientovaných gradientů je popsán v kapitole 4.4.2. Lokální binární vzory jsou deskriptorem textur. LBP pak vypočítavají lokální reprezentaci textur. Tato lokální reprezentace je vytvořena porovnáním každého pixelu s okolními pixely. Pro každý pixel v obrázku je vybrána oblast velikosti r obklopující středový pixel. Hodnota LBP se pak vypočítá pro tento středový pixel a uloží do výstupního 2D pole se stejnou šířkou a výškou jako vstupní obraz. Ukázka funkce je na obrázku 23 a na obrázku 24 je ukázka po aplikaci LBP.[53]



Obrázek 22: Ukázky DMM pro a: podání v tenise, b: kopnutí vygenerované ze sekvence hloubkových map.[52]

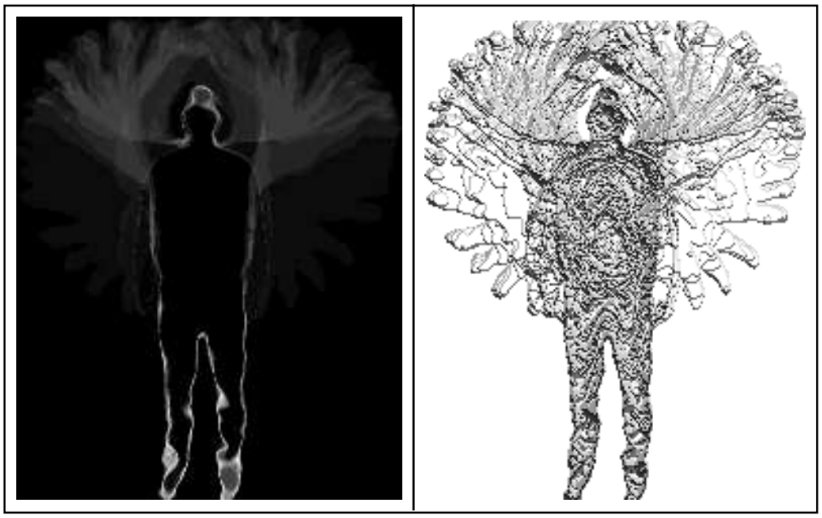


(a)



(b)

Obrázek 23: Prvním krokem při konstrukci LBP je vzít 8 sousedních pixelů obklopujících středový pixel a pomocí jejich prahu se vytvoří 8 binárních číslic. Na obrázku (a) je druhý krok a to převzetí 8-bitových binárních sousedů středového pixelu a jejich převedení na desetinnou reprezentaci. Na obrázku (b) je uložení vypočteného LBP do výstupního pole se stejnou šířkou a výškou jako původní obraz.[53]



Obrázek 24: Ukázka po aplikování LBP.[54]

5.2 Temporal Self-Similarities

Tato metoda rozpoznávání akcí je založena na soběpodobnosti sekvence snímků v průběhu času, kdy pro danou sekvenci snímků se vypočítá prostorová vzdálenost mezi jednotlivými klouby na těle člověka pro všechny páry snímků a tyto výsledky se uloží do matice soběpodobnosti (SSM). SSM je tudíž grafické vyobrazení podobných snímků v sérii dat. Pro výpočet SSM se používa tohoto vzorce:

$$D(I) = [d_{ij}]_{ij=1,2,\ldots,T} = \begin{bmatrix} 0 & d_{12} & d_{13} & \ldots & d_{1T} \\ d_{21} & 0 & d_{23} & \ldots & d_{2T} \\ \vdots & \vdots & \vdots & & \vdots \\ d_{T1} & d_{T2} & d_{T3} & \ldots & 0 \end{bmatrix}. \tag{2}$$

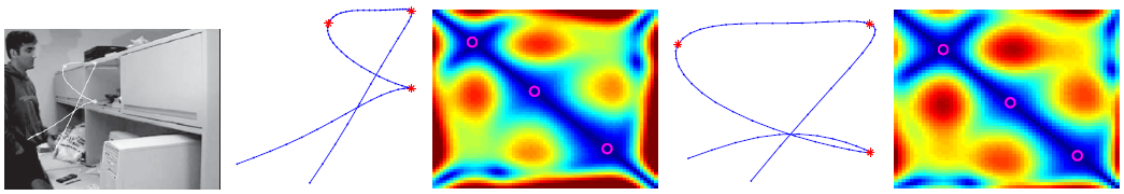
Na diagonále této matice jsou vždy nulové hodnoty, jelikož diagonála odpovídá porovnání snímků se sebou samými a dochází tak ke stoprocentní shodě.

SSM jsou poměrně robustní a celkem snadné na výpočet. Jelikož by bylo neefektivní počítat matici soběpodobnosti pro každý kloub zvlášť, dochází v každém snímku sekvence ke zprůměrování hodnot Euklidovské vzdálenosti všech dvojic kloubů. Pro výpočet Euklidovské vzdálenosti se využívá vzorce:

$$d_{ij} = \frac{1}{k} \sum_{k} || \, x_i^k - x_j^k \, ||_2, \tag{3}$$

kde x_i^k, x_j^k značí pozici bodů na dráze k ve snímcích i, j. Výstupem této metody jsou tak symetrické matice různých velikostí (viz obrázek 25), které jsou po úpravě na stejnou velikost použity jako vstup do klasifikátoru akcí. Pro úpravu na stejnou velikost je možné použít nelineární redukci velikosti. Mezi populární metody nelinární redukce velikosti patří např. Isomap. Tato metoda vypočte vzdálenosti mezi všemi páry obrázků. Tyto vypočtené vzdálenosti představují matici sousednosti, v níž každý obraz představuje uzel v grafu.[51] V této bakalářské práci byl

proveden experiment, přičemž matice soběpodobnosti byly upraveny na stejnou velikost pomocí interpolace.



Obrázek 25: Srovnání SSM pro dvě osoby, které otevírají skřínku. Na obrázcích (b) a (d) je znázorněna trajektorie pohybu ruky a na obrázcích (c) a (e) jsou vypočteny matice soběpodobnosti pro obě trajektorie. Na obou maticích lze vidět podobné vzory.[51]

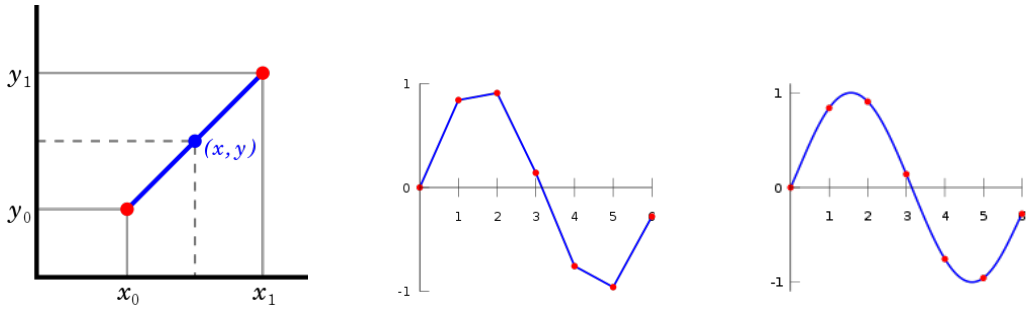
Interpolace je způsob konstrukce nových datových bodů v rozsahu diskrétní množiny (množina která je tvořena pouze izolovanými body) známých datových bodů. Jedná se tedy o dodatečné vytváření obrazových bodů. Je několik metod interpolace.[59]

Jednou z nejjednodušších metod je **lineární interpolace**. Jedná se o metodu prokládání křivek za použití lineárních mnohočlenů. Pokud jsou dány dva známé body souřadnicemi (x_0, y_0) a (x_1, y_1), lineární interpolace je potom přímka mezi těmito dvěma body viz obrázek 26. Lineární interpolace je rychlá a snadná, ale není příliš přesná. Další nevýhodou je, že interpolant není diferencovatelný v místě x_k.[57]

Rozšířením lineární interpolace pro interpolaci funkce dvou proměnných na pravidelnou prostorovou mřížku je **bilineární interpolace**. Tato metoda provádí lineární interpolaci nejprve v jednom směru a pak i ve druhém směru.[58] Další metodou je **polynomiální interpolace**, což je zobecnění lineární interpolace. Lineární interpolant lineární interpolace je nahrazen polynomem vyššího stupně viz obrázek 26. Obecně platí, že pokud je n datových bodů, existuje přesně jeden polynom stupně nejvýše $n-1$ procházející všemi datovými body. Interpolační chyba je úměrná vzdálenosti mezi datovými body a mocniny n. Interpolant je navíc polynom a tedy nekonečně diferencovatelný. Polynomiální interpolace tak překonává většinu problémů lineární interpolace. Nicméně polynomiální interpolace má také nevýhody. První z nich je výpočetní náročnost interpolačního polynomu ve srovnání s lineární interpolací. Druhou je, že polynomiální interpolace může vytvářet oscilační artefakty a to obzvláště u koncových bodů.[59]

Jednou z dalších metod je **spline interpolace**, která používá polynomy nízkého stupně v každém z intervalů a vybírá polynomiální kusy tak, aby se hodily k sobě. Výsledná funkce se nazývá spline. Interpolace spline, stejně jako polynomiální interpolace, způsobuje menší chybu než lineární interpolace a interpolant je hladší. Interpolant je však snazší vyhodnotit než polynomy vysokého stupně použité v polynomiální interpolaci.[59] Mezi spline interpolace se řadí **kubická interpolace**, také označována jako kubický Hermite spline a **bikubická interpolace**,

což je rozšíření kubické interpolace pro interpolaci datových bodů na dvojrozměrné pravidelné mřížce.[60]



(a) (b) (c)

Obrázek 26: Na obrázku (a) jsou dány dva červené body, modrá úsečka lineární interpolace mezi těmito body a hodnoty x a y jsou nalezeny pomocí lineární interpolace. Na obrázku (b) je graf dat s aplikovanou lineární interpolací a na obrázku (c) je graf dat s aplikovanou polynomiální interpolací.[57][59]

6 Testování a výsledky

Aplikace byla vyvíjena v prostředí Microsoft Visual Studio 2015 v programovacím jazyce C++ (viz kapitola 2.1) s využitím knihovny OpenCV ve verzi 3.2.0 (viz kapitola 2.1) a testována na notebooku ASUS K70IC s operačním systému Windows 10 64-bit, procesorem Intel Core 2 Duo P8700 2.53 GHz a 4GB RAM. K testování metod byly použity dva datasety z kapitoly 2.5.

U datasetu MSR Action3D se u prvního testu využívá na trénování první polovina subjektů a na testování druhá polovina subjektů (dále jen S 1:1). U druhého testu se využívá k trénování první epizoda a k testování druhá a třetí epizoda (dále jen E 1:2). U datasetu UTKinect-Action3D se u prvního testu využívá na trénování první polovina subjektů a na testování druhá polovina subjektů (dále jen S 1:1). U druhého testu se využívá k trénování první epizoda a k testování druhá epizoda (dále jen E 1:1). Epizodou se rozumí opakování určité akce daným subjektem. Pro klasifikaci akcí byla využita třída klasifikátoru SVM z knihovny OpenCV s experimentálním nastavením na kernel průsečíků histogramů (intersection kernel) (dále jen SVM_{int}) a pak na lineární kernel (linear kernel) (dále jen SVM_{lin}). Úspěšnost rozpoznávání byla vypočítána pomocí následujících vzorců [61]:

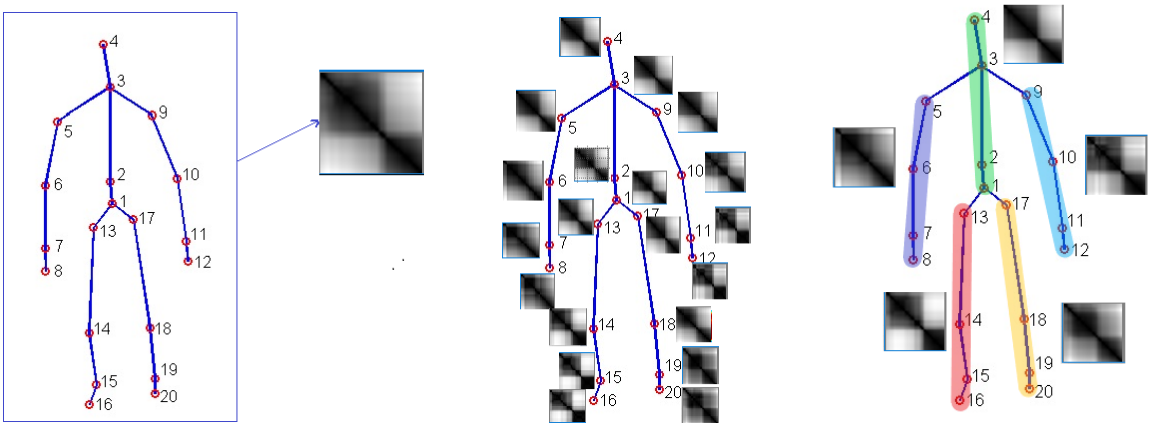
$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4}$$

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \tag{5}$$

kde F1 a ACC je typ určení procentuální úspěšnosti při testování predikce na testovacích datech. Tyto vzorce využívají proměnné TP (true-positive), TN (true-negative), FP (false-positive) a FN (false-negative). TP je celkový počet kdy se detekovaná akce shoduje se skutečnou akcí. TN je celkový počet detekování, že se o danou akci nejedná. FP je celkový počet kdy se detekovaná akce neshoduje se skutečnou akcí. FN je celkový počet kdy se skutečná akce neshoduje s detekovanou akcí.

U metody Temporal Self-Similarities byla pro úpravu matic soběpodobnosti na stejnou velikost využita kubická interpolace. Pro expirementování se v této práci použily matice o velikosti 55×55. Tato velikost byla zvolena pro zachování dostatečné velikosti při zobrazení a zároveň pro přijatelnou velikost natrénovaných souborů z SVM. Dalším důvodem pro zvolení této velikosti byl počet snímků, který se pohyboval kolem zvolené velikosti matice. Při prvním testování se matice soběpodobnosti počítala nad všemi klouby dohromady. Ovšem tyto testy měly celkem nízkou přesnost rozpoznávání. Proto i přes nižší efektivnost, byl vyzkoušen způsob s výpočtem matice soběpodobnosti pro každý kloub zvlášť. Výsledky rozpoznávání byly mnohem lepší, ale čas klasifikace byl delší. Tudíž byl proveden třetí experiment, kdy se matice soběpodobnosti počítala zvlášť pro trup s hlavou (klouby 1 až 4) a každou končetinu (levá ruka – klouby 9 až 12, pravá ruka – klouby 5 až 8, levá noha – klouby 17 až 20, pravá noha – klouby 13 až 16) bez následného centrování. Ukázky těchto experimentů jsou na obrázku 27. Náročnost testů S 1:1

je vyšší na rozpoznávání akcí, protože je naučena jen první polovina subjektů. Tyto subjekty vykonávají dané akce odlišným pohybem oproti druhé polovině subjektů. Testy E 1:1 pro dataset UTKinect-Action3D nebo E 1:2 pro dataset MSR Action3D jsou pro rozpoznávání akcí snazší, protože jsou naučené všechny subjekty, které v testech zopakují danou akci jedenkrát pro E 1:1 nebo dvakrát pro E 1:2. Proto jsou odlišnosti ve vykonávání akcí podstatně menší oproti testům S 1:1.



Obrázek 27: Ukázka matice soběpodobnosti pro chůzi člověka. Na obrázku (a) je výpočet matice soběpodobnosti nad všemi klouby dohromady. Na obrázku (b) je výpočet matice soběpodobnosti pro každý kloub kostry dané akce zvlášť a na obrázku (c) je výpočet matice soběpodobnosti zvlášť pro trup s hlavou a končetiny.[62]

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	93.15	31.49	84	91
S 1:1 (SVM_{lin})	93.08	30.8	62	65
E 1:2 (SVM_{int})	94.04	40.43	60	105
E 1:2 (SVM_{lin})	93.45	34.5	43	80

(a)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	94.53	45.33	988	726
S 1:1 (SVM_{lin})	94.46	44.64	471	405
E 1:2 (SVM_{int})	95.88	58.76	544	760
E 1:2 (SVM_{lin})	95.74	57.41	316	495

(b)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{inbt})	94.12	41.18	277	218
S 1:1 (SVM_{lin})	94.15	41.52	155	139
E 1:2 (SVM_{int})	95.31	53.10	161	237
E 1:2 (SVM_{lin})	94.82	48.25	101	171

(c)

Tabulka 2: Úspěšnost metody Temporal Self-Similarities na datasetu MSR Action3D. V tabulce (a) je úspěšnost, kdy je SSM vypočítána nad všemi klouby dohromady. V tabulce (b) je úspěšnost, kdy je SSM vypočítána pro každý kloub v akci zvlášť a v tabulce (c) je úspěšnost, kdy je SSM vypočítána pro trup s hlavou a každou končetinu v akci zvlášť.

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	91.52	57.58	20	16
S 1:1 (SVM_{lin})	89.7	48.48	15	14
E 1:1 (SVM_{int})	93.94	69.7	20	18
E 1:1 (SVM_{lin})	92.12	60.61	14	14

(a)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	94.14	70.71	194	150
S 1:1 (SVM_{lin})	93.13	65.66	136	112
E 1:1 (SVM_{int})	94.34	71.72	191	151
E 1:1 (SVM_{lin})	93.74	68.69	128	113

(b)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	93.13	65.66	55	45
S 1:1 (SVM_{lin})	91.92	59.6	38	35
E 1:1 (SVM_{int})	93.74	68.69	53	43
E 1:1 (SVM_{lin})	92.53	62.93	39	34

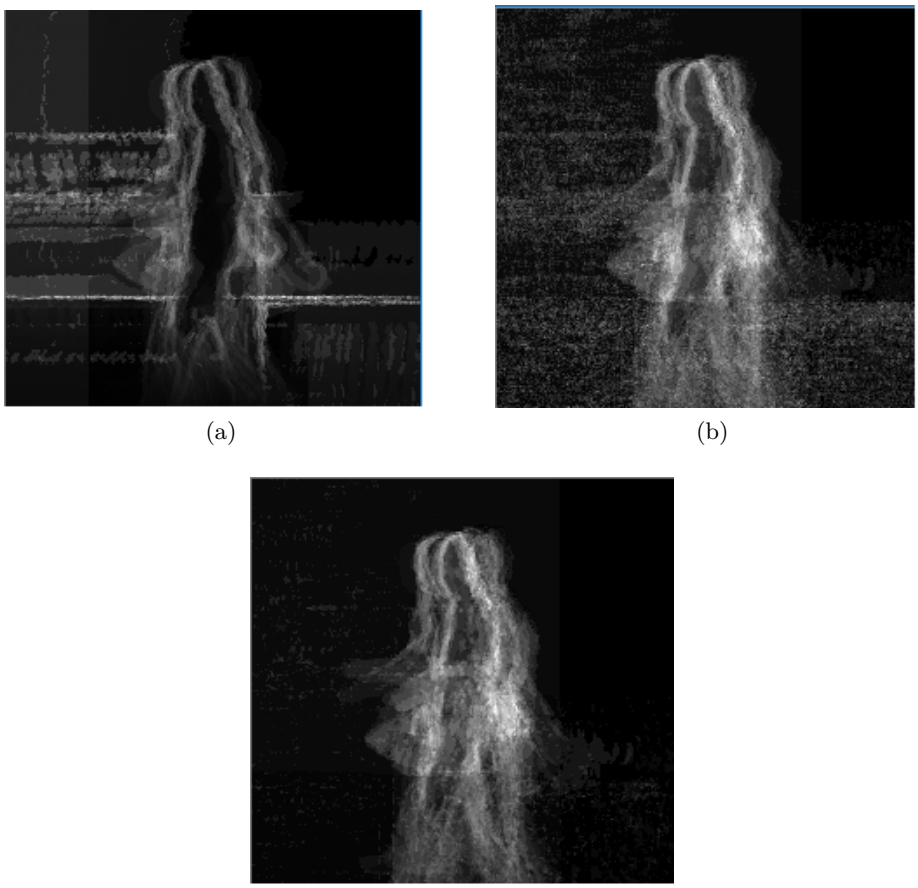
(c)

Tabulka 3: Úspěšnost metody Temporal Self-Similarities na datasetu UTKinect-Action3D. V tabulce (a) je úspěšnost, kdy je SSM vypočítána nad všemi klouby dohromady. V tabulce (b) je úspěšnost, kdy je SSM vypočítána pro každý kloub v akci zvlášť a v tabulce (c) je úspěšnost, kdy je SSM vypočítána pro trup s hlavou a každou končetinu v akci zvlášť.

Z tabulek 2b a 3b je patrné, že metoda Temporal Self-Similarities na obou datasetech dosahuje nejvyšší F_1 úspěšnosti na úkor časové náročnosti, když se SSM počítá pro každý kloub v akci zvlášť. Při počítání SSM zvlášť pro trup s hlavou a končetiny dosahovala metoda Temporal Self-Similarities přibližně o 5% nižší F_1 úspěšnosti, ale s přibližně třikrát nižší časovou náročností (viz tabulka 2c a 3c).

U metody Depth Motion Maps pro dataset UTKinect-Action3D proběhly dva testy. Jeden bez odstranění pozadí, druhý s aplikovanou metodou na odstranění pozadí MOG2 3.1.1. Přesto po aplikování MOG2 zůstal ve snímcích šum, proto byl proveden experiment s aplikovanou metodou na odstranění pozadí MOG2 3.1.1 a morfologickým operátorem – erozí (viz kapitola 3), která tento šum redukovala. Eroze proběhla na každém snímku jednou s nastaveným obdélníkovým jádrem o velikosti 2×2. Ukázky jsou na obrázcích 28. Pro každou sekvenci činnosti byl

vytvářen nový model pozadí. Po segmentaci pozadí bylo na obou datasetech provedeno ořezávání snímků na velikost 240×240. U datasetu UTKinect-Action3D byly snímky ořezávány na základě středu pánve. Mapování tohoto bodu do hloubkové mapy proběhlo převedením souřadnic tohoto 3D bodu, které byly v metrech do 2D obrazů.



(a) (b)

(c)

Obrázek 28: Ukázka DMM pro chůzi člověka. Na obrázku (a) je DMM vytvořená ze snímků bez odstraněného pozadí. Na obrázku (b) je DMM vytvořená ze snímků s odstraněným pozazadím pomocí MOG2 a na obrázku (c) je DMM vytvořená ze snímků s odstraněným pozadím pomocí MOG2 a aplikovanou erozí.[62]

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	95.7	57.04	1016	973
S 1:1 (SVM_{lin})	95.09	50.86	562	598
E 1:2 (SVM_{int})	99.38	93.8	631	968
E 1:2 (SVM_{lin})	99.03	90.3	402	691

Tabulka 4: Úspěšnost metody Depth Motion Maps na datasetu MSR Action3D

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	91.63	58.16	245	305
S 1:1 (SVM_{lin})	91.22	56.12	186	187
E 1:1 (SVM_{int})	92.65	63.27	300	276
E 1:1 (SVM_{lin})	90.82	54.08	183	238

(a)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	93.88	69.39	367	247
S 1:1 (SVM_{lin})	94.29	71.43	294	232
E 1:1 (SVM_{int})	93.88	69.39	343	311
E 1:1 (SVM_{lin})	95.10	75.51	237	273

(b)

	$ACC[\%]$	$F_1[\%]$	Čas $[s]$	
			trénování	testování
S 1:1 (SVM_{int})	92.86	64.29	390	312
S 1:1 (SVM_{lin})	93.88	69.39	267	216
E 1:1 (SVM_{int})	93.27	66.33	341	328
E 1:1 (SVM_{lin})	94.9	74.49	269	291

(c)

Tabulka 5: Úspěšnost metody Depth Motion Maps na datasetu UTKinect-Action3D. V tabulce (a) je úspěšnost na snímcích bez odstraněného pozadí. V tabulce (b) je úspěšnost na snímcích s aplikovaným MOG2 a v tabulce (c) je úspěšnost na snímcích s aplikovaným MOG2 a erozí.

U metody Depth Motion Maps na datasetu UTKinect-Action3D je vidět z tabulky 5b, že segmentace pozadí pomocí MOG2 pomohla k lepším výsledkům, a to zejména při nastavení SVM na lineární kernel, přičemž F_1 úspěšnost stoupla až o 21.43 % oproti snímkům bez odstraněného pozadí. Časová náročnost přitom stoupla u testování pro S 1:1 (SVM_{lin}) o 45 sekund a pro E 1:1 o 35 sekund. Pro test S 1:1 (SVM_{int}) časová náročnost naopak klesla o 58 sekund. Redukování šumu z MOG2 snímků pomocí eroze nevedlo k lepší úspěšnosti (viz tabulka 5c). Na datasetu MSR Action3D dosahovala metoda Depth Motion Maps mnohem lepších výsledků než metoda Temporal Self-Similarities, viz výsledky z tabulek 2 a 4. Na datasetu UTKinect-Action3D byla úspěšnost obou metod podobná, viz výsledky z tabulek 3 a 5. Při preciznějším odstranění pozadí by metoda Depth Motion Maps pravděpodobně dosahovala ještě lepších výsledků. Ovšem časová náročnost metody Depth Motion Maps je výrazně vyšší než u metody Temporal Self-Similarities.

Z matic záměn pro testy S 1:1 s SSM_{int} (viz příloha A) je vidět, že nejobtížnější na detekci

byly u datasetu MSR Action3D pro metodu Temporal Self-Similarities tyto akce: zatloukání (nedokázalo rozpoznat), úder dopředu (nedokázalo rozpoznat), hod (nedokázalo rozpoznat), boxování do boku (nedokázalo rozpoznat), kopnutí do boku (nedokázalo rozpoznat) a pro metodu Depth Motion Maps to byly tyto akce: zatloukání (nedokázalo rozpoznat), podání ruky (nedokázalo rozpoznat), úder dopředu (nedokázalo rozpoznat), hod (nedokázalo rozpoznat), kreslení X (zaměňováno za kreslení fajfky a kruhu), úder tenisovou raketou (zaměňováno za mávání a kreslení kruhu). U datasetu UTKinect-Action3D to byly pro metodu Temporal Self-Similarities tyto akce: nesení (zaměňováno za chození a zvednutí), hození (zaměňováno za tlačení, tažení a sednutí), tlačení (zaměňováno za chůzi, sednutí a tažení), tažení (zaměňováno za nesení a postavení) a pro metodu Depth Motion Maps to byly tyto akce: tlačení (zaměňováno za tažení a tleskání), hození (zaměňováno za tlačení a tažení), tažení (zaměňováno za tleskání, hození a postavení), tleskání(zaměňováno za tažení). Záměny jednotlivých akcí mohly být způsobeny podobným pohybem při vykonávání akce a také akcemi, které jsou zaměnitelné i v běžném životě.

7 Závěr

Cílem této bakalářské práce bylo popsat a seznámit se se základními technikami detekce pohybu ve videosekvencích, nastudovat metody Depth Motion Maps (zástupce hloubkových map) a Temporal Self-Similarities (zástupce kostry), následně tyto metody implementovat s pomocí knihovny OpenCV v jazyce C++ a ověřit jejich funkčnost.

Metody byly otestovány na datasetech MSR Action3D a UTKinect-Action3D. U obou metod bylo pro klasifikaci použito SVM experimentálně nastavené na lineární kernel a na kernel průsečíků histogramů. Jak se ukázalo ve většině případů bylo dosaženo lepších výsledků při použití kernelu průsečíků histogramů.

V rámci metody Temporal Self-Similarities se dosahovalo nejlepších výsledků s výpočtem matice soběpodobnosti pro každý kloub zvlášť, ale s vyšší časovou náročností. Ideálním kompromisem se ukázalo počítat matice soběpodobnosti pro trup s hlavou a končetiny, kdy úspěšnost byla podobná a časová náročnost byla přibližně třikrát nižší.

Metoda Depth Motion Maps dosahovala nejlepších výsledků na datasetu MSR Action3D, který má u hloubkových map odstraněné pozadí. Na datasetu UTKinect-Action3D se do úspěšnosti promítl vliv pozadí na rozpoznávání akcí. Při aplikování metody MOG2 se úspěšnost zvýšila, ale odstranění pozadí touto metodou nebylo dokonalé.

Z výše uvedeného vyplývá, že metoda Depth Motion Maps je na datasetu MSR Action3D mnohem úspěšnější než metoda Temporal Self-Similarities. Na datasetu MSR Action3D dosahovala metoda Depth Motion Maps a metoda Temporal Self-Similarities podobných výsledků. Při důkladném odstranění pozadí z hloubkových map v datasetu UTKinect-Action3D by metoda Depth Motion Maps pravděpodobně dosahovala lepších výsledků než metoda Temporal Self-Similarities. Na druhou stranu časová náročnost metody Depth Motion Maps je vyšší.

Literatura

[1] C++. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2018-11-12 [cit. 2018-12-09]. Dostupné z: https://cs.wikipedia.org/wiki/C%2B%2B

[2] TIOBE Index [online]. 2018-12 [cit. 2018-12-09]. Dostupné z: https://www.tiobe.com/tiobe-index/

[3] About OpenCV [online]. [cit. 2018-12-09]. Dostupné z: https://opencv.org/about.htm

[4] Depth map. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2018-04-24 [cit. 2018-12-09]. Dostupné z: https://en.wikipedia.org/wiki/Depth_map

[5] Precision of the kinect depth camera [online]. 2017 [cit. 2018-12-09]. Dostupné z: https://stackoverflow.com/questions/7696436/precision-of-the-kinect-depth-camera

[6] Kinect. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2018-11-23 [cit. 2018-12-09]. Dostupné z: https://en.wikipedia.org/wiki/Kinect#Kinect_for_Windows

[7] The difference between Kinect v2 and v1 [online]. 2016 [cit. 2018-12-09]. Dostupné z: https://skarredghost.com/2016/12/02/the-difference-between-kinect-v2-and-v1/

[8] Schematic for Kinect V1. In: ResearchGate [online]. 2015 [cit. 2018-12-09]. Dostupné z: https://www.researchgate.net/figure/Schematic-for-Kinect-V1-Source-Microsoft-Library_fig1_291356921

[9] Využití senzoru Kinect pro komerční prezentaci produktů [online]. Praha, 2013 [cit. 2018-12-09]. Dostupné z: https://cyber.felk.cvut.cz/theses/papers/366.pdf. Diplomová práce. České vysoké učení technické v Praze.

[10] How The Kinect Works. In: Depth Biomechanics [online]. Sheffield Hallam University [cit. 2018-12-09]. Dostupné z: http://www.depthbiomechanics.co.uk/?p=100

[11] Depth Sensors Comparison [online]. 2018 [cit. 2019-01-27]. Dostupné z: http://docs.ipisoft.com/Depth_Sensors_Comparison

[12] Xia, L. and Chen, C.C. and Aggarwal. JK View invariant human action recognition using histograms of 3D joints: UTKinect-Action3D Dataset [online]. [cit. 2018-12-22]. Dostupné z: http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html

[13] Xia, L. and Chen, C.C. and Aggarwal. JK View invariant human action recognition using histograms of 3D joints: UTKinect-Action3D Dataset. In: UTKinect-Action3D Dataset

[online]. [cit. 2018-12-22]. Dostupné z: http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html

[14] MSR Action3D Dataset [online]. [cit. 2018-12-22]. Dostupné z: https://www.uow.edu.au/~wanqing/#Datasets

[15] MSR Action 3D. In: Datasets [online]. [cit. 2018-12-22]. Dostupné z: http://users.eecs.northwestern.edu/~jwa368/my_data.html

[16] Cross-validation (statistics). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-21]. Dostupné z: https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[17] Počítačové vidění [online]. Vysoké učení technické v Brně, 2008 [cit. 2018-12-01]. Dostupné z: http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf

[18] Background subtraction techniques: a review [online]. [cit. 2019-01-05]. Dostupné z: http://profs.sci.univr.it/~cristanm/teaching/sar_files/lezione4/Piccardi.pdf

[19] Comparison of Background Subtraction Methods on Near Infra-Red Spectrum Video Sequences [online]. [cit. 2019-03-17]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1877705817327005

[20] How to Use Background Subtraction Methods. In: OpenCV [online]. [cit. 2019-01-05]. Dostupné z: https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

[21] Segmentace obrazu. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2013-02-12 [cit. 2019-01-05]. Dostupné z: https://cs.wikipedia.org/wiki/Segmentace_obrazu

[22] Thresholding. In: Scikit-image [online]. [cit. 2019-03-16]. Dostupné z: http://scikit-image.org/docs/0.12.x/auto_examples/segmentation/plot_otsu.html

[23] Detekce hran [online]. [cit. 2019-01-05]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise08/content_cz.php

[24] Canny Edge Detection [online]. [cit. 2019-03-04]. Dostupné z: https://docs.opencv.org/3.4/d7/de1/tutorial_js_canny.html

[25] Canny Edge Detection [online]. In: . [cit. 2019-01-05]. Dostupné z: https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

[26] Detekce pohybu v obraze z kamery [online]. [cit. 2019-01-05]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=132438

[27] VIVA Hand Detection. In: VIVA [online]. [cit. 2019-01-05]. Dostupné z: http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-detection/

[28] View Invariant Human Action Recognition Using Histograms of 3D Joints [online]. [cit. 2018-12-26]. Dostupné z: https://pdfs.semanticscholar.org/103a/c34236fb995809ec55b38c1d16c6df8b9f94.pdf

[29] Computer Vision: Correlation, Convolution, and Gradient [online]. [cit. 2018-12-27]. Dostupné z: https://www.slideshare.net/AhmedGadFCIT/computer-vision-correlation-convolution-and-gradient

[30] CVB Optical Flow. In: IMACO [online]. [cit. 2018-12-27]. Dostupné z: http://www.imaco.pl/pl/produkty/rodzina/cvb.optical-flow

[31] Graphical representation of the 20 joints composing the skeleton model used by the Kinect SDK for tracking the motion of a person. In: ResearchGate [online]. [cit. 2018-12-27]. Dostupné z: https://www.researchgate.net/figure/Skeleton-Graphical-representation-of-the-20-joints-composing-the-skeleton-model-used-by_fig7_285574233

[32] Histogram of oriented gradients descriptor. In: ResearchGate [online]. [cit. 2018-12-27]. Dostupné z: https://www.researchgate.net/figure/Histogram-of-oriented-gradients-descriptor-a-The-histogram-of-oriented-gradients-HoG_fig3_256451629

[33] Histogram of Oriented Gradients (HOG) Descriptor. In: Intel® Software [online]. [cit. 2019-03-19]. Dostupné z: https://software.intel.com/en-us/ipp-dev-reference-histogram-of-oriented-gradients-hog-descriptor

[34] Histograms of Oriented Gradients for Human Detection [online]. [cit. 2019-04-06]. Dostupné z: https://hal.inria.fr/file/index/docid/548512/filename/hog_cvpr2005.pdf

[35] Realistic Modeling of Simple and Complex Cell Tuning in the HMAX Model, and Implications for Invariant Object Recognition in Cortex [online]. [cit. 2019-04-06]. Dostupné z: http://riesenhuberlab.neuro.georgetown.edu/docs/publications/AIM-2004-017.pdf

[36] Gabor Filters [online]. [cit. 2019-04-06]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TRAPP1/filter.html

[37] Understanding of Convolutional Neural Network (CNN)-Deep Learning [online]. [cit. 2019-04-06]. Dostupné z: https://medium.com/\spacefactor\@m{}RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[38] Sample frames of the MSR-Action 3D dataset. In: ResearchGate [online]. [cit. 2018-12-27]. Dostupné z: https://www.researchgate.net/figure/Sample-frames-of-the-MSR-Action-3D-dataset_fig7_310824145

[39] Deep Learning [online]. [cit. 2019-03-04]. Dostupné z: https://www.techopedia.com/definition/30325/deep-learning

[40] Support vector machines. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-22]. Dostupné z: https://cs.wikipedia.org/wiki/Support_vector_machines

[41] Machine Learning Algorithms Today: Usage and Results [online]. 2017 [cit. 2019-02-01]. Dostupné z: https://www.dataversity.net/machine-learning-algorithms-today-usage-results/#

[42] Daniel Weinland, Rémi Ronfard, Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding. [online]. 2014 [cit. 2018-12-28]. Dostupné z: https://hal.inria.fr/hal-00640088/file/weinland10preprint.pdf

[43] Feature Extraction [online]. [cit. 2019-03-06]. Dostupné z: https://deepai.org/machine-learning-glossary-and-terms/feature-extraction

[44] Types of Machine Learning Algorithms: Supervised and Unsupervised Learning [online]. 2018 [cit. 2018-12-30]. Dostupné z: https://www.netguru.co/blog/types-of-machine-learning-algorithms-supervised-and-unsupervised-learning

[45] Mining Actionlet Ensemble for Action Recognition with Depth Cameras [online]. [cit. 2019-01-11]. Dostupné z: https://users.eecs.northwestern.edu/~jwa368/pdfs/actionlet.pdf

[46] Robust 3D Action Recognition with Random Occupancy Patterns [online]. [cit. 2019-01-12]. Dostupné z: https://users.eecs.northwestern.edu/~jwa368/pdfs/sampling_grid.pdf

[47] Regularization and variable selection via the elastic net [online]. [cit. 2019-04-06]. Dostupné z: https://web.stanford.edu/~hastie/Papers/B67.2%20%282005%29%20301-320%20Zou%20&%20Hastie.pdf

[48] On-line human action detection using space-time interest points [online]. [cit. 2019-01-12]. Dostupné z: http://www.fit.vutbr.cz/~ireznice/pubs.php?file=%2Fpub%2F9783%2FITAT2011.pdf&id=9783

[49] STIP: Spatio Temporal Interest Points [online]. [cit. 2019-01-12]. Dostupné z: http://www.micc.unifi.it/seidenari/wp-content/uploads/2010/01/A51-Spatio-temporal-features1.pdf

[50] STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences [online]. [cit. 2019-01-13]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.724.8106&rep=rep1&type=pdf

[51] Cross-View Action Recognition from Temporal Self-Similarities [online]. 2008 [cit. 2019-01-19]. Dostupné z: http://www.irisa.fr/vista/Papers/2008-eccv-junejo.pdf

[52] Real-time human action recognition based on depth motion maps [online]. 2013 [cit. 2019-01-20]. Dostupné z: https://www.utdallas.edu/~cxc123730/JRTIP-Action_Recognition.pdf

[53] Local Binary Patterns with Python & OpenCV [online]. [cit. 2019-03-19]. Dostupné z: https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

[54] Action Recognition from Depth Sequences Using Depth Motion Maps-based Local Binary Patterns [online]. [cit. 2019-04-06]. Dostupné z: https://www.researchgate.net/publication/268805560_Action_Recognition_from_Depth_Sequences_Using_Depth_Motion_Maps-based_Local_Binary_Patterns

[55] Human Action Recognition Based on DMMs, HOGs and Contourlet Transform [online]. [cit. 2019-04-06]. Dostupné z: https://www.researchgate.net/publication/275334526_Human_Action_Recognition_Based_on_DMMs_HOGs_and_Contourlet_Transform

[56] Histogram. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2018-11-21 [cit. 2019-02-03]. Dostupné z: https://cs.wikipedia.org/wiki/Histogram

[57] Lineární interpolace. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-18]. Dostupné z: https://cs.wikipedia.org/wiki/Line%C3%A1rn%C3%AD_interpolace

[58] Bilineární interpolace. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-18]. Dostupné z: https://cs.wikipedia.org/wiki/Biline%C3%A1rn%C3%AD_interpolace

[59] Interpolation. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-18]. Dostupné z: https://en.wikipedia.org/wiki/Interpolation

[60] Cubic Hermite spline. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-22]. Dostupné z: https://en.wikipedia.org/wiki/Cubic_Hermite_spline

[61] Confusion matrix. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-04-12]. Dostupné z: https://en.wikipedia.org/wiki/Confusion_matrix

[62] What is the best algorithm for static posture recognition with Kinect skeletal joints? [online]. [cit. 2019-04-14]. Dostupné z: https://stackoverflow.com/questions/34864198/what-is-the-best-algorithm-for-static-posture-recognition-with-kinect-skeletal-j

[63] Matrix Descriptor of Changes [online]. 2018 [cit. 2019-04-18]. Dostupné z: https://link.springer.com/chapter/10.1007%2F978-3-030-01449-0_2

[64] View Invariant Human Action Recognition Using Histograms of 3D Joints [online]. [cit. 2019-04-19]. Dostupné z: http://cvrc.ece.utexas.edu/Publications/Xia_HAU3D12.pdf

[65] Edge Enhanced Depth Motion Map for Dynamic Hand Gesture Recognition [online]. [cit. 2019-04-24]. Dostupné z: http://openaccess.thecvf.com/content_cvpr_workshops_2013/W12/papers/Zhang_Edge_Enhanced_Depth_2013_CVPR_paper.pdf

[66] Částečně řízené učení algoritmů strojového učení (semi-supervised learning) [online]. Brno, 2016 [cit. 2019-04-24]. Dostupné z: https://theses.cz/id/cjxceu/zaverecna_prace.pdf.Diplomovápráce.MendelovauniverzitavBrně.

[67] Background Subtraction [online]. [cit. 2019-04-25]. Dostupné z: https://docs.opencv.org/3.4.3/db/d5c/tutorial_py_bg_subtraction.html

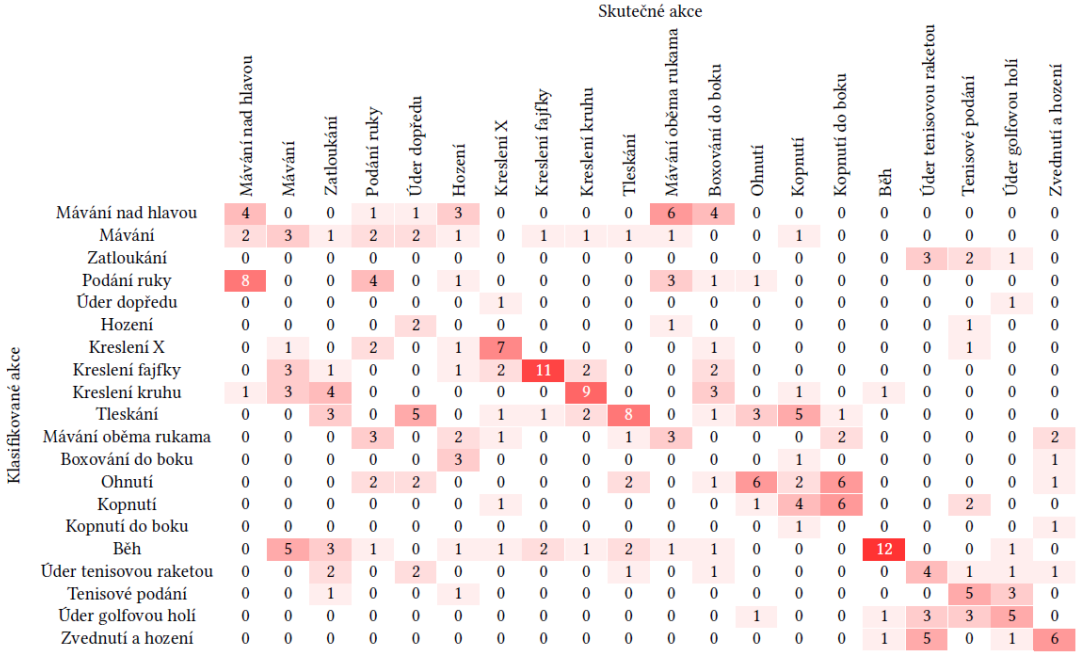
[68] Improved Adaptive Gaussian Mixture Model for Background Subtraction [online]. [cit. 2019-04-25]. Dostupné z: http://www.zoranz.net/Publications/zivkovic2004ICPR.pdf

[69] Eroding and Dilating [online]. [cit. 2019-04-26]. Dostupné z: https://docs.opencv.org/3.4.3/db/df6/tutorial_erosion_dilatation.html

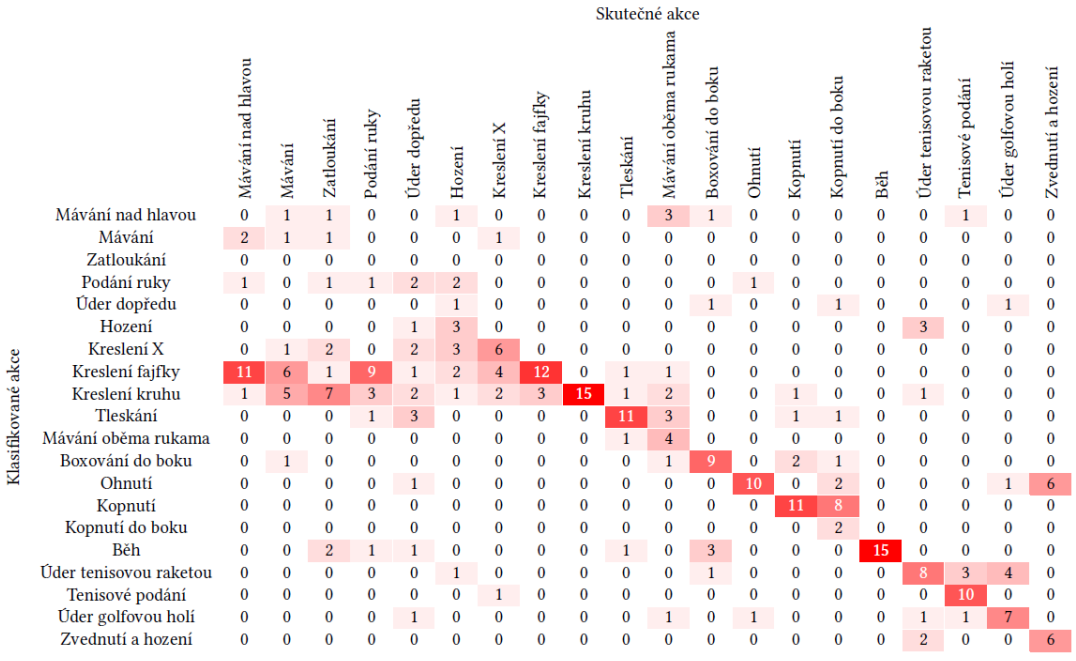
[70] A Computational Approach to Edge Detection [online]. 1986 [cit. 2019-04-27]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3300&rep=rep1&type=pdf

[71] Morfologické operace [online]. [cit. 2019-04-29]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise10/content_cz.php

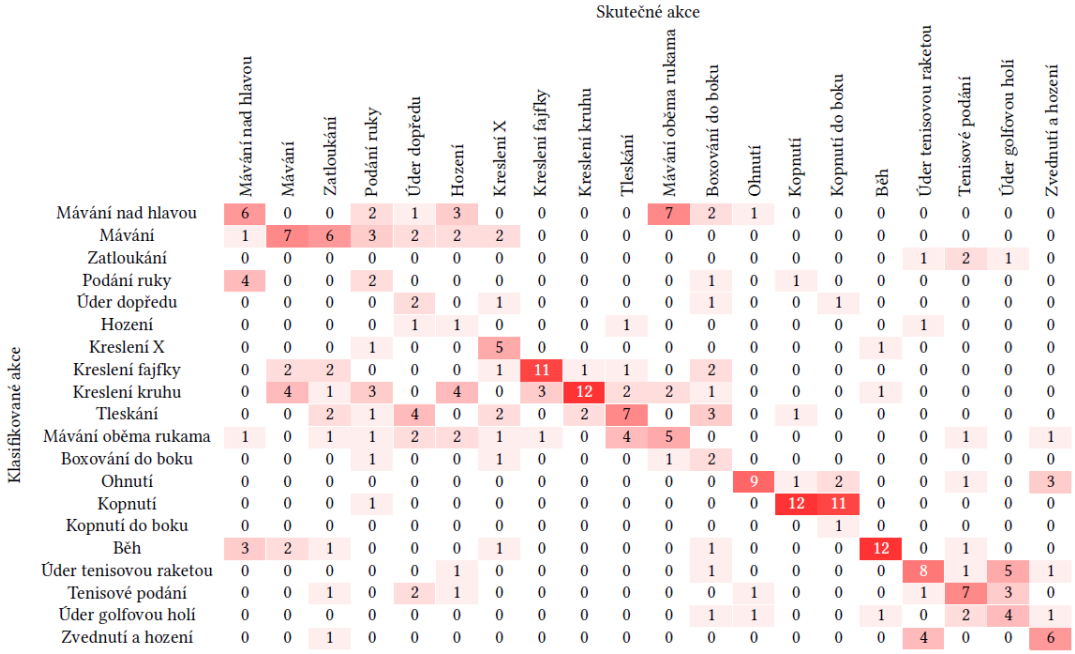
A Ukázky matic záměn



(a)

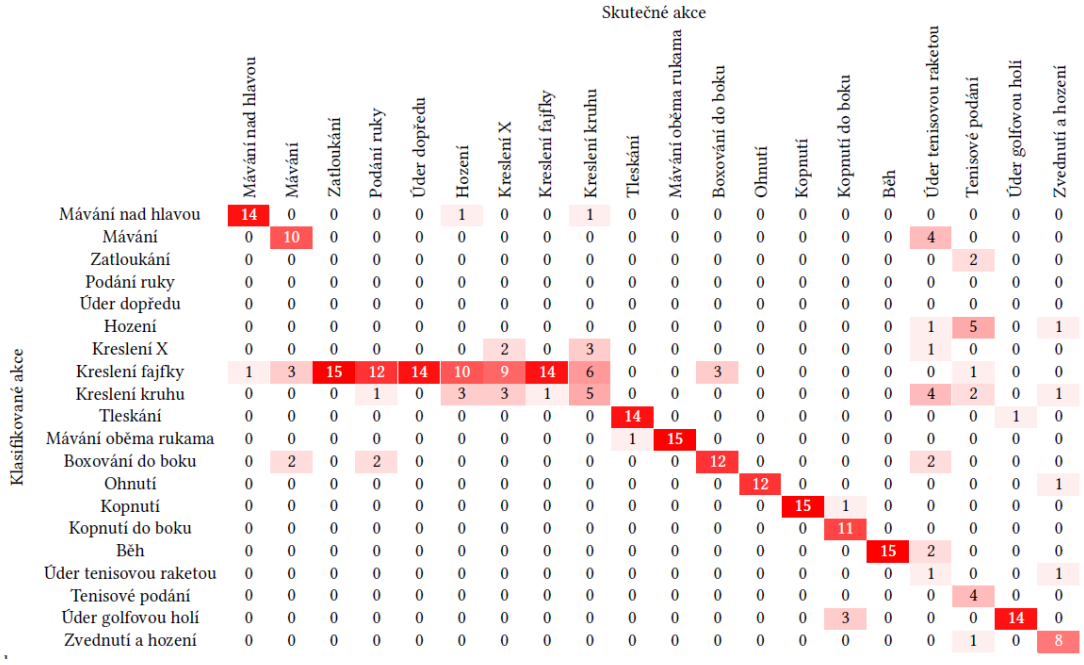


(b)

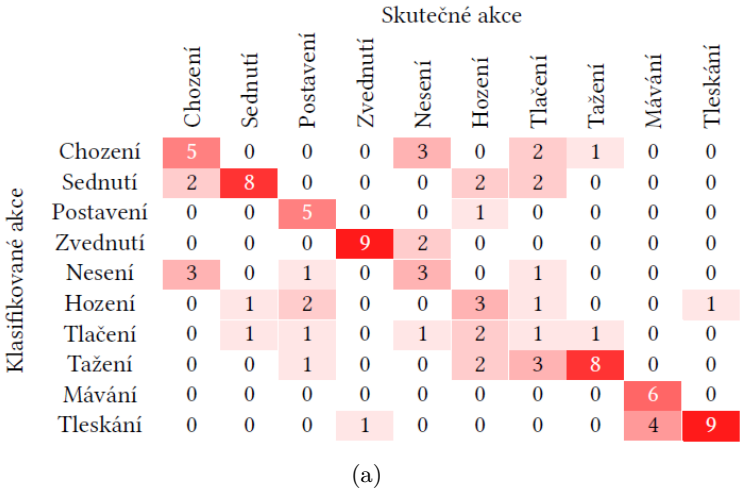


(c)

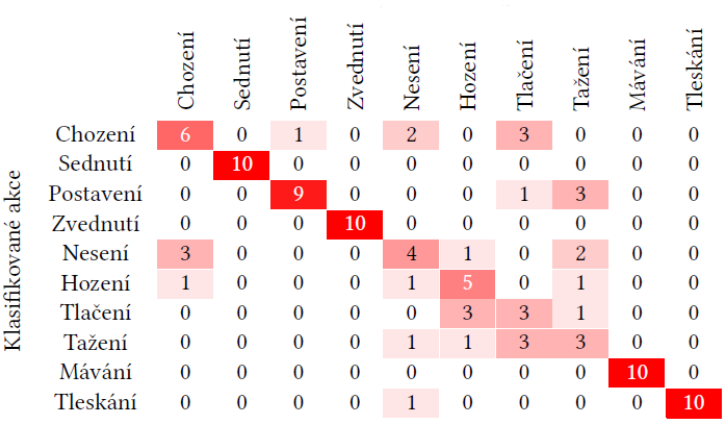
Obrázek 29: Matice záměn pro dataset MSR Action3D a metodu Temporal Self-Similarities. Na obrázku (a) je matice záměn pro SSM s výpočtem nad všemi klouby dohromady. Na obrázku (b) je matice záměn pro SSM s výpočtem pro každý bod dané akce a na obrázku (c) je matice záměn pro SSM s výpočtem pro trup s hlavou a končetiny.



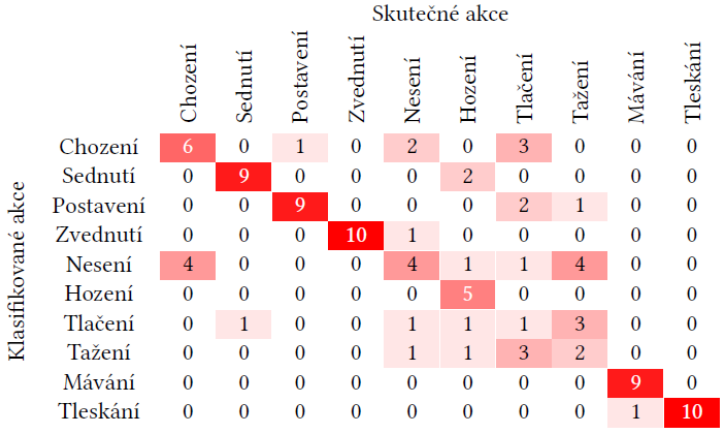
Obrázek 30: Matice záměn pro dataset MSR Action3D a metodu Depth Motion Maps.



(a)

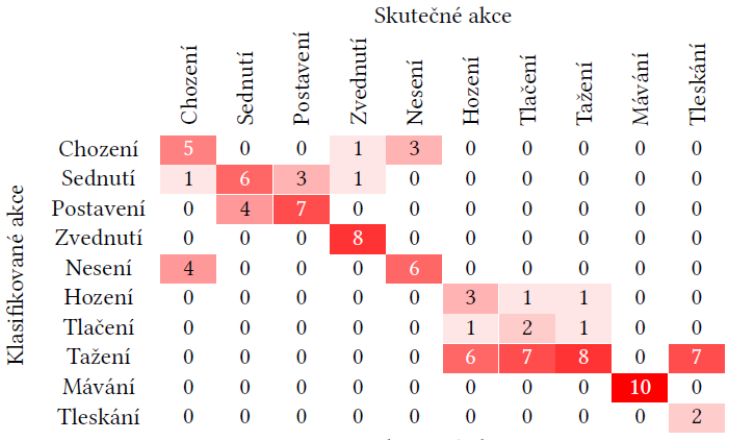


(b)

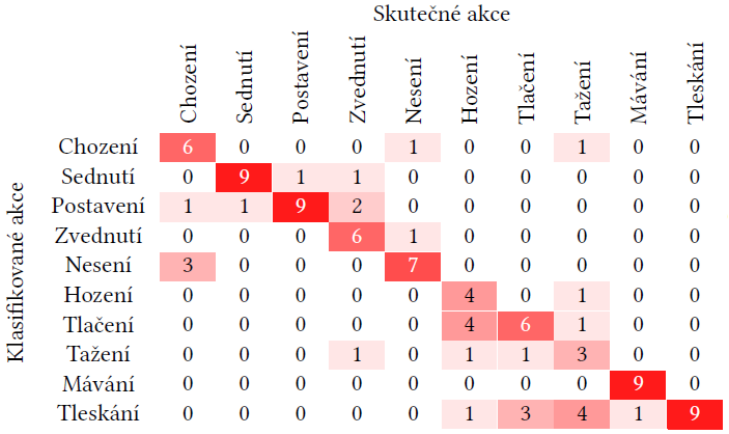


(c)

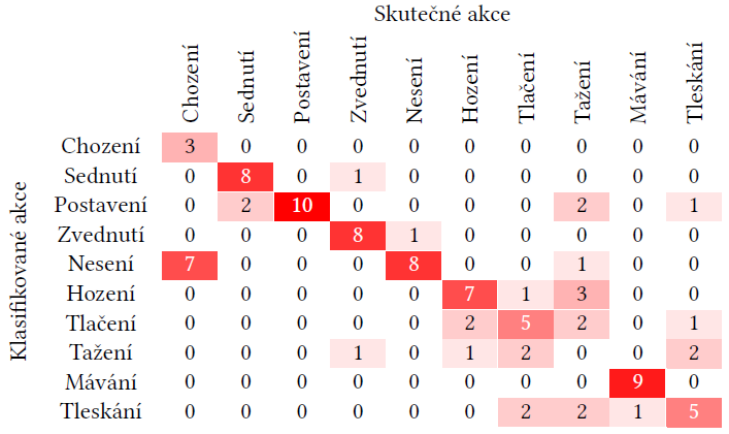
Obrázek 31: Matice záměn pro dataset UTKinect-Action3D a metodu Temporal Self-Similarities. Na obrázku (a) je matice záměn pro SSM s výpočtem nad všemi klouby dohromady. Na obrázku (b) je matice záměn pro SSM s výpočtem pro každý bod dané akce a na obrázku (c) je matice záměn pro SSM s výpočtem pro trup s hlavou a končetiny.



(a)



(b)



(c)

Obrázek 32: Matice záměn pro dataset UTKinect-Action3D a metodu Depth Motion Maps. Na obrázku (a) je matice záměn pro snímky bez odstraněného pozadí. Na obrázku (b) je matice záměn pro snímky s aplikovaným MOG2 a na obrázku (c) je matice záměn pro snímky s aplikovaným MOG2 a erozí.