


Monitorování počtu osob v budovách

Monitoring the number of people in the buildings

Bc. Martin Bobek

Diplomová práce
2010

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin BOBEK**
Osobní číslo: **A08498**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Monitorování počtu osob v budovách**

Zásady pro vypracování:

1. Zhodnoťte současný stav v oblasti běžně používaných přístupových systémů a vypracujte literární rešerši na dané téma.
2. Navrhněte vhodnou strukturu měřicího pracoviště pro monitorování počtu osob v budově.
3. Použijte vhodné algoritmy pro zpracování obrazu nezbytných pro detekci pohybu.
4. Z dostupných prostředků sestavte měřicí systém a proveďte základní testovací měření.
5. Proveďte měření v budově UTB na U5.
6. Zhodnoťte dosažené výsledky a odhadněte meze použitelnosti zařízení.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Křeček S.: Příručka zabezpečovací techniky, Cricetus, 2006
2. Čandík M.: Objektová bezpečnost II, UTB Zlín, 2004
3. Zaplatílek K., Doňar B.: MATLAB -- tvorba uživatelských aplikací, Praha, 2004
4. Dobeš M.: Zpracování obrazu a algoritmy v C, BEN, Praha 2008
5. Javidi B.: Image recognition and classification, algorithms, systems and application, Marcel Dekker, Inc., 2002
6. Cappellini V., Time-varying image processing and moving object recognition, Elsevier, 1997
7. Gonzales R., C., Woods R., E., Eddins S., L.: Digital image processing using Matlab, Prentice Hall, 2004
8. Jain A. K.: Fundamentals of digital image processing, Prentice Hall, 1988
9. Voříšek J.: Význam kontroly pohybu osob v chráněné zóně z hlediska průmyslu komerční bezpečnosti, diplomová práce UTB Zlín, 2007
10. MINTRON Enterprise Co., Ltd.. Mintron MIP 6430-2 [online]. 2009 [cit. 2010-01-21]. Dostupný z WWW: <http://www.mintron.com/HTM/New_PRODUCTS/ALPHA/image/MIP-6430/MIP-6430.pdf>

Vedoucí diplomové práce:

Ing. Milan Navrátil, Ph.D.

Ústav elektroniky a měření

Datum zadání diplomové práce:

19. února 2010

Termín odevzdání diplomové práce:

7. června 2010

Ve Zlíně dne 19. února 2010



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Předložená diplomová práce se zabývá monitorováním počtů osob v budovách pomocí kamerových systémů s využitím zpracování obrazu pomocí vhodných algoritmů. V teoretické části je popsáno základní rozdělení přístupových systémů, základy zpracování obrazové informace v počítačové technice a nejběžněji používané metody detekce pohybujících se objektů s využitím kamerových systémů. Praktická část obsahuje návrh a popis měřicího systému včetně popisu programového vybavení k vytvoření algoritmu, který je určen ke sledování pohybujících se osob v obraze. V závěrečné části se zabývám funkcí navrženého algoritmu, jeho praktickým využitím a návrhy na jeho zlepšení.

Klíčová slova: Přístupový systém, kamerový systém, detekce pohybu, algoritmus, Matlab

ABSTRACT

Submitted graduation thesis deals with monitoring the number of people in the buildings with the help of camera systems and for it intended algorithms. Theoretical part contains the basic division of access systems, basics of elaboration of visual information in computer technology and the most commonly used methods of detecting moving objects with the use of camera systems. Practical part consists of the description of the used appliances and firmware that is used to create an algorithm, which is intended for monitoring moving people on the videotape. The final part deals with the functionality of the designed algorithm, its practical usage and suggestions of its improvement.

Keywords: Access system, camera system, motion detection, algorithms, Matlab

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce Ing. Milanu Navrátilovi Ph.D za odborné vedení, připomínky, náměty a poskytnuté konzultace. Dále bych chtěl poděkovat své rodině a přátelům, kteří mi pomohli cennými radami a podporou, dostat tuto práci do konečné podoby.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

| | |
|---|-----------|
| ÚVOD | 8 |
| I TEORETICKÁ ČÁST | 9 |
| 1 PŘÍSTUPOVÉ SYSTÉMY | 10 |
| 1.1 TURNIKETY..... | 10 |
| 1.2 OPTICKÉ IDENTIFIKAČNÍ SYSTÉMY..... | 11 |
| 1.2.1 Optické rozpoznávání znaků (OCR)..... | 11 |
| 1.2.2 Optické čárové kódy..... | 12 |
| 1.3 MAGNETICKÉ SYSTÉMY..... | 14 |
| 1.4 ČIPOVÉ SYSTÉMY..... | 16 |
| 1.4.1 Kontaktní čipové systémy..... | 16 |
| 1.4.2 Bezkontaktní čipové rádiové systémy RFID..... | 18 |
| 1.5 BIOMETRICKÉ SYSTÉMY..... | 21 |
| 1.5.1 Rozpoznávání otisků prstů..... | 22 |
| 1.5.2 Rozpoznávání geometrie ruky..... | 24 |
| 1.5.3 Rozpoznávání obličeje..... | 25 |
| 1.5.4 Rozpoznávání duhovky oka..... | 27 |
| 1.5.5 Rozpoznávání sítnice oka..... | 28 |
| 1.5.6 Rozpoznávání žil na ruce..... | 28 |
| 1.5.7 Rozpoznávání hlasu..... | 29 |
| 1.5.8 Rozpoznávání písma..... | 30 |
| 1.5.9 Rozpoznávání DNA..... | 30 |
| 2 DETEKCE POHYBU ZA POMOCI KAMEROVÝCH SYSTÉMŮ | 32 |
| 2.1 ZPRACOVÁNÍ OBRAZU..... | 32 |
| 2.2 ZÁKLADNÍ METODY DETEKCE POHYBU V OBRAZE..... | 35 |
| 2.2.1 Porovnávání histogramu mezi snímky..... | 36 |
| 2.2.2 Sledování rozdílných bodů mezi snímky..... | 38 |
| 2.2.3 Další možné detekce pohybu v obraze..... | 41 |
| II PRAKTICKÁ ČÁST | 43 |
| 3 NÁSTROJE A PROSTŘEDKY VYUŽITÉ PŘI TVORBĚ APLIKACE PRO MONITOROVÁNÍ POČTU OSOB V BUDOVÁCH | 44 |
| 3.1 AXIS 221 NETWORK CAMERA..... | 45 |
| 3.1.1 Nastavení kamery..... | 45 |
| 3.2 MATLAB..... | 48 |
| 3.2.1 Image Acquisition Toolbox..... | 50 |
| 3.2.2 Image Processing Toolbox..... | 50 |
| 3.2.3 Pracovní prostředí Matlabu..... | 51 |
| 4 POPIS A FUNKCE NAVRŽENÉ APLIKACE | 53 |
| 4.1 UŽIVATELSKÉ GRAFICKÉ ROZHRAŇÍ..... | 53 |
| 4.2 HLAVNÍ ČÁST PROGRAMU..... | 55 |
| 4.3 POZNATKY Z MĚŘENÍ..... | 57 |
| 4.4 ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ A NÁVRH ÚPRAV..... | 60 |
| ZÁVĚR | 62 |

| | |
|---|-----------|
| ZÁVĚR V ANGLIČTINĚ..... | 63 |
| SEZNAM POUŽITÉ LITERATURY | 65 |
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 67 |
| SEZNAM OBRÁZKŮ..... | 68 |
| SEZNAM TABULEK | 70 |
| SEZNAM PŘÍLOH..... | 71 |

ÚVOD

Monitorování počtu osob v budovách se dá realizovat několika způsoby. Jednou z možností je využít k tomuto účelu kamerových systémů s příslušným softwarovým vybavením. Jedná se zřejmě o nejjednodušší a nejefektivnější metodu řešení daného problému.

V dnešní době jsou kamery běžnou součástí našich životů a denně se s nimi ať už vědomě, či nevědomě setkáváme. Jejich využití je různorodé a velmi rozsáhlé. Využívají se především k monitorování osob na veřejných prostranstvích, k identifikaci podezřelých a nežádoucích osob, ke kontrole dodržování silničních předpisů nebo k ochraně majetku a zdraví. Ve všech vyjmenovaných případech pouhá instalace kamerového systému nestačí. Je potřeba, aby daný systém obsahoval programy a algoritmy zabezpečující vhodné reakce pro konkrétní situace.

Kamerové systémy za posledních pár let prošly velmi razantním vývojem a nejinak je tomu v případě programového vybavení, které je schopné s kamerovými systémy spolupracovat. V bezpečnostních aplikacích se setkáme s mnoha algoritmy, které vyhodnocují obraz získaný z kamer a podle zjištěných skutečností reagují na nastalé situace. Existují algoritmy, které dokáží vyhodnotit pohyb ve sledované scéně a sledovat jej, přečíst SPZ automobilu, detekovat opuštěné zavazadlo na letišti apod.

Kamerový systém s aplikací monitorující pohyb osob v budově by mohl být využit třeba jako doplněk přístupových systémů. Další možnost využití se nabízí v rámci statistických měření. Daný systém by mohl být velmi užitečný složkám IZS, pokud by byl například nasazen v budově, která by musela být evakuována. Zasahující složky IZS by znaly přesný počet lidí, nacházející se v inkriminované budově a mohly by účinněji řídit evakuaci osob.

I. TEORETICKÁ ČÁST

1 PŘÍSTUPOVÉ SYSTÉMY

Přístupové systémy (ACS) nám umožňují efektivně a spolehlivě zajišťovat automatickou identifikaci osob a kontrolu jejich oprávnění ke vstupu do vymezených prostor. Systémy ACS se skládají z přístupového snímacího terminálu, vyhodnocovací elektroniky, softwaru a osobního identifikačního média či hesla. Jako identifikační médium slouží různé druhy karet, čipy nebo lze využít biometrických údajů. Díky systému ACS máme přehled o pohybu lidí ve vymezených prostorech, vzhledem k jejich povinnostem a oprávnění. Jednotlivé ACS systémy lze navíc mezi sebou vhodně kombinovat a doplnit o další bezpečnostní prvky, tak aby byla zajištěna požadovaná úroveň zabezpečení.

1.1 Turnikety

Turniket je zařízení, které funguje jako brána, jíž může v jednu chvíli projít pouze jeden člověk. Původně byly turnikety využívány při chovu zvířat. Instalovaly se do zvířecích ohrad, kde umožňovaly lidem průchod a zvířatům zabraňovaly v jejich opuštění. V dnešní době máme mnoho druhů turniketů s rozličnými vlastnostmi, které vymezují jejich použití. Z konstrukčního hlediska můžeme turnikety rozdělit na čtyři základní druhy:

- 1) Branky a zábrany
- 2) Otočné turnikety
- 3) Tripodové turnikety (turnikety s tříramenným otočným mechanismem)
- 4) Plnoprůchodové turnikety



Obr. 1 Tripodový turniket [12]



Obr. 2 Plnoprůchodový turniket [12]

Nejčastěji se dnes setkáváme s jednosměrnými brankovými nebo otočnými turnikety umístěnými v obchodech a supermarketech, které nevyžadují žádný poplatek nebo oprávnění k průchodu. Slouží k usměrnění pohybu lidí po prodejně a zamezují jim v opuštění prodejny jiným, než k tomu určeným východem.

Dalším typem jsou turnikety vyžadující vložení mincí nebo ověření dokladu o zaplacení. Tento systém se využívá především v místech placeného vstupu, jako jsou například areály metra, sportovní areály, koupaliště, zoologické zahrady nebo lyžařské vleky. Po vhození mincí nebo přečtení informací optickým snímačem z lístku o zaplacení je člověku povolen průchod turniketem.

Turnikety se také využívají ke kontrole oprávnění vstupu a k evidenci příchodů a odchodů. Průchod turniketem je povolen po jednoznačné identifikaci člověka a ověření jeho přístupových práv přístupovým snímacím terminálem. K identifikaci osoby je možno využít jednu z metod, které budou popsány níže.

Součástí těchto systémů je i databáze, ve které mohou být uložena data o jednotlivých uživateli, přístupová práva nebo historie. V objektech vyžadujících vysokou míru zabezpečení je vhodná dodatečná kontrola turniketů, která zabrání jejich překonání. Druhou možností je použití robustních plnopřechodových turniketů.

1.2 Optické identifikační systémy

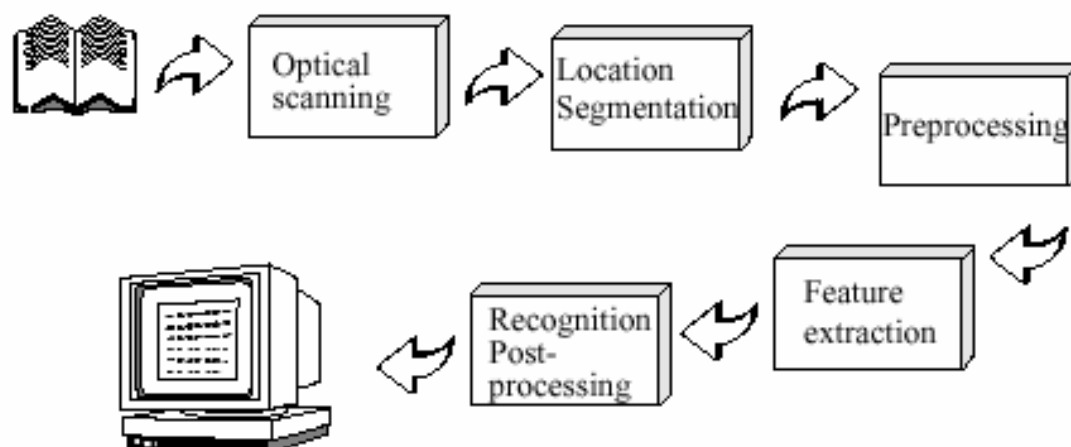
Jedná se o bezdotykovou formu identifikace a patří sem optické rozpoznávání znaků OCR (Optical Character Recognition) a čárové kódy. Osobní identifikační médium je opatřeno optickým identifikačním prvkem (písmo nebo čárový kód), který je optickými metodami přečten a vyhodnocen počítačem.

1.2.1 Optické rozpoznávání znaků (OCR)

OCR je softwarovou technologií převodu textu uloženého v bitmapovém formátu do formátu textového. OCR je speciálním případem vektorizace, tedy rozpoznávání písma. Text uložený v bitmapě není chápán jako text, ale jen jako sada tmavých a světlých bodů v obrázku. [11]

Převáděný text je snímán optickými metodami a digitalizován. Následně je provedena lokace a segmentace snímaného textu. Vhodným programem je zajištěna eliminaci šumů (tzv. preprocessing), extrakce vzhledu a porovnávání jednotlivých znaků

s předlohou. OCR systémy mají schopnost rozpoznat standardizované znaky písma (různé druhy fontů) a dokonce i individuální rukopis. Při vyhodnocování velmi záleží na kvalitě snímaného textu, jelikož program nemusí být schopen rozpoznat rozmazané nebo jinak poškozené znaky. Program prochází text několikrát za sebou a snaží se nečitelné znaky identifikovat. Součástí novějších programů je i kontrola pravopisu, díky které dokáže program doplnit písmena do neúplných slov. Po provedení oprav (tzv. postprocessingu) jsou získané informace využity k rekonstrukci originálního textu a ten je uložen v počítači pro případné další zpracování.



Obr. 3 Schéma principu činnosti systému OCR [11]

1.2.2 Optické čárové kódy

Čárový kód je z optických identifikačních systémů nejrozšířenější. V podstatě se jedná o jednoduché optické digitální paměti, které se využívají především k označení spotřebního zboží.

Každý čárový kód je tvořen sekvencí čar a mezer s definovanou šířkou. Ty jsou při čtení transformovány podle své sytosti na posloupnost elektrických impulsů různé šířky a porovnávány s tabulkou přípustných kombinací. Pokud je posloupnost v tabulce nalezena, je prohlášena za odpovídající znakový řetězec. Nositelem informace je nejenom tištěná čára, ale i mezera mezi jednotlivými dílčími čarami. Krajní skupiny čar mají specifický význam a slouží jako synchronizační pro čtecí zařízení, které podle nich generuje signál Start/Stop. Technická specifikace pak vyžaduje ochranné světlé pásmo bez potisku před a za synchronizačními čarami. [11]

Existuje skoro 200 různých druhů čárových kódů, ale všechny vycházejí ze tří základních standardů, kterými jsou:

- 1) Jednorozměrný (1D) kód
- 2) Dvourozměrný (2D) kód (různé modifikace)
- 3) Třírozměrný (3D) kód



Obr. 4 Jednorozměrný čárový kód [13]



Obr. 5 Dvourozměrný čárový kód [13]

Nejběžnější a nejvíce používaný je jednorozměrný lineární (1D) kód, kterým se značí spotřební zboží v obchodech a supermarketech. Dvourozměrné (2D) kódy jsou novější a existují v několika modifikacích (např. maticové nebo sloupcové). Tyto kódy se využívají u poštovních známek nebo ke značení plošných spojů. Nejnovější je třírozměrný (3D) kód, u něhož je třetím rozměrem barva. Data se kódují pomocí barevných oblastí různé šířky nebo kontrastem či barvou jednotlivých ploch.

| Hledisko | Čárový kód | OCR |
|-----------------------|------------|-----------|
| Kapacita dat | malá | malá |
| Čitelnost (očima) | možná | jednoduše |
| Vliv špíny, vlhka | velký | velký |
| Chybějící viditelnost | výpadek | výpadek |
| Vliv opotřebování | podmíněný | podmíněný |
| Cena nosiče dat | malá | malá |
| Cena čtečky | střední | velká |
| Manipulace s daty dat | snadná | snadná |

Tab. 1 Porovnání vlastností optických systémů [13]

Z předcházející tabulky jasně vyplývají přednosti a nedostatky jednotlivých optických systémů. Mezi přednosti optických systémů patří levné nosiče dat, standardizované techniky a velká rozšířenost těchto systémů. Hlavní nevýhoda optických systému je kapacita dat, které může datový nosič uchovat a také zpravidla jen čtecí přístup k datům. Dále je zde nutnost přímé viditelnosti mezi datovým nosičem a čtečkou. Datové nosiče jsou navíc náchylné k poškození nepříznivými vlivy okolí.

1.3 Magnetické systémy

Magnetické systémy byly zavedeny na počátku sedmdesátých let, kdy byl systém používán k realizování bankovních transakcí a identifikačním účelům. Jednalo se o magnetický proužek obsahující jednu stopu, který byl umístěn na papírových ID kartách nebo kreditních kartách a nesl informace o uživateli. Použití těchto systémů je možné pouze ve spojení s identifikátory o velikosti kreditních karet. Jiné provedení je v dnešní době nemyslitelné. Hlavní výhodou jsou malé náklady na výrobu karty. Bohužel se také dají lehce padělat a podléhají určité míře opotřebení, která je dána:

- vlastním protahováním karty snímačem (mechanické poškození)
- přítomností magnetických polí, které mohou karty trvale znehodnotit

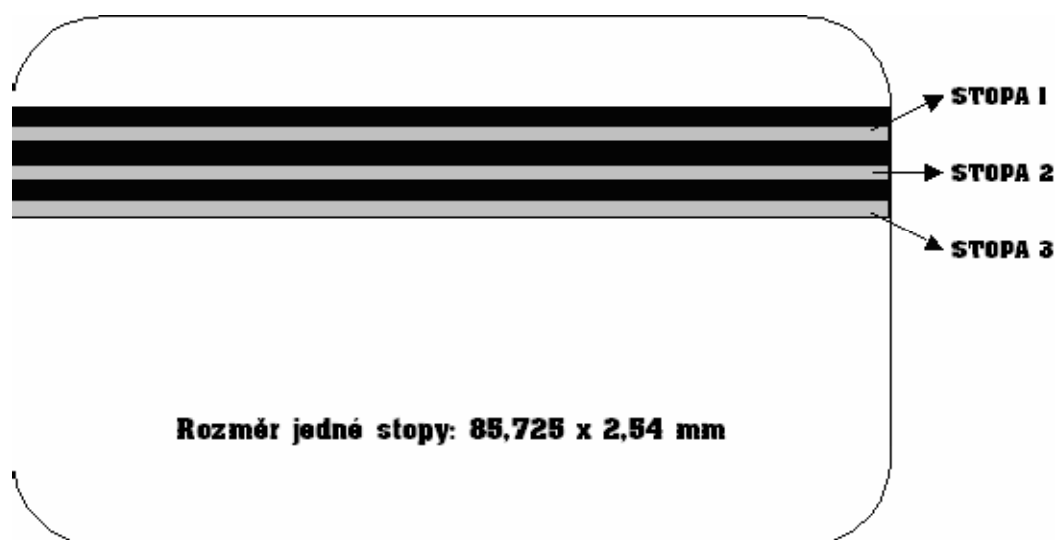
Záznam (kódování) dat do magnetického pásku se provádí zmagnetizováním, které ve vodorovné stopě vytvoří spoustu malých permanentních magnetů. Samotný záznam se na magnetický pásek provádí silnou magnetickou indukci, která působí na permanentní magnety. Je nutné, aby magnetická indukce působila jen na jedno místo, jinak by mohlo dojít k ovlivnění procesu zápisu a znehodnocení okolních dat, která jsme přepsat nechtěli. Čtení je realizované pomocí magnetických čteček, jejichž čtecí hlava přejíždí po zmagnetizovaném povrchu karty. Načtený záznam informací z magnetického proužku ID karty je převeden na elektrický signál a předán k dalšímu zpracování.

Dnes se již standardně používají karty vyrobené z plastu, jejichž magnetický proužek obsahuje tři stopy. Každá stopa má svůj význam a umožňuje uložit určité množství informací:

1.stopa - tato stopa byla definována Mezinárodní asociací leteckých dopravců IATA (International Air Transportation Association), aby usnadnila automatické odbavení cestujících, již v roce 1969. Následně tuto normu přijaly v roce 1970 i americké banky. Tato první stopa je schopna pojmout až 79 alfanumerických znaků. [14]

2.stopa - ABA (American Bankers Association) vytvořila standart pro tuto stopu, aby tím umožnila použití karet při on-line finančních transakcích, kde se používá nejvíce. Do této stopy je možné uložit 40 numerických znaků 0-9 a rovnítko. [14]

3.stopa - i tato třetí stopa byla vytvořena bankami (THRIFT) pro finanční transakce. Tato stopa se nejčastěji používá pro uložení informací, které umožňují ověřit PIN při bankovních operacích. Na rozdíl od zbývajících dvou vrstev je tato vrstva definována jako read/write, což znamená, že je možné informace uložené v této stopě přehrávat, například při odečítání kreditů, apd. Do této stopy je možné uložit až 107 numerických znaků 0-9, rovnítko a dvojtečku. [14]



Obr. 6 Rozdělení stop na magnetickém proužku ID karty [11]

Mezi nevýhody magnetických systémů patří umístění čtečky na viditelné místo, protože musí být přístupná pro vložení ID karet a může se tak stát terčem útoku vandalů. Hlavní nevýhodou magnetických systémů je nutnost při každém průchodu protáhnout kartu čtečkou, čímž se její povrch opotřebovává. Tím se také snižuje spolehlivost čtení a setkáváme se i s případy, kdy je nutné kartu několikrát znovu protáhnout čtečkou, než jsou informace z karty úspěšně načteny a ověřeny. Technologie magnetických systémů je po celém světě velmi rozšířena, ale postupně ustupuje čipové technologii, se kterou se počítá do budoucna. Postupný přechod od magnetických karet k čipovým dal vzniknout tzv. hybridním kartám. Tyto karty jsou kombinací čipové karty s magnetickým proužkem.

1.4 Čipové systémy

V současnosti se používá mnoho variant kontaktních a bezkontaktních rádiových čipových karet. Čipové karty mohou uložit mnohem více informací než karty s magnetickou stopou a mohou být přeprogramovány k uložení, smazání, nebo setřídění dat. Čipové karty byly vynalezeny v Evropě roku 1970 a v Západní Evropě se začaly používat počátkem osmdesátých let. [15] Vznikly, aby odstranily hlavní nedostatky magnetických karet, kterými byla nízká bezpečnost a občasná nespolehlivost. Obě technologie mohou být kombinovány, ale do budoucna se počítá s převahou modernějších čipových systémů.

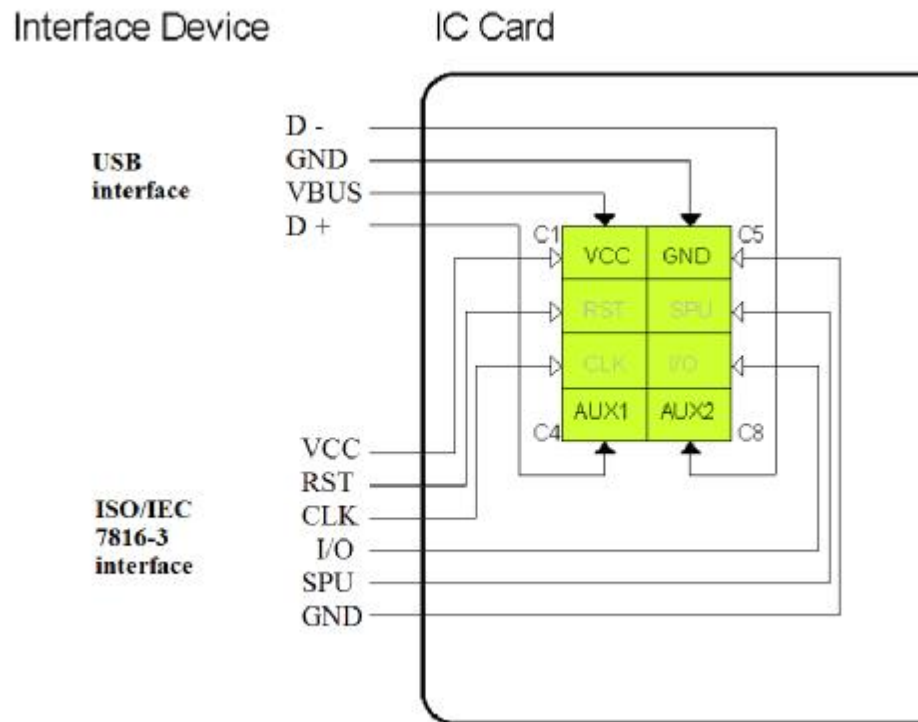
1.4.1 Kontaktní čipové systémy

Kontaktní čipová karta se svými rozměry nijak neliší od karty magnetické, ale namísto magnetického proužku je do karty zapuštěn integrovaný polovodičový čip. Tento čip může obsahovat buď pouze paměť pro uchovávání dat, nebo paměť a mikroprocesor. Čipy vybavené pouze pamětí pracují jako disketa. Jsou levnější, než karty vybavené mikroprocesorem, ale nemají tak dokonalé zabezpečení. Kvůli tomuto nedostatku je není vhodné využívat k ukládání citlivých nebo cenných informací. Čipy, které obsahují mikroprocesor, mají inteligentní řídicí obvod, který se využívá k bezpečnému přidání, odebrání či změně obsahu paměti. Dokonalejší mikroprocesory mají zabudovány pokročilejší bezpečnostní kryptografické funkce, které zamezují neautorizovanému přístupu k paměti čipu.

Aplikace čipových karet zahrnuje jejich nejčastější použití jako úvěrové nebo ATM (Automated Teller Machine) bankovní karty, SIM (Subscriber Identity Module) karty pro mobilní telefony, autorizační moduly pro placenou televizi či média pro ověření elektronického podpisu. [11] Čipová karta poskytuje služby pro identifikaci a verifikaci držitele, autentizaci uživatele informačního nebo komunikačního systému, vytvoření zaručeného elektronického podpisu a obecně pro bezpečné uložení informace. [16]

Kontaktní čipová karta má kontaktní plošku s osmi kontakty, jejichž funkce a umístění na čipové kartě je standardizováno normou ISO/IEC 7816-2. Jednotlivé kontakty slouží pro napájení čipu, sériovou komunikaci, přivedení externího taktovacího signálu a programovacího napětí. Důležité rozšíření komunikačních možností čipové karty specifikuje relativně nový standard ISO/IEC 7816-12, na jehož základě jsou již dnes vyráběny karty integrující USB rozhraní přímo na čipu, označované USB-ICC. Jejich

hlavní výhodou je možnost eliminace čtečky čipových karet, která je nahrazena standardním USB rozhraním počítače, ke kterému je připojen kontaktní adaptér obsahující čipovou kartu v SIM formátu. [16]



Obr. 7 Schéma kontaktní plošky čipové karty dle ISO/IEC 7816-12

Čipové karty se vyznačují vysokou mírou bezpečnosti uložených údajů. To je zajištěno pomocí kryptografické ochrany. Hlavní klíče, které jsou generovány a uloženy v bezpečném systému sloužícím ke správě klíčů, jsou následně implementovány na čipovou kartu pro rutinní použití. Díky programovému vybavení můžeme bezpečně šifrovat a dešifrovat pracovní šifrovací klíče, které jsou následně využity k vlastnímu šifrování nebo dešifrování dat. Pokud dojde k poškození nebo ztrátě karty, je možné hlavní klíče přenést na náhradní kartu.

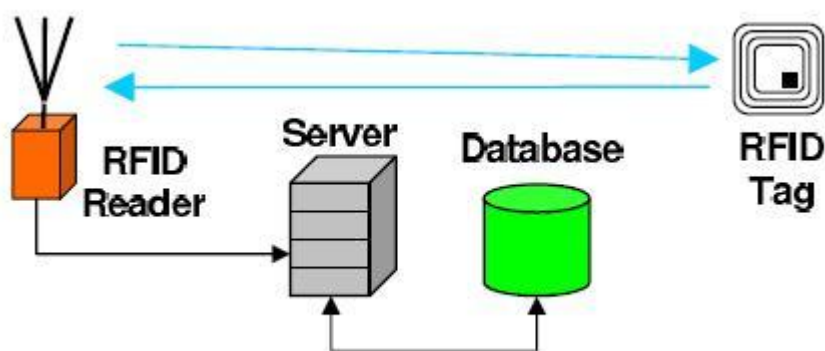
Důležitou kryptografickou operací je vytvoření zaručeného elektronického podpisu. Tento podpis umožňuje autentizovat dokumenty, ověřit jejich integritu a zaručit nepopiratelnost podepisující osoby. Na rozdíl od operací identifikace nebo autentizace osoby, které mají víceméně jednorázový charakter, je platnost elektronického podpisu permanentní. Tím naléhavější je potřeba využití bezpečného zařízení pro vytvoření (ale i

ověření) elektronického podpisu. Čipová karta s podporou asymetrické kryptografie umožňuje vygenerovat potřebné kryptografické klíče přímo na čipu a využít privátní klíč pro vytvoření elektronického podpisu, samozřejmě pouze po úspěšné verifikaci podepisující osoby. Podepisující osoba má jistotu, že bez jejího vědomí (tedy bez držení karty a současně znalosti kódu PIN) není možné elektronický podpis vytvořit. [16]

Snímání kontaktních čipových karet je zajištěno čtečkou, v níž se musí její kontakty fyzicky dotýkat povrchu čipu. Tyto karty se používají v mnoha aplikacích, jako jsou elektronické peněženky, zabezpečení přístupu do sítě, věrnostní aplikace, jídelny nebo prodejní automaty.

1.4.2 Bezkontaktní čipové rádiové systémy RFID

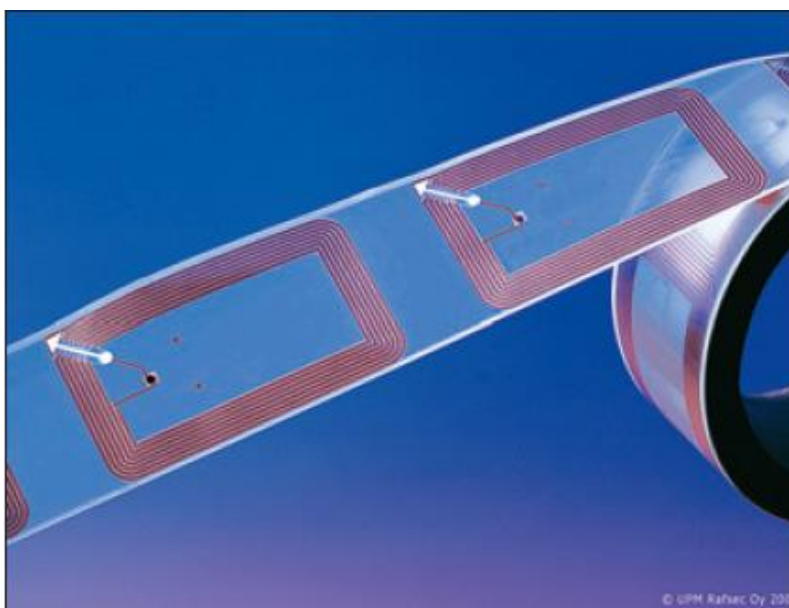
Bezkontaktní čipové rádiové systémy RFID (Radio Frequency Identification) jsou moderní technologií identifikace objektů a osob. S myšlenkou této technologie přišla největší maloobchodní firma WalMart, která stála již za vznikem optických čárových kódů. Cílem bylo vyvinout technologii, která dokáže objekt identifikovat na velkou vzdálenost, bez přímé viditelnosti tak, aby v reálném čase bylo možné zpracovat více objektů současně. RFID technologie je považována za nástupce optických čárových kódů a v současné době se tato technologie rychle rozvíjí. Uplatnění nachází hlavně v logistice, výrobě a sledování objektů nebo lidí.



Obr. 8 Schéma principu činnosti systému RFID

Systémy RFID jsou elektronické paměti (s mikroprocesorem nebo bez něj), u nichž je přístup k datům realizován bezdrátově, rádiovým elektromagnetickým polem a

příslušnými periferními zařízeními. Rádiová identifikace provádí přenos dat elektromagnetickými vlnami mezi nosičem dat a vyhodnocovací jednotkou, a to oběma směry. Systémy RFID sestávají se ze dvou základních částí: nosiče dat (transpondéru) a čtečky (případně kombinované se zapisovací jednotkou), která je vybavena příslušnými programy. Systémy RFID jsou schopny vykonávat nejen běžné identifikační rozpoznávací úkoly, nýbrž i dokumentaci při průmyslovém použití. V praxi se uplatňují různé uživatelské varianty systémů RFID, které se vyznačují odlišným uspořádáním a používanými přenosovými postupy. [13]



Obr. 9 Anténa a čip na fólii transpondéru RFID [13]

Pro využití systémů RFID jsou důležité i další vlastnosti a funkce, kterými se odlišují od optických kódovacích postupů. Významnou předností je možnost změny dat. Transpondéry typu Read and Write dovolují editovat zapsaná data a doplňovat je o další, například o informace o historii výroby, pohybu zboží. Také kapacita oproti optickým kódům je mnohem větší (dnes 32 kByte). Další výhodou RFID je možnost současného čtení informací z většího množství transpondérů (50 až 70) v dané oblasti (tzv. multitagging) Kombinovaným přístrojem lze jednotlivé transpondéry vyhledat a zaznamenat na ně potřebná data. [13]

Hlavním plusem technologie RFID je rychlost čtení a zaznamenávání dat. Vše ovšem závisí na rychlosti datového přenosu a tím pádem také na kmitočtovém rozsahu

transpodéru. Čtecí vzdálenost se pohybuje od 2 centimetrů do několika metrů v závislosti na přenosových kmitočtech a použitém napájení (aktivní nebo pasivní). Díky tomu, že mezi čtečkou a transpodérem nemusí být přímá viditelnost, lze volit téměř libovolné umístění transpodérů na identifikovaném předmětu či člověku. Velmi důležitá je také orientace antény transpodérů vůči elektromagnetickému poli, kdy je odezva úměrná úhlu mezi elektrickou složkou a anténou transpodéru. Provoz systémů RFID není téměř vůbec ovlivňován vnějšími podmínkami, jako jsou špína, vlhkost, teplota, nárazy nebo poškrábání povrchu. Výrobní náklady jsou oproti dalším technologiím vyšší, a to zejména u transpodérů. Zde se ovšem dá očekávat postupné zlevňování vlivem technologického vývoje a masové výroby. Hlavní nevýhodou technologie RFID jsou chybějící standardy, které jsou zatím ve vývojovém stádiu.

| Hledisko | Čipové karty | RFID |
|-----------------------|--------------|-----------|
| Kapacita dat | velká | velká |
| Čitelnost (očima) | nemožná | nemožná |
| Vliv špíny, vlhka | podmíněný | žádný |
| Chybějící viditelnost | - | bez vlivu |
| Vliv opotřebování | kontakt | bez vlivu |
| Cena nosiče dat | střední | velká |
| Cena čtečky | malá | malá |
| Manipulace s daty dat | nemožná | nemožná |

Tab. 2 Porovnání vlastností čipových systémů [13]

Čipové systémy dokáží uchovat velké množství informací. Jsou mnohem bezpečnější než výše zmiňované systémy a mají velmi dlouhou životnost. Vliv okolí je téměř zanedbatelný, stejně jako vliv opotřebení. Čipovým kartám se již podařilo z větší části nahradit své magnetické předchůdce, a pokud se povede podstatně vylepšit poměr ceny k užitku RFID technologie, můžeme se v blízké budoucnosti dočkat jejího masového rozšíření na úkor optických čárových kódů.

1.5 Biometrické systémy

Biometrické systémy nám slouží k jednoznačné identifikaci člověka na základě jeho fyziologických vlastností nebo projevů, kterými mohou být např. otisky prstů, duhovka, geometrie ruky, hlas, pohyby těla apod. Díky novým technologiím můžeme člověka identifikovat automaticky, rychle a spolehlivě. Biometrické systémy nabízejí výrazně vyšší míru zabezpečení než ostatní přístupové systémy, jelikož využívají charakteristik, které jsou pro každou osobu unikátní, stálé v průběhu času a není možné je odcizit nebo padělat.

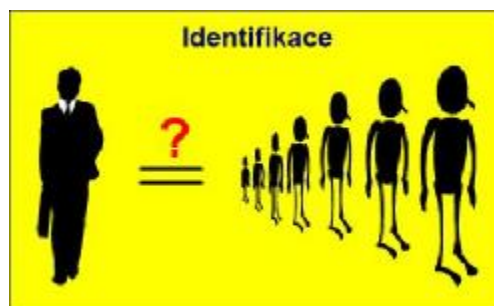
Na základě biometrické identifikace osob lze jednotlivé biometrické charakteristiky každého jedince rozdělit na anatomické (fyzické) a dynamické (chování):

- **anatomické vlastnosti** – identifikace je založena na základě vědeckých poznatků o unikátních lidských vlastnostech, které jsou časově neměnné. Mezi anatomické vlastnosti patří: otisk prstu, tvar obličeje, duhovka/sítnice oka, geometrie ruky, dlaň, žíly na rukou, dentální obraz, termogram obličeje, termogram ruky nebo DNA.
- **dynamické vlastnosti** – jsou založeny na poznacích o lidském hlase, pohybu těla či znalosti a dovednosti psaní. Tyto vlastnosti jsou sice taktéž unikátní, ale mohou se měnit např. po úraze nebo po překonání některých onemocnění. Mezi dynamické vlastnosti patří: hlas/řeč, dynamika podpisu, chůze, mimika obličeje, dynamika stisku kláves nebo pohyby rtů.

Proces identifikace za pomoci biometrických systémů může v zásadě probíhat dvojím způsobem. Rozlišujeme zde mezi verifikací a identifikací. U verifikace se předpokládá, že objekt (člověk) sám udá svou identitu. Identifikační systém sejme biometrickou charakteristiku objektu a porovná ji se šablonou, která je uložena v profilu identifikované osoby. Na tomto principu pracuje dnes většina přístupových systémů. Identifikace je na rozdíl od verifikace mnohem náročnější na výkon identifikačního systému. Při identifikaci neznáme identitu objektu (jedince) a po sejmutí biometrické charakteristiky musí identifikační systém porovnat získaný vzorek se všemi šablonami v databázi.

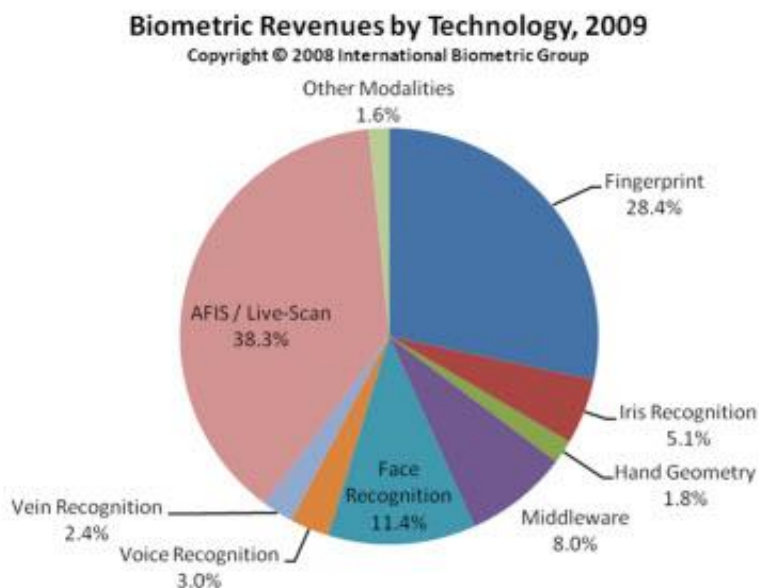


Obr. 10 Princip verifikace [11]



Obr. 11 Princip identifikace [11]

Na následujícím obrázku je vidět procentuální vyjádření výnosů z jednotlivých biometrických technologií za rok 2009.



Obr. 12 Graf výnosů jednotlivých biometrických systémů za rok 2009 [17]

V následujících kapitolách popíši nejpoužívanější biometrické technologie, které se v současné době využívají k identifikaci osob.

1.5.1 Rozpoznávání otisků prstů

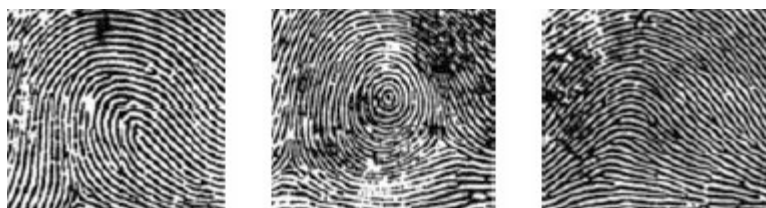
Identifikace osob na základě rozpoznávání otisků prstů je nejstarší, nejznámější a nejrozšířenější biometrická metoda. Otisky prstů jsou nejvyužívanější způsob identifikace, jak v běžném životě, tak v kriminalistice. Biometrické systémy založené na identifikaci

podle otisku prstu dokáže poměrně rychle identifikovat oprávněnou osobu na základě předem vytvořeného referenčního vzorku v databázi a umožní ji přístup do objektu, k různým systémům, službám apod. Biometrická identifikace na základě rozpoznávání otisků prstů se řadí mezi daktyloskopické identifikační metody.

Daktyloskopie je nauka, která zkoumá otisky obrazců papilárních linií na vnitřní straně prstů, na dlaních, chodidlech a o stopách, v nichž jsou papilární linie zobrazeny. [18] Obrazce papilárních linií jsou u jednotlivých osob odlišné. Zkoumání tvarů papilárních linií, vedlo k poznání tří daktyloskopických zákonů:

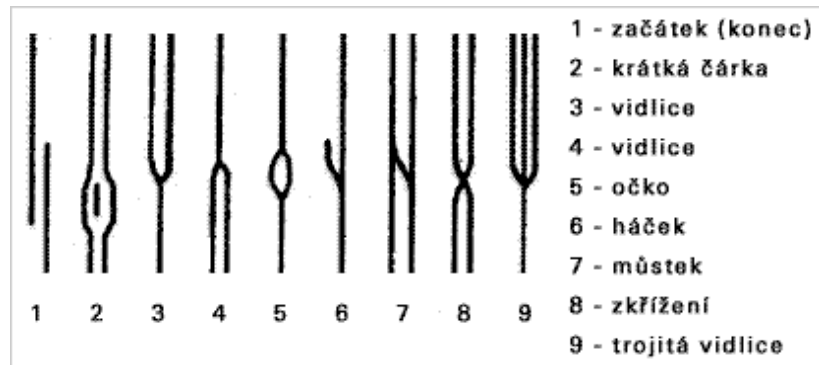
- **neopakovatelnost** – neexistují dvě osoby, které mají stejné obrazce papilárních linií
- **relativní neměnnost** – obrazce papilárních linií jsou v průběhu lidského života relativně neměnné (růst)
- **neodstranitelnost** – pokud není zničena nebo odstraněna zárodečná vrstva kůže, tak se na pokožce tvoří stále stejné obrazce papilárních linií

Vnitřní povrch prstů obsahuje vyvýšené, drobné, brázdovité útvary, které vytvářejí různé vzory. Tyto vzory se dělí do tří hlavních kategorií. Jsou to smyčky, přesleny a oblouky. Důležité je to, s jakou frekvencí se vyskytují. Například smyčky obsahuje 65% ze všech otisků, přesleny něco kolem 30% a oblouky jen asi 5% všech otisků. [19]



Obr. 13 Hlavní identifikační vzory – smyčka, přeslen, oblouk [11]

Kromě vyvýšených vzorů na vnitřním povrchu prstů, je třeba určit ještě individuální znaky jednotlivých papilárních linií tzv. markanty. Markant je jakákoliv změna v průběhu papilární linie, kterou se odlišuje od ostatních. Rozložení těchto znaků (markantů) je pro každého člověka jedinečné. Následující obrázek zobrazuje některé typy těchto znaků:

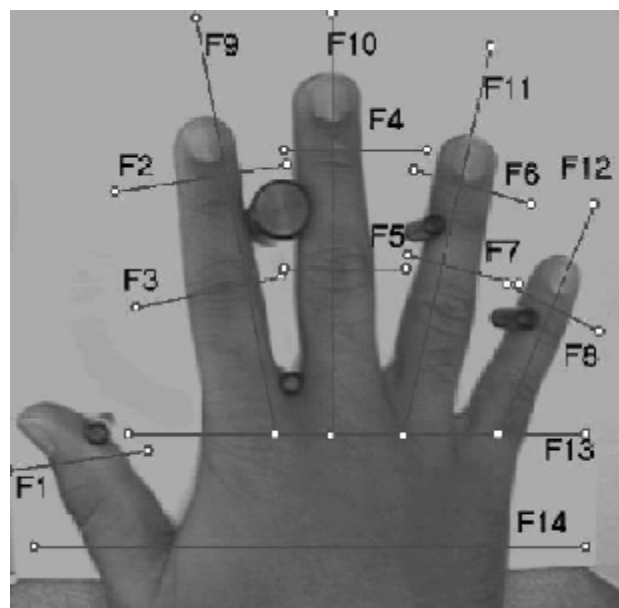


Obr. 14 Individuální znaky papilárních linií [20]

Čteček pro snímání otisků prstů, existuje na trhu celá řada. Dělí se dle použitých principů snímání, ale v praxi se nejvíce uplatňují dva typy těchto snímačů – optické a kapacitní.

1.5.2 Rozpoznávání geometrie ruky

Tato biometrická metoda je méně přesná, než třeba výše zmiňovaná metoda otisků prstů, a proto se využívá pouze k verifikaci. Způsob identifikace je založen na faktu, že specifický tvar ruky se u člověka od určitého věku nemění. Podstatou této metody je dvourozměrné nebo třírozměrné měření poměrů délek a šířek prstů, kloubů nebo kostí.



Obr. 15 Osy měření geometrie ruky [21]

Výhodou této technologie je, že na rozdíl od otisků prstů, kterých za sebou denně zanecháme stovky, siluetu celé ruky jen tak někde neotiskneme a pro potencionálního útočníka je velmi obtížné ji získat. Hodí se pro větší databáze nebo pro uživatele, kteří jsou méně disciplinovaní z hlediska užívání biometrických systémů. Systém je poměrně přesný a v praxi se úspěšně využívá již více než třicet let. Nevýhodou mohou být velké rozměry čtečky, z čehož plynou i vyšší pořizovací náklady.

1.5.3 Rozpoznávání obličeje

S rozpoznáváním obličejů se v životě setkáváme dnes a denně. Sami je totiž provádíme za pomoci našich očí a mozku při rozpoznávání druhých osob. Tyto systémy se využívají, jak pro verifikaci, tak pro identifikaci. Existuje mnoho metod rozpoznávání obličeje, přičemž je snaha o pochopení principů, parametrů a srovnávacích metod, které pro rozpoznání lidí využívá lidský mozek. Mezi nepoužívanější metody rozpoznávání obličejů patří:

- 1) 2D rozpoznávání obličeje
- 2) 3D rozpoznávání obličeje
- 3) rozpoznávání termosnímků obličeje

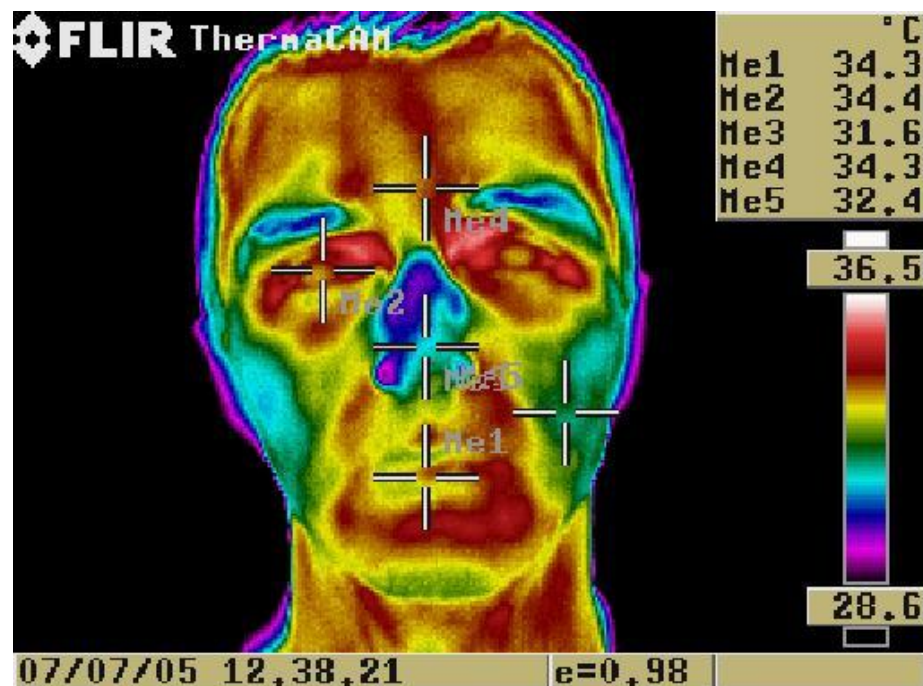


Obr. 16 Měření vzdáleností mezi specifickými body obličeje [23]

2D a 3D rozpoznávání obličejů funguje na podobném principu. Obličej osoby je snímán kamerou a sejmутý obraz je v databázi porovnáván s informacemi, které vedou

k identifikaci nebo verifikaci osoby. Obě metody jsou založeny na měření vzdáleností mezi specifickými body obličeje. *Měří se tak např. vzdálenost mezi očima, nosem, ústy a jejich velikost. Kromě toho se měří i celková velikost a tvar obličeje i jednotlivých jeho částí. Osvětlení a úhel snímání může značně ovlivnit spolehlivost této metody. Nevýhodou prvních 2D systémů bylo, že je šlo poměrně snadno oklamat např. vytištěnou fotografií. Je zřejmé, že v případě 3D systémů výše popsany útok není možné použít, neboť se zde měří i hloubka. [22]* Nevýhodou je, že může dojít k nesprávné identifikaci uživatele s podobnými rysy, jako je např. člen rodiny nebo příbuzný. Také musíme brát v úvahu možnost morfologických změn obličeje, které jsou způsobené stárnutím.

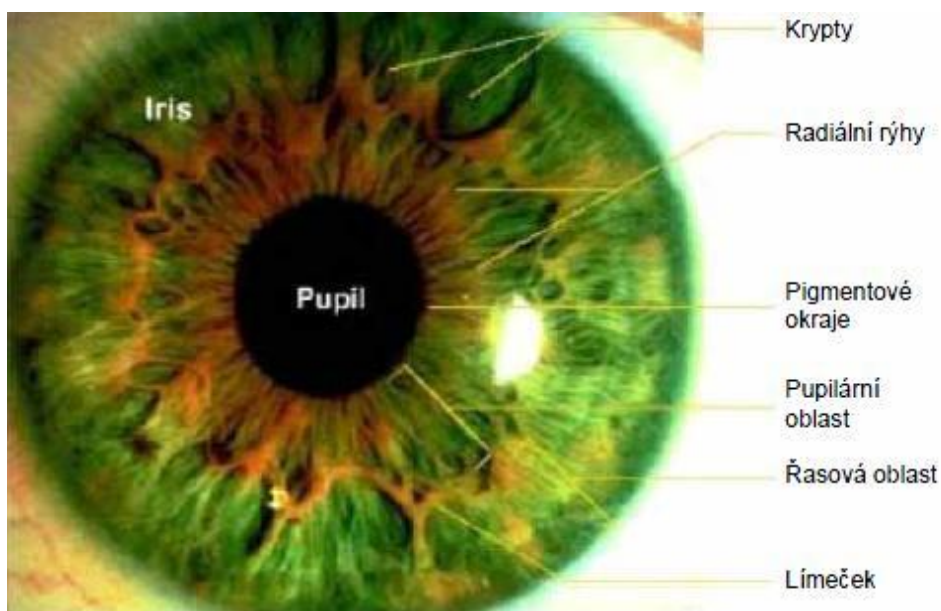
Rozpoznávání termosnímků obličeje je poměrně nová metoda biometrické identifikace. Snímek obličeje je v tomto případě pořizován za pomoci termokamery. Snímky pořízené termokamerou jsou jen obtížně zfalšovatelné. Vyrobení falzifikátu, který by dokázal oklamat bezpečnostní systém je téměř nemožné. Další výhodou, oproti klasické identifikaci obličeje je nezávislost snímku na světelných podmínkách a možnost spolehlivého získání snímku i v noci. I tato metoda je založena na měření vzdáleností mezi specifickými body obličeje, jako jsou pozice očí, nosu, úst, uší, apod. Nevýhodou je dynamicky se měnící teplota obličeje v závislosti na venkovní teplotě, fyzické zátěži, emocích apod.



Obr. 17 Rozpoznávání termosnímků obličeje [21]

1.5.4 Rozpoznávání duhovky oka

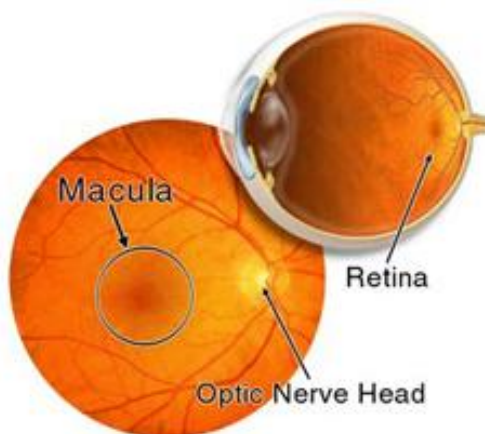
Identifikace založená na rozpoznávání duhovky oka patří mezi nejpřesnější biometrické metody. Vnější znaky (kruhy, skvrny, rýhy, koróny atd.), které duhovka obsahuje, jsou stabilizovány během prvního roku po narození a po celý zbytek života zůstávají neměnné. V první fázi identifikační procedury je pořízena černobílá digitální fotografie očí ve vysokém rozlišení. Během snímání duhovky není vyžadován fyzický kontakt mezi duhovkou a snímací kamerou. Vzor duhovky je následně systémem rozložen na malá políčka a jejich obsah převede na číselné hodnoty, které reprezentují jednotlivé stupně šedi. Políčka, která jsou světlejší než průměr, se změní na bílá a zbytek políček přejde v barvu černou. Jednotlivá bílá a černá políčka nyní představují jedničky a nuly. Celé číselné pole, reprezentující skenovanou duhovku, se následně porovná s jednotlivými vzory v databázi. Oční duhovka nabízí nejvíce rozlišovacích možností z biometrických identifikačních systémů. Pravděpodobnost nalezení dvou identických duhovek za pomoci náhodného výběru je mnohonásobně menší, než nalezení dvou identických otisků prstů. Ani obě duhovky jednoho člověka nejsou identické. Používání této biometrické metody je rychlé, pohodlné a velmi přesné. Ani fotografie duhovky oka nebo skleněné oko nemůžou přelstít tento identifikační systém.



Obr. 18 Duhovka lidského oka [21]

1.5.5 Rozpoznávání sítnice oka

Společně s rozpoznáváním duhovky se tato identifikační metoda řadí k nejpřesnějším biometrickým metodám. Stejně jako duhovka obsahuje oční sítnice velké množství specifických anatomických znaků, což zajišťuje vysokou míru identifikační přesnosti. Jelikož není sítnice viditelným orgánem, skenuje se okolí tzv. slepé skvrny, jejíž struktura je v podstatě jakýmsi unikátním “otiskem prstu”. Ke skenování se využívá infračervených paprsků, které způsobují, že na výsledném obraze jsou viditelné cévy sítnice. Tato metoda je v praxi drahá, náročná a vyskytují se také problémy technického rázu. Je vyžadován pohled do vymezeného prostoru a ostření na daný bod. Metoda je také časově náročnější a snímaný subjekt nesmí mít nasazené brýle nebo kontaktní čočky.



Obr. 19 Sítnice lidského oka [23]

1.5.6 Rozpoznávání žil na ruce

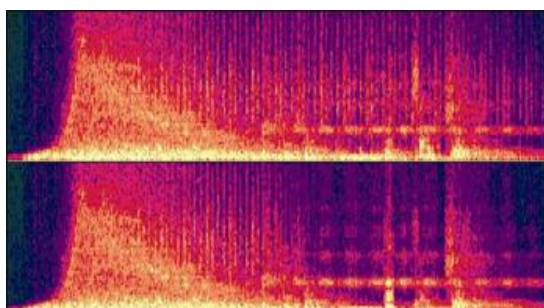
Jedná se o poměrně mladou metodu biometrické identifikace. Můžeme rozpoznávat strukturu žilního řečiště na celé ruce (dlaň, hřbet) nebo pouze na prstu, což je jedna z nejnovějších metod biometrické identifikace. Žíly ruky se vytvářejí již v prenatalním věku a jsou po zbytek života neměnné. Rozlišovací schopnost této metody je velmi vysoká a je vyšší než třeba u geometrie ruky. Metoda je velmi rychlá, nevyžaduje se přímý kontakt se snímacím zařízením a na provedení verifikace nemají vliv prsteny, kožní onemocnění nebo revma. Prozatím u metody nebyly testovány vlivy stárnutí nebo změny po proděláním srdečního infarktu.



Obr. 20 Čtečka žilního řečiště prstu [24]

1.5.7 Rozpoznávání hlasu

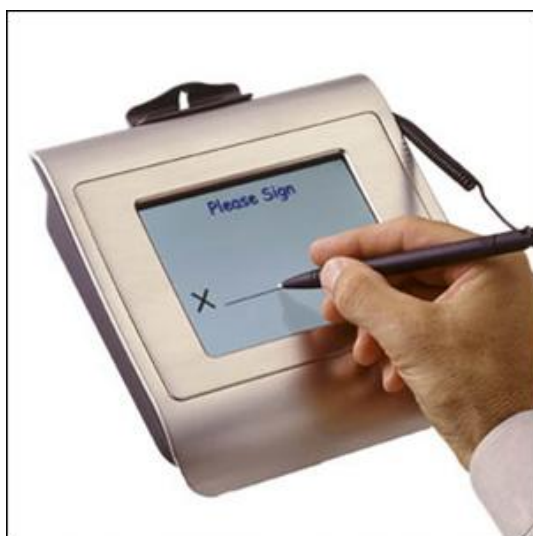
Jedná se o rychlou, spolehlivou, jednoduchou a cenově přijatelnou metodu biometrické identifikace. Mluvené slovo je pro člověka efektivním, přirozeným a pohodlným způsobem komunikace. Tato metoda vychází z elektronické analýzy hlasu identifikované osoby. Hlas každého člověka je specifický. Záleží na subjektivním vlivu osobnosti mluvčího (barva hlasu, rytmus, atd.), akustice a lingvistické struktury (gramatika a skladba řeči). Tvar zubů, dutiny ústní, potažmo hlasivek a jazyka způsobuje, že je rezonance vokálního traktu u jednotlivých osob dostatečně odlišná. Pořízený záznam hlasu se převede se na elektrický signál, který se analyzuje a porovná s hlasy osob uloženými v databázi. Nevýhodou této metody je, že verifikace může být za určitých okolností (okolní šum, nastydnutí, ztráta hlasu atd.) dost komplikovaná.



Obr. 21 Porovnání dvou zvukových signálů [23]

1.5.8 Rozpoznávání písma

Tato metoda je založena na porovnávání zrychlení v jednotlivých částech podpisu, zarovnání jednotlivých částí podpisu, celkové rychlosti, změně tlaku, dráze a době pohybu pera po papíře a nad ním. K tomu, abychom byli schopni sledovat všechny tyto parametry, je nutné, aby se osoba podepsala na speciální podložku za pomoci speciálního pera. Biometrický systém následně porovná všechny výše zmíněné charakteristiky s podpisovým vzorem. V případě verifikace moc nezáleží na podobě podpisu, ale spíše na dynamických vlastnostech podpisu. Výhodou této identifikační metody je to, že lidé jsou zvyklí se podepisovat při různých příležitostech, díky čemuž se jedná o velmi příjemnou metodu identifikace. Zařízení na snímání podpisů jsou velice přesná. V současnosti se rozpoznávání písma používá, v porovnání s ostatními biometrickými technikami, jen zřídka.

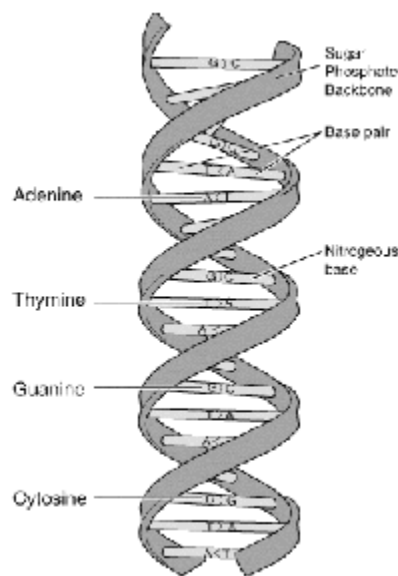


Obr. 22 Speciální podložka s perem [23]

1.5.9 Rozpoznávání DNA

Tato metoda využívá jako identifikační prvek DNA (deoxyribonukleonová kyselina), která je nositelem genetické informace. Jediná lidská buňka obsahuje přes dva metry dlouhý řetězec DNA, který je uložen v jejím jádře o průměru pět tisícín milimetru. Právě zde jsou uloženy veškeré genetické informace potřebné k vytvoření lidské bytosti. Každý člověk má jedinečnou strukturu DNA a ta se v průběhu života nemění. Výjimku tvoří

jednovaječná dvojčata, jejichž DNA je stejná. Celá metoda je velice drahá a časově náročná. Je téměř nemožné ji využít k okamžité identifikaci nebo verifikaci. I kdybychom dosáhli snížení náročnosti zpracování na přijatelnou a použitelnou hodnotu, masovějšímu rozšíření bude bránit možnost snadného získání biologického materiálu obsahující DNA, bez vědomí dané osoby. Možnost zneužití u této metody identifikace by pak byla mnohem větší, než u ostatních biometrických metod.



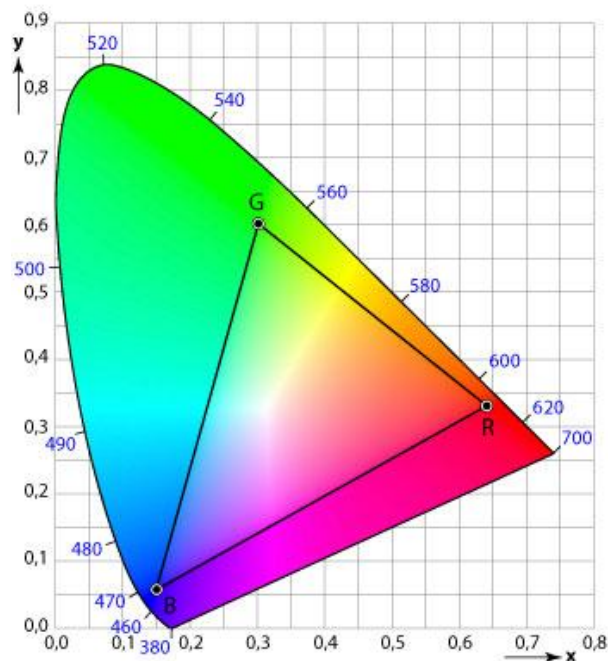
Obr. 23 Řetězec DNA [21]

2 DETEKCE POHYBU ZA POMOCI KAMEROVÝCH SYSTÉMŮ

Detekce pohybujících se objektů za pomoci kamerových systémů hraje v dnešní době významnou roli v mnoha oblastech lidské činnosti. Dnes se tyto systémy úspěšně využívají například ke sledování a rozpoznávání lidí (přístupové systémy, monitoring lidí, hlídání prostor s omezeným přístupem, hledání konkrétních osob ve videozáznamu), sledování vozidel v autodopravě (dodržování pravidel silničního provozu, hledání odcizených vozidel, výběr mýtného na zpoplatněných komunikacích) nebo ve vojenských aplikacích (detekce pohybu, monitoring cíle).

2.1 Zpracování obrazu

Obraz můžeme definovat jako spojitou dvourozměrnou funkci, $z = f(x, y)$. Jedná se o obrazovou rovnici, která nabývá hodnoty z v oboru hodnot $H(f)$. Reálná čísla x a y jsou prostorové souřadnice bodu ve dvourozměrném prostoru a amplituda f nám udává intenzitu obrazu v daném bodě. Hodnota obrazového bodu může být pouze jediné číslo nebo více hodnot v závislosti na použitém barevném modelu.



Obr. 24 Rozsah vnímání barev lidským okem proti rozsahu RGB [25]

Existuje mnoho barevných modelů, které můžeme při práci s obrazem využít. Běžně se využívá tradičního modelu RGB (výsledná barva bodu vznikne složením tří barevných složek červená, zelená a modrá), ale pro některé aplikace je výhodnější využít barevných modelů, u kterých hodnotu obrazového bodu udává pouze jedno číslo.

Kromě omezeného definičního oboru je obraz zpracováván v diskrétní podobě neboli rastru. Rastr je složen z obrazových elementů, které se v počítačové grafice nazývají pixely (picture element). Diskrétní podoby obrazové funkce $f(x, y)$ se dosáhne její digitalizací. V podstatě se jedná o převod reálného obrazu, který má teoreticky nekonečný rozsah obrazových hodnot, na konečné množství pixelů a barev. Barevná informace je zpravidla redukována na 16 miliónů barev, jelikož dokáže pokrýt prakticky celý rozpoznávací rozsah lidského oka. Další možností je převod na 256 odstínů šedi, čímž docílíme stejně kvalitní nebarevné informace. Zde je hodnota obrazového bodu udávána pouze jedním číslem, což nám při následném zpracování může ušetřit cenné systémové prostředky.

Pro lepší interpretaci získaného obrazu a snížení výpočetních nároků při dalším zpracování, využíváme při zpracování obrazu různých filtrů. Nejčastěji je používáme k redukci šumu, zaostření obrazu nebo prahování.

Šum je definován, jako nová informace, které byla přidána k původnímu obrazu. Většinou je způsobován zařízením, které obraz pořídí nebo vzniká následně, během transportu. Šumu se nikdy nedokážeme zbavit úplně, jelikož je svázán s původní informací a museli bychom mít k dispozici původní funkci. V našem případě je obraz reprezentován pixely, a proto musíme šum odstranit jinak. Jednou z možností odstranění šumu je pořízení více obrázků jednoho místa. Kamery dnes běžně dokáží pořídít 25 snímků za vteřinu, a jelikož při detekci pohybu za pomoci kamerových systémů snímáme většinou pouze jednu scénu, pořízení většího množství obrázků dané scény není problém. Redukce šumu spočívá ve vzájemném porovnávání pixelů jednotlivých obrázků, ze kterých se jako výsledná hodnota vezme buď jejich průměr, nebo v lepším případě nejčastěji se opakující hodnota. Zmíněná redukce šumu se nemůže provádět pro všechny snímky, jelikož by to bylo výkonově velice nákladné. Navíc bychom neustálým porovnáváním a průměrováním jednotlivých obrázků přišli o informaci změny, způsobenou pohybem objektu v námi sledované scéně.



Obr. 25 Obrázek obsahující šum



Obr. 26 Obrázek s odfiltrovaným šumem

Odstranění šumu může mít nepříznivý vliv na ostrost získaného obrazu. Scéna může být po aplikaci filtrů redukující šum rozmazaná, a proto je potřeba získaný obraz znovu zaostřit. Při ostření se využívá detekce a zvýraznění. Hranu si můžeme představit jako místo, kde dochází k výrazné změně dvou sousedních pixelů. Zaostření lze provádět mnoha různými metodami, přičemž nejznámější jsou založeny na použití Robertsova nebo Sobelova operátoru.



Obr. 27 Aktuální snímek



Obr. 28 Použití prahovacího filtru

Další z filtrů s oblibou využívaný v algoritmech pro detekci pohybu je prahování (thresholding). Jedná se o filtr, který získaný obraz převede do monochromatické, zpravidla černobílé reprezentace. Rozhodnutí o tom, zda výsledný pixel bude mít bílou nebo černou barvu, je stanoveno prahovou hodnotou. Pokud je intenzita jasu pixelu větší než prahová hodnota, bude mít výsledný pixel černou barvu. U nižší hodnoty bude barva pixelu bílá.

Způsob získání informace o intenzitě jasu jednotlivých pixelů záleží na použitém barevném modelu. Například tradiční model RGB je nutno nejprve převést na 256 odstínů šedi. Tento filtr hraje významnou roli u detekce pohybu, která je založena na rozdílech mezi dvěma snímky. Díky tomu můžeme při odečtení dvou snímků zjistit, kolik je mezi nimi rozdílných pixelů.

Následující filtr se velmi často využívá po aplikaci předchozího prahovacího filtru. Jedná se o filtr erosion, který pracuje pouze s monochromatickým obrazem. Využívá se pro vyplnění siluety pohybujících se objektů jednou barvou a pro odstranění náhodných osamocených pixelů, které mohou vzniknout po odečtení dvou snímků. Princip filtru je velmi jednoduchý. Jedná se o postupné procházení jednotlivých pixelů obrazu, přičemž se vezme nejmenší hodnota celého okolí (většinou okolí 3x3 pixelů) a přiřadí se jako nová hodnota danému pixelu. Tímto způsobem dokážeme zahladit osamocené pixely, které mohou být způsobeny např. šumem, a zároveň můžeme dosáhnout jednolitě vyplnění siluet pohybujících se objektů.



Obr. 29 Použití erosion filtru

2.2 Základní metody detekce pohybu v obraze

Abychom mohli ve snímané scéně sledovat pohyb, musíme dodržet určité zásady zpracování obrazu, které jsou pro správné fungování detekčních algoritmů nezbytné. Především se jedná o oddělení pozadí snímané scény od jejího popředí. Pozadí je část obrazu, na které chceme sledovat změny, a popředí v sobě zahrnuje právě tyto změny. Pozadí nebývá většinou statické a obsahuje menší i větší změny, které mohou být

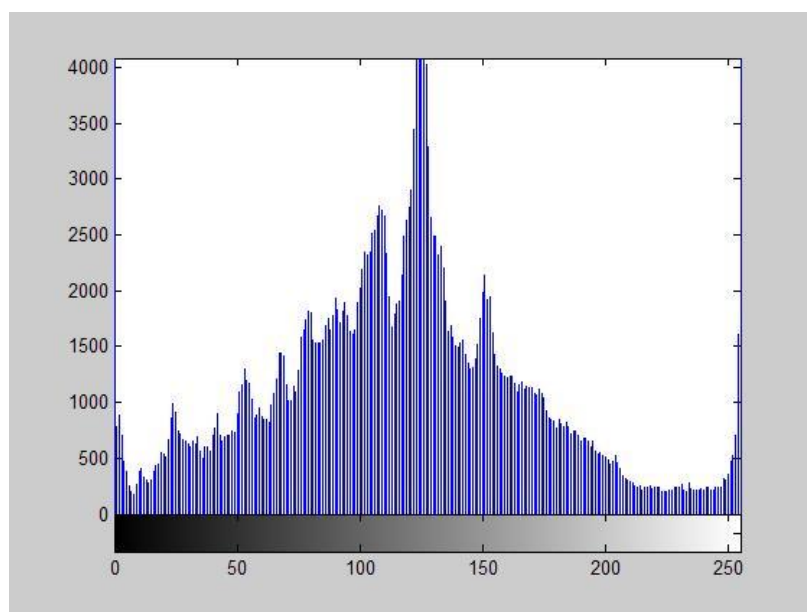
způsobeny šumem, změnou osvětlení, běžný pohyb v prostředí apod. Jedná se o změny, které musíme za pomoci vhodných filtrů co nejvíce eliminovat a přitom zachovat změny, které vytváří pohyb na popředí. Existuje mnoho metod, které dokáží řešit jednotlivé problémy. Každá metoda je jinak složitá a systémově náročná. Proto je nutné zvolit nejvýhodnější metodu pro daný systém.

V následujících kapitolách popíší základní principy fungování algoritmů, které mohou být využity pro detekci pohybu za pomoci kamerových systémů.

2.2.1 Porovnávání histogramu mezi snímky

Jedná se o nejjednodušší metodu detekce pohybu v obraze. Je založena na porovnávání histogramu (světelná charakteristika obrazu) aktuálních snímků se snímkem referenčním. Referenční snímek nám reprezentuje statickou scénu bez pohybujících se objektů neboli pozadí. Jakýkoliv pohyb ve sledované scéně způsobí, že histogram aktuálního snímku bude mít odlišnou jasovou charakteristiku.

Histogram je graf, který nám poskytuje základní informace o úrovni jasu v obraze. Na osu x je vyneseno 256 bodů, které odpovídají 256 odstínům. Na svislé ose y je znázorněn počet pixelů, které odpovídají jednotlivým úrovním jasu. Jasová hodnota se vezme přímo jako jedna ze složek barevného modelu (každá z barev R, G, B má 256 odstínů) nebo ji můžeme získat převedením obrazu na odstíny šedi.



Obr. 30 Histogram

Tato metoda je výpočetně málo náročná. Při samotném porovnávání máme referenční snímek připravený dopředu a můžeme si dovolit zbavit ho šumu i náročnějšími filtry, jelikož jde o operaci, která se provádí pouze jednou za čas. Z upraveného a šumů zbaveného referenčního snímku získáme jeho histogram, a ten je připraven k porovnávání s histogramy aktuálních snímků. Musíme ovšem počítat s tím, že bude potřeba referenční snímek po nějaké době obnovit, jelikož tato metoda je velice náchylná ke změnám v prostředí, které ne vždy dokážeme dopředu správně odhadnout a naplánovat. Správná aktualizace referenčního snímku je velmi důležitá a natolik obsáhlá, že by vydala na celou další práci. Zmíním zde pouze o základních věcech, které bychom měli brát v úvahu ve spojení s touto problematikou.

Cílem je, aby program vykonával svou funkci co nejdéle bez zásahu člověka. Je nutné popřemýšlet o podmínkách a prostorách, ve kterých budeme tuto metodu detekce pohybu provozovat. S některými negativními proměnlivými faktory se můžeme vypořádat již dopředu. V tomto případě mám na mysli např. aktualizaci referenčního snímku podle denní doby. Další negativní faktory je potřeba řešit přímo za běhu. Naším hlavním nepřítelem je šum, který se k původní informaci dodatečně přidá a stává se její nedělitelnou součástí. Jak jsem řekl již výše, můžeme tuto negativní složku alespoň potlačit některým z filtrů k tomu určeným, ale zvýší se tím výpočetní náročnost celého programu. Protože se šumu nezbavíme nikdy úplně, musíme s ním i nadále počítat. V takovém případě je vhodné stanovit hranici, nad kterou změny mezi jednotlivými histogramy budeme považovat za pohyb. Pod touto hranicí bude veškerý pohyb považován za chybu, způsobenou vlivem šumu.

Porovnávání jednotlivých histogramů probíhá tak, že od sebe odečteme histogramy referenčního a aktuálního snímku. Výsledný histogram reprezentuje odlišné pixely, které by měly představovat pohyb v popředí. V tomto histogramu sečteme počet rozdílných pixelů a porovnáme jej se stanovenou hranicí. Po jejím překročení je vyvolán alarm. Touto metodou zjistíme pohyb ve sledované scéně, ale už se nedozvíme, ve které části scény k tomuto pohybu došlo. Zde se nabízí možnost rozložit jednotlivé snímky na několik oblastí, přičemž každá z těchto oblastí bude mít svůj vlastní histogram. Díky rozumnému rozdělení oblastí v konkrétní scéně, můžeme získat docela přesné údaje o pohybujícím se objektu. Musíme si ovšem uvědomit, že nám narůstají systémové nároky, kvůli nutnosti porovnávání více histogramů.

Tato metoda je vhodná především pro aplikace typu alarm, kdy máme algoritmus v základní podobě a potřebujeme detekovat pouze pohyb ve snímané scéně. Ideální by bylo sledování prostor se stálými světelnými podmínkami, například skladiště, kde jediným neproměnlivým faktorem by byl šum. Díky tomu by odpadla nutnost aktualizovat referenční snímek a výpočetní nároky by zůstaly na nízké úrovni. V případě využití v prostorech s nestálými světelnými podmínkami nebo dalšími předvídatelnými změnami scény, bychom museli referenční snímek podle potřeby aktualizovat. Další možností je rozdělení scény na více oblastí, díky nimž můžeme získat informaci o místě, kde probíhá pohyb.

Nevýhodou této metody může být velmi silná reakce na každou změnu v případě použití referenčního snímku bez vhodných úprav. Naopak může nastat situace, kdy je referenční snímek složen průměrováním několika po sobě jdoucích snímků. Zde se můžeme setkat se slabou reakcí na pomalu se pohybující malý objekt, který se může velmi snadno stát součástí pozadí. Dalším problémem může reakce pouze na změnu poměru odstínů mezi aktuálním a referenčním snímkem. Může totiž nastat situace, kdy se objekt v obraze sice pohybuje, ale celková jasová charakteristika snímku zůstane stejná.

2.2.2 Sledování rozdílných bodů mezi snímky

Tato metoda detekce pohybu je již mnohem náročnější, jelikož pracuje se všemi pixely v referenčním i aktuálním snímku. Tato metoda je ve své podstatě velice jednoduchá. Porovnává jednotlivé pixely mezi dvěma snímky, a pokud se jejich hodnoty výrazně liší, označují tyto pixely pohybující se objekt. Z tohoto popisu fungování je jasné, že tato metoda bude při každém porovnávání snímků pracovat se všemi pixely obou snímků. To je ovšem výkonostně velmi náročné, a proto musíme nějakým způsobem tuto náročnost snížit.

Nabízí se snížení rozlišení snímaných obrázků, čímž ušetříme hodně systémových prostředků. Získané snímky můžeme také převést na odstíny šedi. Díky převodu na odstíny šedi nemusíme u jednotlivých pixelů porovnávat tři složky, jak je tomu u standardního barevného modelu RGB. Navíc ani po převodu neztratíme informace o celkovém uspořádání scény a ještě k tomu dokážeme uspořít důležité systémové prostředky. Tyto prostředky můžeme následně využít k vypořádání s šumem, které probíhá v průběhu zpracování snímků. Obrázky v následujícím textu budou znázorňovat celý průběh této metody od vstupu snímků až po vyhodnocení pohybu ve sledované scéně.



Obr. 31 Referenční snímek



Obr. 32 Aktuální snímek

Na obrázku 31 vidíme již upravený referenční snímek v odstínech šedi. Vedle něj, na obrázku 32 se nachází aktuální snímek v odstínech šedi, na kterém se nachází osoba, která na referenčním snímku nebyla. Nyní od sebe oba snímky odečteme. Výsledný snímek vidíme na obrázku 33. Z obrázku je patrné, že pixely, které se u obou snímků lišily, vytvářejí světlejší siluetu osoby, která do sledované scény vstoupila. Již nyní bychom mohli spočítat pixely, které se nacházejí nad uživatelem definovanou hranicí šumu a vyhodnotit, zda se jedná o pohyb nebo ne. Pokud ale máme dostatek výpočetního výkonu, můžeme aplikovat některé další filtry, kterými dosáhneme mnohem přesnějších výsledků.



Obr. 33 Rozdílový snímek



Obr. 34 Prahování rozdílového snímku

Jako první aplikujeme prahovací filtr, který porovná hodnoty všech pixelů ve snímku a na základě uživatelem stanovené hranice přiřadí jednotlivým pixelům bílou nebo černou barvu. Jedná se o převod do monochromatické dvoubitové podoby, ve které černě zbarvené pixely znázorňují pozadí a bíle zbarvené pixely pohybující se osobu neboli

popředí. Výsledek práce tohoto filtru můžeme vidět na obrázku 34. Na tomto obrázku si také můžeme všimnout šumu, který vytváří bílé pixely mimo siluetu sledované osoby. Tyto bílé pixely můžeme odstranit aplikací dalšího filtru, který se nazývá erosion. Na obrázku 35 vidíme výsledný obraz. Nyní spočítáme počet bílých pixelů ve výsledném snímku a porovnáme s hranicí definovanou uživatelem. Pokud je počet bílých pixelů větší než uživatelem stanovená hranice, bude výsledek vyhodnocen jako pohyb.

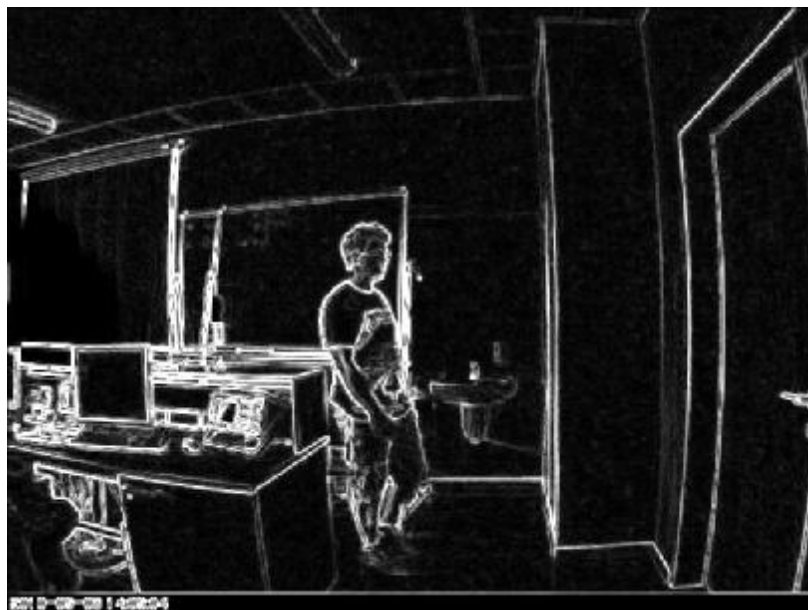


Obr. 35 Použití erosion filtru na upravený rozdílový snímek

Tato metoda dokáže přesně určit, kde ve scéně přesně k pohybu dochází. Díky nutnosti porovnávat všechny pixely v obou snímcích je tato metoda o něco náročnější na výpočetní výkon, ale plně tento nedostatek vynahrazuje přesnou vizualizací výsledků a jednoduchou implementací. Metoda si dokáže velmi dobře poradit s šumem. To je ovšem vykoupeno navýšením výpočetních nároků a zvýšením nutných zásahů uživatele. Ten musí kromě nastavení hranice, kdy detektor výslednou změnu v porovnávaných snímcích vyhodnotí jako pohyb, musí ještě nastavit všechny potřebné filtry. Jednotlivé filtry musí uživatel nastavit pro danou scénu, která bude neměnná. V případě, že se ve scéně určitých změn nedokážeme vyvarovat, musí uživatel zvolit vhodný interval aktualizace referenčního snímku. Pokud je pozadí relativně neměnné a úroveň šumu je na minimální úrovni, dokáže tato metoda rozlišit i malé, pomalu se pohybující objekty. Nemá taktéž problémy s pohybujícími se objekty, které mají stejnou barevnou charakteristiku jako referenční snímek. Celkově je tento algoritmus lepší než metoda porovnávání histogramů a její jediné negativum je malinko větší výpočetní náročnost.

2.2.3 Další možné detekce pohybu v obraze

Máme mnoho dalších metod, které můžeme využít k detekci pohybu ve sledované scéně. Následující metoda je velmi podobná výše zmíněným. Jedná se o porovnávání snímků, které jsou zpracovány filtrem detekce a zvýraznění hran. K tomu lze využít například Robertsův nebo Sobelův detektor hran. V podstatě se jedná o aplikaci speciálních masek, které dokáží v obraze detekovat a zvýraznit svislé a vodorovné hrany. Takto upravený snímek můžeme převést na histogram a použít metodu detekce pohybu s porovnávání histogramů. Nebo můžeme výsledný snímek převést na monochromatický obraz a porovnat mezi sebou jednotlivé pixely. Obě metody jsou výkonově náročnější než jejich varianty bez detekce a zvýraznění hran. Hodně záleží na tom, jak náročný filtr pro vyhledání a zvýraznění hran použijeme.



Obr. 36 Použití filtru zvýraznění hran

Následující metoda detekce pohybujících se objektů je poměrně nová. Vznikla teprve před pár lety na Finské univerzitě v Oulu. Jedná se o metodu LBP (Local Binary Patterns). Hlavní využití této metody je v rozpoznávání objektů v obraze. Jedná se o velmi rychlou a vysoce efektivní metodu detekce pohybu v obraze. Uplatňuje se i tam, kde se vyžaduje zpracování obrazu v reálném čase. Navíc má ještě další zajímavou vlastnost, a tou je tolerance ke změnám osvětlení. Díky této vlastnosti je metoda LBP schopná plně automaticky korigovat světelné změny v průběhu denní doby, ale i náhlé změny způsobené

například kolísáním proudu v žárovce nebo zastínění snímané scény mrakem. Princip předchozích algoritmů si šlo vcelku lehce představit. U LBP to ale tak jednoduché není. Jednotlivé pixely obrazu jsou porovnávány s 8 sousedními pixely a výsledná nová hodnota daného pixelu (LBP koeficient) spíše vyjadřuje jeho charakter v rámci okolí. Krajní body obrazu, které nemají kolem sebe dostatečný počet pixelů pro porovnávání, jsou vynechány nebo jim je přiřazena hodnota 0. Poté, co algoritmus tímto způsobem upraví celý obraz, nelze již jednotlivé pixely znovu zobrazit do nám srozumitelné podoby. Následně se z hodnot LBP koeficientů vytvoří LBP histogramy, se kterými se dále pracuje jako v metodě porovnávání histogramů mezi jednotlivými snímky.

Poslední metoda detekce pohybu v obraze, kterou tady zmíním je metoda tzv. optického toku, neboli optical flow. Tento algoritmus dokáže sledovat pohyb pixelu mezi jednotlivými snímky video sekvence, z čehož dokáže zjistit velikost rychlosti a směr pohybu. Tento algoritmus je výpočetně velice náročný a není možné využít ho pro všechny pixely v obraze. K tomu, aby zmíněná metoda mohla plnit funkci detektoru pohybu, nám postačí počítat optický tok pro řídkou síť vhodně rozložených pixelů, abychom co nejvíce snížili výpočetní náročnost algoritmu. V momentě, kdy by se nějaký ze sledovaných pixelů pohnul, přešel by algoritmus do plného výpočetního zatížení a začal by sledovat pohyb všech pixelů v obraze. Tato metoda se ovšem jako základ pro program detekující pohyb ve video sekvencích moc nehodí, ale lze ho k této funkci využít.

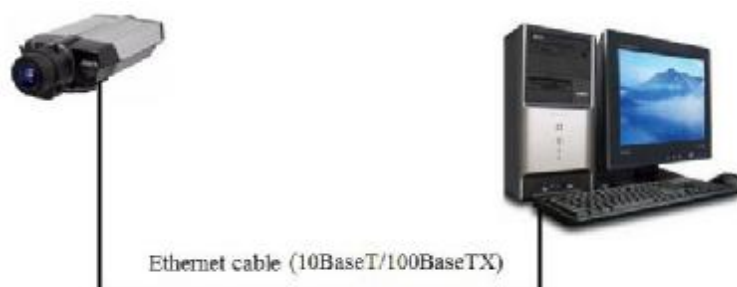
II. PRAKTICKÁ ČÁST

3 NÁSTROJE A PROSTŘEDKY VYUŽITÉ PŘI TVORBĚ APLIKACE PRO MONITOROVÁNÍ POČTU OSOB V BUDOVÁCH

K vytvoření měřicího pracoviště s aplikací pro monitorování počtu osob v budovách jsem dostal k dispozici síťovou kameru AXIS 221 Network Camera. Ke správné funkci a nastavení kamery je potřeba využít několika programů, které jsou dodávány jejím výrobcem. Všechny budou popsány v kapitole 3.1. Pro tvorbu algoritmu detekce pohybu byl využit program Matlab ve verzi R2008a. Software Matlab byl zvolen proto, že obsahuje velmi účinné nástroje pro zpracování a interpretaci obrazových informací. Posledním článkem měřicího pracoviště bude PC, které splňuje minimální požadavky pro provozování síťové kamery AXIS 221 a softwaru Matlab. Minimální požadavky na PC tedy budou:

- operační systém Windows XP (Service Pack 1, 2 nebo 3), Windows Server 2003 (Service Pack 1, nebo 2), Windows Vista (Service Pack 1)
- procesor Intel Pentium 4 nebo ekvivalent od AMD
- 510MB až 4GB místa na disku (dle zvolených součástí)
- 512MB RAM (doporučeno 1024MB RAM)
- 16-, 24-, 32-bitová OpenGL kompatibilní grafická karta

Minimální konfiguraci potřebnou ke spuštění výše zmíněných komponent, splňuje v dnešní době většina PC dostupných ve školách nebo domácnostech. Kvůli mobilitě a možnosti pracovat na algoritmu i doma jsem využil svého notebooku, který splňuje všechny výše zmíněné parametry. Na následujícím obrázku je příklad zapojení měřicího pracoviště.



Obr. 37 Schéma zapojení měřicího pracoviště

3.1 Axis 221 Network Camera

Axis 221 je součástí jedné z posledních generací síťových kamer od společnosti Axis Communications, která je založena na kompresním čipu AXIS ARTPEC-2. Jedná se o vysoce výkonnou kameru, která dokáže díky skvělé čočce Pentax, vysoce citlivému CCD čipu a automatickému odnímatelnému infračervenému filtru pořizovat velmi kvalitní snímky i za všech světelných podmínek. Díky tomu může být nasazena jak ve vnitřních prostorech, tak ve venkovních. Kamera je ideálním řešením pro zabezpečení budov a jejich okolí, parkovacích míst, nádraží nebo letišť.

Kamera dokáže poskytnout až 45 snímků za sekundu v rozlišení VGA (640x480). Obraz z kamery je k dispozici na síti v reálném čase a to v Motion JPEG formátu nebo v pokročilém video streamu MPEG-4 pro optimalizaci kvality obrazu a nároků na propustnost. Kamera také obsahuje detekci pohybu, buffery pro uložení video sekvence před vyhlášením a po vyhlášení poplachu, dva poplachové vstupy a jeden výstup pro připojení k externím zařízením, jako jsou například dveřní senzory a omezený přístup ke kameře pomocí uživatelských hesel, filtrování IP adres a HTTPS.



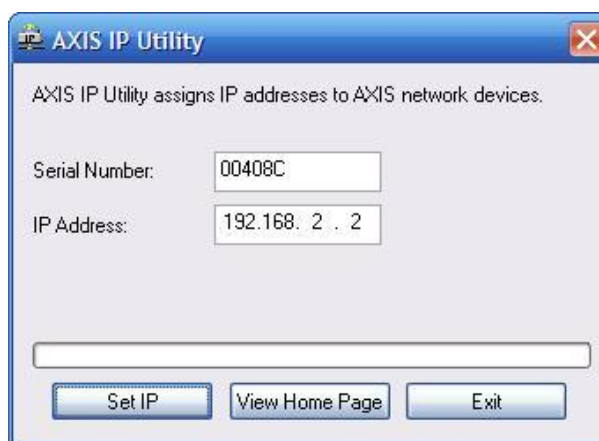
Obr. 38 Axis 221 Network Camera [10]

3.1.1 Nastavení kamery

Nejdříve bude potřeba nastavit vyhovující IP adresu. Proto musíme kameru resetovat do původního továrního nastavení. Toho docílíme odpojením napájecího kabelu, stisknutím tlačítka Reset na zadní straně kamery. Opět připojíme napájecí kabel a stále držíme tlačítko Reset po dobu asi 15 sekund. Po uvolnění tlačítka Reset se nastaví do původního továrního nastavení. Celá akce trvá přibližně minutu. Její úspěšné dokončení

indikuje zeleně svítící indikátor stavu na přední straně kamery. IP adresa kamery byla automaticky nastavena na 192.168.0.90.

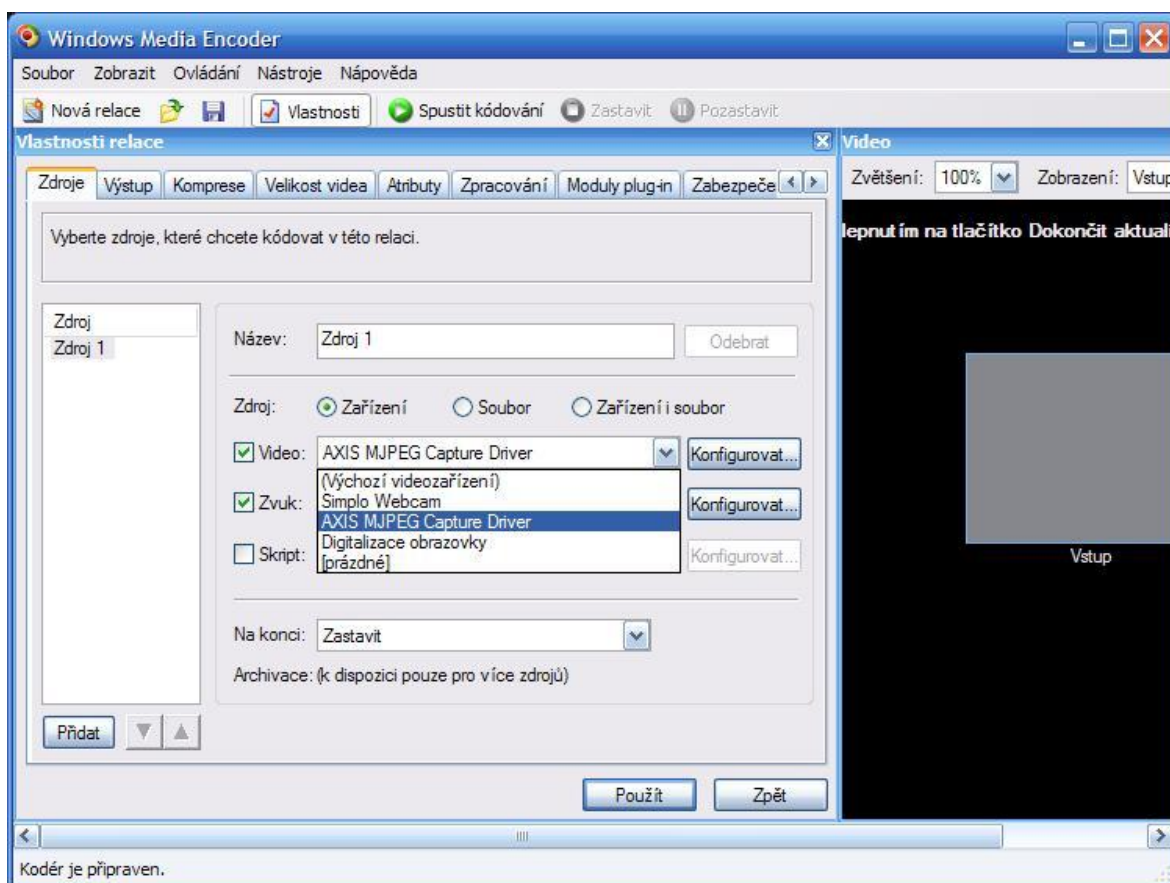
Pokud chceme tuto IP adresu změnit, použijeme prográmeček AXIS IP Utility. Zde zadáme sériové číslo kamery a námi požadovanou IP adresu. Program si kameru sám vyhledá a přiřadí ji námi požadovanou IP adresu.



Obr. 39 Axis IP Utility

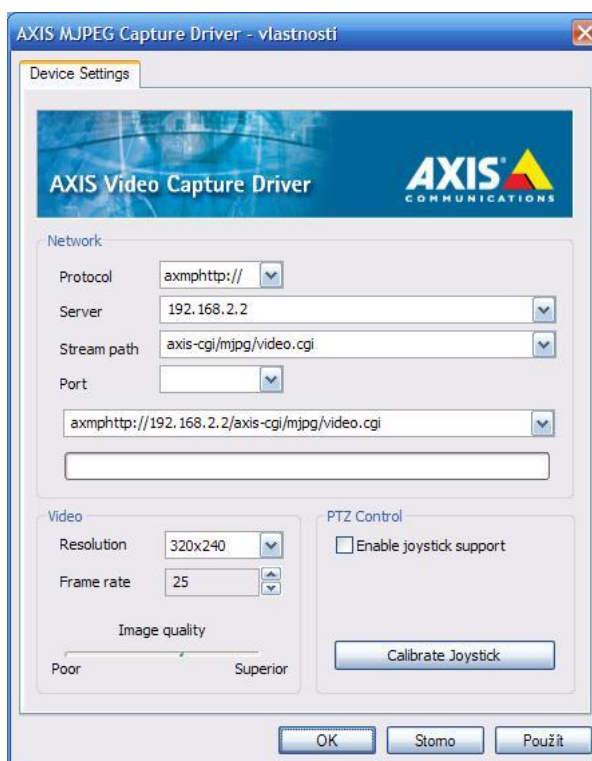
Nyní, když už máme nastavenou správnou IP adresu, se můžeme pustit do nastavování dalších parametrů. K tomuto účelu můžeme využít dvou metod. První metoda nevyžaduje instalaci žádného dalšího programu a je realizována za pomoci internetového prohlížeče (zadáním IP adresy). Oficiálně je podporován pouze prohlížeč Internet Explorer. Přístup ke kameře a jejímu nastavení funguje i za pomoci prohlížečů Mozilla Firefox a Google Chrome, Zde jsme ale ochuzeni o nadstandardní součásti ovládání, které jsou určeny pouze pro Internet Explorer. Druhá možnost spočívá v instalaci programu AXIS Camera Management, který je volně k dispozici na stránkách výrobce kamery. Díky tomuto programu máme možnost spravovat nastavení veškerých kamer od firmy AXIS, které máme připojené k PC. První přístup ke kameře vyžaduje použití internetového prohlížeče, jelikož nás přesměruje na stránky výrobce kamery, kde si zvolíme své přístupové heslo. Po úspěšném nastavení a opětovném ověření hesla jsme vpuštěni do internetové aplikace, která je určená pro sledování živého vysílání a nastavení veškerých parametrů kamery. Nastavením jednotlivých možností se tady věnovat nebudu, jelikož by to vyšlo na menší bakalářskou práci a odkáži na uživatelský manuál, kde jsou veškerá nastavení podrobně popsána.

Další důležitou věc, kterou budeme potřebovat, je instalace programu AXIS Video Capture Driver, díky níž můžeme nastavit veškeré parametry zachytávání videa. Je pro nás mnohem důležitější než předchozí aplikace, které jsou spíše určeny k nastavení živého streamu kamery. Tento program vyžaduje instalaci programu Windows Media Encoder 9 od společnosti Microsoft, který je na internetu volně k dispozici. Bez něj se k nastavení jednotlivých parametrů nedostaneme a aplikace pro monitorování počtu osob v budovách bude pracovat s defaultním nastavením. To je pro nás velice nevýhodné, jelikož velikost získaných snímků je 640x480, čímž vzrůstá výpočetní výkon a také čas, během kterého jsou jednotlivé snímky mezi sebou porovnávány. Po spuštění programu Windows Media Encoder 9 klikneme v horní nástrojové liště na ikonu Vlastnosti relace. V záložce Zdroje, vybereme u položky Video možnost AXIS MJPEG Capture Driver a zadáme Konfigurovat.



Obr. 40 Windows Media Encoder

Objeví se panel pro zprávu konfigurace zachytávání videa pomocí kamery. Jelikož nám vytvořená aplikace pracuje s rozlišením 320x240, zvolíme právě toto rozlišení. Dále nastavíme vhodný počet obrázků za vteřinu a kvalitu obrazu. Jelikož vytvořená aplikace pro detekci pohybu pracuje zhruba se 4 snímky za vteřinu, můžeme nastavit menší hodnotu, než je standardních 25 snímků. Kvalitu obrazu můžeme v průběhu měření měnit a sledovat, jak se její změna projeví v práci navržené aplikace.



Obr. 41 Axis MJPEG Capture Driver

3.2 Matlab

Matlab je vysoce výkonný nástroj pro technické výpočty. Tento systém se stal během uplynulých několika let celosvětovým standardem v oblasti technických výpočtů, počítačové simulace, měření, zpracování signálů, prezentaci dat a ve vzdělávání. Poskytuje svým uživatelům mocné výpočetní a grafické nástroje, ale také rozsáhlou databázi specializovaných knihoven, tzv. toolboxů. Další výhodou je otevřená architektura celého systému, kdy si můžeme do systému přidávat jinými uživateli vytvořené knihovny, jejichž funkce můžeme po začlenění do našeho systému používat, editovat nebo se jimi nechat

inspirovat k vytvoření vlastních funkcí. Společně s programovacím jazykem Matlabu, který je mnohem jednodušší než například Pacsal, Fortran nebo C, nám tyto vlastnosti poskytují velký potenciál a vysoce výkonný programovací nástroj, který je díky mimořádně rychlému výpočetnímu jádru určen pro početně náročné úlohy. Matlab je možné používat na všech významných platformách, kterými jsou například Windows, Linux nebo Mac.

Základem výpočetního jádra Matlabu jsou algoritmy pro operace s maticemi reálných a komplexních čísel. V nejjednodušší podobě lze tedy Matlab použít jako maticový kalkulátor. Kromě matic a jednodušších datových typů podporuje Matlab ještě mnohem složitější typy, kterými jsou například vícerozměrná pole reálných nebo komplexních čísel. Významná je také možnost práce s objekty, čímž je prostředí Matlabu rozšířeno o další datové typy, ve kterých má uživatel možnost definovat si libovolné funkce a operátory.

Ke snadné prezentaci získaných výsledků lze využít grafiku obsaženou v systému Matlab. Existuje zde mnoho možností interpretace výsledků do různých grafů (2D grafy, 3D grafy, koláčové grafy, histogramy, atd.), přičemž všem grafickým objektům lze měnit libovolně jejich vzhled. Každý vytvořený objekt má navíc svůj identifikátor, jehož prostřednictvím můžeme měnit jeho vlastnosti a tím i jeho vzhled. Navíc zde existuje možnost vytvoření ovládacích prvků (tlačítek, posuvníků atd.) pomocí grafického systému nazvaného Handle Graphic. Díky tomuto systému můžeme vytvářet aktivní graficky ovládané uživatelské rozhraní.

Matlab je úplný programovací jazyk a otevřená architektura celého systému umožňuje vytváření funkcí přesně na míru dané aplikaci. Vytvořené funkce ukládáme do tzv. m-souborů a v případě potřeby je můžeme volat stejně jako vestavěné funkce. Většina funkcí systému Matlab je uložena právě v těchto souborech, které jsou rozděleny podle oblasti využití v jednotlivých knihovnách. V jádru systému Matlab jsou vestavěny pouze funkce základní. Knihovny a funkce je možno snadno přenášet mezi různými platformami, které jsou systémem Matlab podporovány. Právě díky otevřené architektuře a možnosti tvorby vlastních funkcí a knihoven, je programovací jazyk Matlabu téměř neomezeně rozšiřitelný.

Programovací jazyk Matlabu obsahuje všechny příkazy, které jsou potřeba pro psaní programů (for, if, switch, case, while, break, otherwise, end). Programovací jazyk Matlabu

je navíc těsně propojen s jazykem JAVA. Objekty vytvořené jazykem JAVA mohou být použity programem v Matlabu, čímž se uživatelům dostává do rukou další účinný nástroj k vytváření grafických rozhraní s použitím grafických objektů JAVY. Pro jazyk JAVA je vytvořeno velké množství volně dostupných knihoven, které můžeme v Matlabu použít. K systému Matlab je možno připojit také moduly psané v jazyce C a Fortranu.

V následujících kapitolách stručně popíši dvě nejvyužívanější knihovny pro práci s obrazem a pracovní prostředí systému Matlab.

3.2.1 Image Acquisition Toolbox

Image Acquisition Toolbox slouží k získávání obrazu a videa z kamer, které lze připojit k PC. Dokáže tato zařízení automaticky detekovat, konfigurovat jejich nastavení, prezentovat nasbíraná data, získané obrázky a videa. Toolbox podporuje velké množství zařízení od levných síťových a průmyslových kamer až po vysoce výkonné kamery s nadstandardními funkcemi (vysoké rozlišení, vysoká snímková frekvence, kamery pro použití v noci, atd.).

3.2.2 Image Processing Toolbox

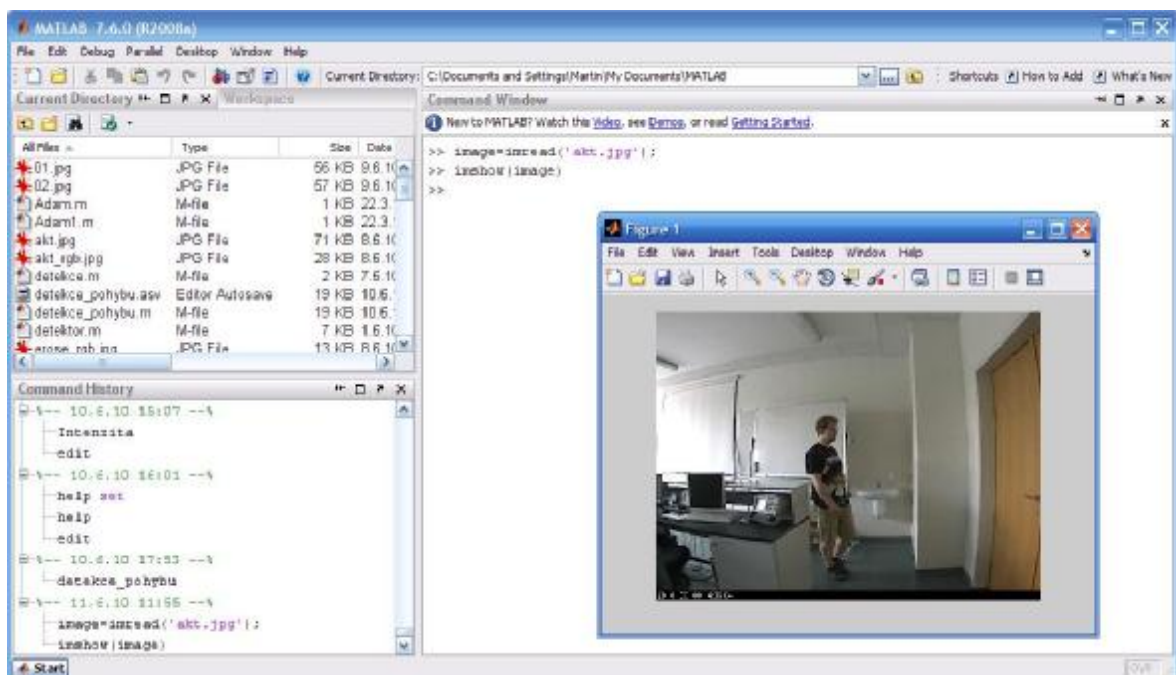
Image Processing Toolbox poskytuje komplexní sadu algoritmů a grafických nástrojů pro zpracování obrazu, jeho vizualizaci a vývoj softwaru pro práci s obrazem. Na obraz můžeme aplikovat velké množství filtrů (změna rozlišení, ostření, převod mezi jednotlivými barevnými modely, redukce šumu atd.) a velmi efektivně jej přizpůsobit našim potřebám. Získaný obraz a video můžeme vhodným způsobem zaznamenat. Při aplikaci náročných filtrů dokáže systém pracovat i s vícejádrovými procesory a počítači s více procesory.

Matlab, Image Acquisition Toolbox a Image Processing Toolbox nám dohromady poskytují komplexní prostředí pro vývoj uživatelských aplikací pracujících s obrázky a videem. Podporují velké množství obrazových formátů a video formátů, které díky nim můžeme lehce analyzovat, prezentovat nebo vyvíjet nové algoritmy pro jejich zpracování.

3.2.3 Pracovní prostředí Matlabu

Po spuštění programu Matlab se před uživatelem otevře hlavní aplikační okno, které se nazývá Matlab desktop. Součástí hlavního aplikačního okna je pět podoken:

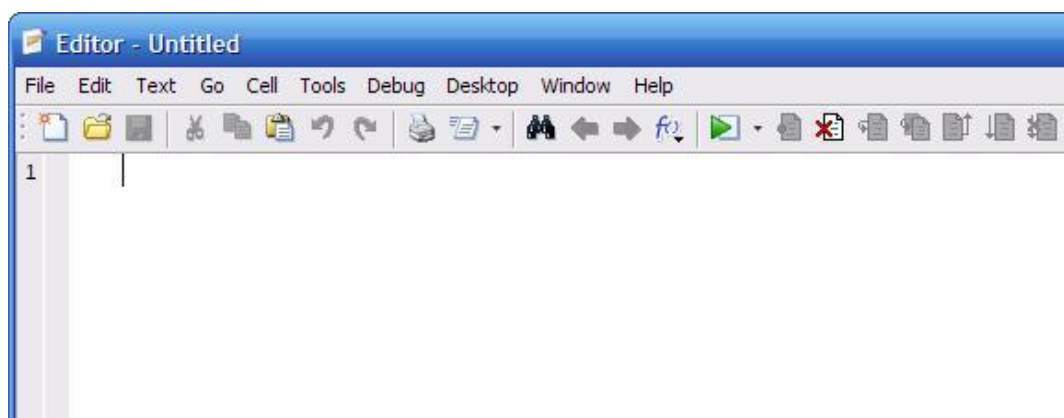
- Command Window – příkazové okno, do kterého se zadávají příkazy v jazyku Matlabu (příkazy se potvrzují klávesou Enter)
- Workspace – zde se zobrazuje obsah paměti (jednotlivé proměnné je možno editovat)
- Command History – zobrazuje historii zadaných příkazů (označení datem a časem)
- Current Directory – jedná se o prohlížeč, ve kterém jsou zobrazeny soubory a složky aktuálního adresáře
- Figure – zobrazuje se pouze při zobrazení grafiky



Obr. 42 Pracovní prostředí Matlabu

Uživatelské rozhraní je konfigurovatelné, takže si uživatel může přizpůsobit rozměry a počet zobrazených nástrojů tak, aby to maximálně vyhovovalo jeho potřebám. Je tak možné vytvořit pracovní plochu, která vyhovuje jak začátečníkům, tak pokročilým. [26]

Další důležitou součástí Matlabu je editor m-souborů. Otevírá se v samostatném okně a jde o jednoduchý textový editor. Do něj uživatel zapisuje posloupnost po sobě jdoucích příkazů a vytváří tím uživatelské skripty nebo funkce. Skripty a funkce můžeme kdykoliv spustit z hlavního okna Matlabu, zapsáním jejího názvu do okna Command Window. M-soubor musí být uložen v aktuální složce nebo ve složkách, do kterých Matlab vidí (dají se nastavit). Editor m-souborů obsahuje také tzv. debugger, který nám slouží k ladění algoritmů a odstranění případných chyb. Soubory vytvořené v tomto editoru mají svou specifickou koncovku .m (např. detektor.m).

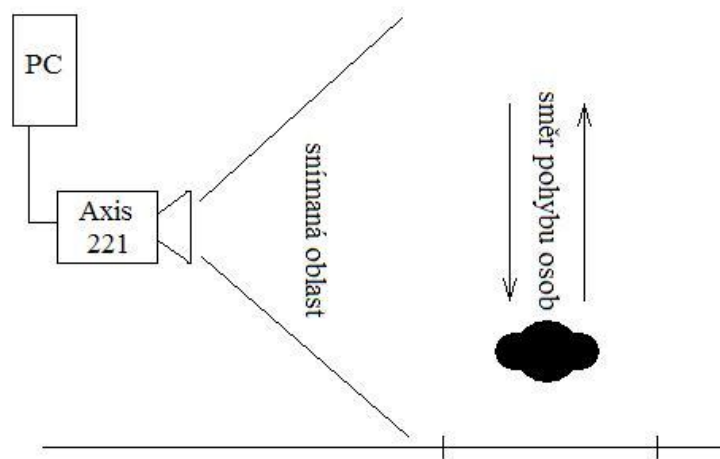


Obr. 43 Editor m-souborů

Nedílnou součástí každého programu je nápověda. Ta je v Matlabu řešena několika způsoby. Pokud známe jméno příkazu, o kterém se chceme dovědět více informací, napíšeme do Command Window příkaz help, mezera a jméno příkazu. Další možností je využití standardní nápovědy, kterou známe z většiny aplikací. Spustíme ji z lišty v hlavním okně aplikace, Help => Product Help. Kromě informací o příkazech a příkladech jejich použití obsahuje tento help i praktické ukázky s částmi zdrojových kódů a přesný popis jejich funkce. Nalezneme zde také Dema, která interaktivním způsobem popisují praktické využití některých vybraných příkazů. Poslední možností nápovědy je Online Help, který je k dispozici na stránkách výrobce systému Matlab.

4 POPIS A FUNKCE NAVRŽENÉ APLIKACE

Vytvořená aplikace se skládá ze dvou částí. V první části je definováno grafické uživatelské rozhraní, které slouží k ovládní programu. Druhá část je vyhodnocovací algoritmus detekující pohyb osob ve sledované scéně. Aplikace je uložena v m-souboru, což znamená, že je spustitelná pouze v prostředí systému Matlab. Její název je `detekce_pohybu.m` a spustíme ji tak, že napíšeme její název bez přípony `.m` do Command Window systému Matlab a stiskneme Enter. Druhou možností je otevření příslušného souboru v editoru m-souborů a spustit ho pomocí ikony **Run** (zelená šipka), která se nachází na horním panelu s ikonami. Součástí aplikace je i funkce `zjisti_okraje.m`, která je její nedílnou součástí. Oba dva soubory se musejí vyskytovat ve společném adresáři. Kamera snímá procházející lidi z boku a sleduje jejich pohyb po x-ových souřadnicích snímaného obrazu. Příklad vhodně umístěné kamery se nachází na obrázku 44. Podrobnějšímu fungování algoritmu se budu věnovat v kapitole 4.2 a jako první popíši hlavní ovládací panel celé aplikace.

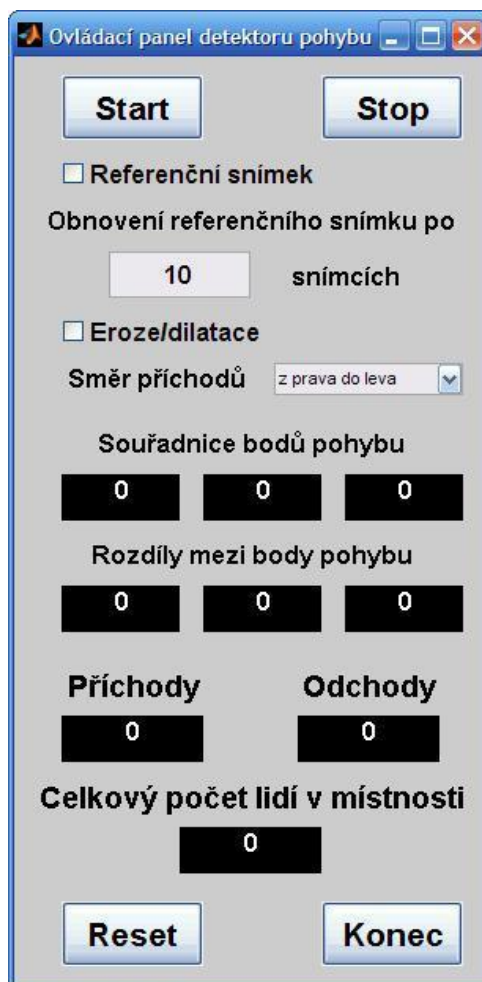


Obr. 44 Princip fungování navrženého algoritmu

4.1 Uživatelské grafické rozhraní

Po spuštění aplikace se na levé straně obrazovky objeví **Ovládací panel detektoru pohybu** (objekt Figure1). Jeho vzhled můžeme nalézt na obrázku 45. Celému panelu dominují čtyři hlavní tlačítka, která jsou umístěna v jednotlivých rozích. V horní části se

nacházejí ovládací prvky nutné k nastavení různých parametrů aplikace. Ve spodní části jsou textová pole určená ke kontrole průběhu a výpisu naměřených hodnot.



Obr. 45 Ovládací panel detektoru pohybu

Jednotlivá tlačítka vyskytující se na ovládacím panelu si nyní podrobně popíšeme:

Start – spouští hlavní vyhodnocovací algoritmus a zároveň blokuje některá nastavení, která se nedají měnit během jeho průběhu

Stop – přeruší hlavní vyhodnocovací algoritmus a odblokuje veškerá nastavení, která zablokovalo spuštění hlavního vyhodnocovacího algoritmu

Referenční snímek – pokud je políčko zaškrtnuté, bude algoritmus pracovat na principu porovnávání referenčního snímku se snímkem aktuálním

Obnovení referenčního snímku po x snímcích – umožňuje nám vhodně nastavit automatické obnovování referenčního snímku (x musí být větší nebo rovno 0)

Eroze/dilatace – zapne nebo vypne filtry eroze a dilatace (jediné nastavení, které zůstane po spuštění hlavního vyhodnocovacího algoritmu aktivní)

Směr příchodů – nastavení pro určení směru příchodů do místnosti

Reset – vynuluje všechny naměřené hodnoty a sledované veličiny

Konec – ukončí a zavře celou aplikaci

Zbylá pole nám znázorní hodnoty momentálního průběhu měření.

4.2 Hlavní část programu

Nyní se zaměřím na fungování hlavního vyhodnocovacího algoritmu, který se spustí po stisknutí tlačítka **Start**. Po jeho spuštění se definují počáteční hodnoty všech proměnných, zablokují se veškerá nastavení, kromě **Eroze/dilatace** a vytvoří se nový objekt (Figure2), ve kterém se budou zobrazovat jednotlivé snímky.

Následuje inicializace videokamery. Budeme požádáni o vložení přihlašovacího jména a hesla, které jsme nastavili v kapitole 3.1.1. Po úspěšném zadání přihlašovacích údajů, sejme kamera referenční snímek a aplikace jej zobrazí v novém okně nazvaném **Detektor pohybu** (objekt Figure2) a my do něj za pomoci osového kříže vyznačíme meze, po jejichž překročení aplikace započítá průchod osoby.



Obr. 46 Zobrazení osového kříže



Obr. 47 Vyznačení mezí

Po určení mezí se začne vykonávat cyklus while, ve kterém probíhá porovnávání snímků, veškeré výpočetní operace, vykreslování snímků a dalších grafických objektů do okna **Detektor pohybu** a odesílání vybraných hodnot do textových polí v hlavním ovládacím panelu. Cyklus se bude automaticky pořád opakovat, dokud ho nepřerušíme stisknutím tlačítka **Stop** nebo tlačítka **Konec**, čímž ukončíme i celou aplikaci.

Jako první se provede porovnávání snímků. Program má v sobě zakomponované dvě různé metody porovnávání snímků a obě jsou založeny na porovnávání rozdílných bodů mezi jednotlivými snímky. První metoda je aktivní, pokud je na hlavním ovládacím panelu zaškrtnuta volba **Referenční snímek**. Jedná se o klasickou metodu porovnávání rozdílných snímků, kdy je aktuální snímek odečten právě od snímku referenčního, který byl pořízen hned po inicializaci kamery. Referenční snímek může být navíc vhodně obnovován. To už záleží jen na podmínkách, ve kterých bude navržená aplikace pracovat. Druhá metoda je založena na porovnávání aktuálního snímku, se snímkem předcházejícím. Ta je automaticky prováděna, pokud v hlavním ovládacím panelu není povolena volba **Referenční snímek**. Ať už zvolíme jednu nebo druhou metodu, po odečtení snímků nám vznikne snímek rozdílový, se kterým budeme dále pracovat. Algoritmus ho nejprve převede do monochromatické dvoubitové podoby, a pokud je povolena volba **Eroze/dilatace**, tak provede úpravu tohoto snímku zmíněnými filtry.

Po provedení výše zmíněných procedur je v okně **Detektor pohybu** vykreslen aktuální snímek. Nyní aplikace vezme odfiltrovaný monochromatický rozdílový snímek a načte jej do pomocné funkce **zjisti_okraje.m**. Tato funkce určí krajní body pohybujícího se objektu a vrátí do hlavního programu jejich souřadnice. Do vykresleného aktuálního snímku se zakreslí hranice mezi a provedou se pomocné výpočty nutné k vykreslení ohraničení kolem pohybujícího se objektu. Pokud je v obraze zjištěn pohyb, vykreslí se ohraničení do aktuálního snímku. Ohraničení je řešeno za pomoci obdélníku, který se vytvoří kolem pohybující se osoby. Uprostřed obdélníku se nachází jeho střed. Vyhodnocování pohybu je založené právě na pohybu tohoto středového bodu po x-ových souřadnicích snímaného obrazu.

Aktuální x-ová souřadnice se uloží vždy na poslední pozici v jednorozměrném poli (pole1), které obsahuje celkem sedm prvků. Při každém provedení cyklu while, se čísla uložená v poli1 posunou o jednu pozici tak, aby další aktuální x-ová souřadnice byla na poslední pozici a nepřepsala žádnou s předchozích šesti hodnot. Nyní, když máme zjištěné souřadnice, kde se sledovaná osoba pohybuje, musíme nějakým způsobem určit směr jejího pohybu. K tomu využijeme dalšího pole (pole2), do kterého budeme ukládat rozdíly mezi jednotlivými prvky pole předchozího. Pole2 je jednorozměrné pole a obsahuje šest prvků, které získáme vzájemným odečtením dvou sousedních prvků v poli1. Na základě znaménka, které bude u jednotlivých prvků pole2 můžeme rozhodnout, zda se jedná o pohyb z prava do leva nebo z leva do prava.

Průchod bude započítán, pokud všechny prvky v poli2 budou mít stejné znaménko a poslední prvek v pole1 bude v blízkosti daného mezního bodu (ke každému meznímu bodu se přičítá nebo odčítá číslo 20, aby byl pohyb vyhodnocen ještě předtím, než sledovaná osoba protne tyto hranice). Po započítání příchodu nebo odchodu se pole1 a pole2 naplní nulami a bude proveden výpočet celkového počtu lidí v místnosti. V poslední fázi cyklu while budou do hlavního ovládacího panelu odeslány údaje o příchodech, odchodech a celkovém počtu lidí, kteří by v danou chvíli měli být v místnosti. Odešlou se také poslední tři prvky pole1 (**Souřadnice bodů pohybu**) a pole2 (**Rozdíly mezi body pohybu**), které slouží především ke kontrole.

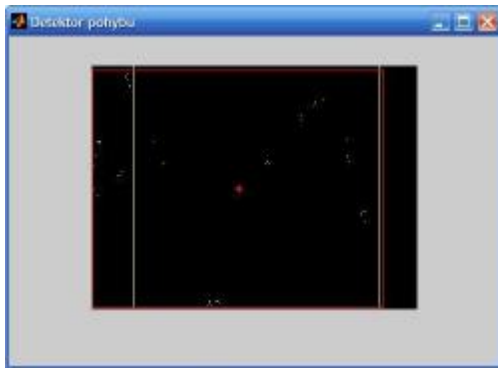
4.3 Poznatky z měření

Měření probíhalo v učebně C304 budovy U5 (FAI). Figurovalo v něm 20 studentů fakulty aplikované informatiky, kteří měli za úkol po jednom vstupovat do místnosti a poté ji zase postupně opouštět. Navrhnutý vyhodnocovací algoritmus dokáže v jednu chvíli sledovat a vyhodnocovat pohyb pouze jedné osoby. Pokud se ve snímané scéně vyskytuje pohyb na více místech, je vyhodnocování pohybu zcela nemožné. Měření bylo prováděno pro několik nastavení se zásahem do zdrojového kódu programu. Byly v něm měněny hodnoty počtu prvků v poli1 a poli2 tak, abychom porovnali funkčnost vyhodnocovacího algoritmu. Meze byly nastaveny na celou šíři obrazovky.



Obr. 48 Snímaná scéna v laboratoři C304

První nepříjemná věc, se kterou jsem se při měření setkal, byla absolutní nemožnost měření s vypnutou volbou **Eroze/dilatace**. Bez použití těchto filtrů se díky šumu ve sledované scéně vyskytoval neustálý pohyb, který narušoval průběh měření.



Obr. 49 Šum (vypnutá Eroze/dilatace)



Obr. 50 Průchod dvou osob současně

Porovnávání aktuálního snímku se snímkem referenčním (ten nebude obnovován)

| Nastavení1: pole1 (4 prvky), pole2 (3 prvky), | | |
|---|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 24 | 23 |
| Běžná chůze | 19 | 20 |
| Rychlá chůze (běh) | 13 | 12 |

Tab. 3 Naměřené hodnoty (ref. snímek – malé pole)

| Nastavení2: pole1 (10 prvků), pole2 (9 prvků), | | |
|--|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 20 | 20 |
| Běžná chůze | 19 | 20 |
| Rychlá chůze (běh) | 6 | 8 |

Tab. 4 Naměřené hodnoty (ref. snímek – středně velké pole)

| Nastavení3: pole1 (16 prvků), pole2 (15 prvků), | | |
|--|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 20 | 19 |
| Běžná chůze | 18 | 19 |
| Rychlá chůze (běh) | 0 | 0 |

Tab. 5 Naměřené hodnoty (ref. snímek – velké pole)

Porovnávání aktuálního snímku s předešlým snímkem

| Nastavení1: pole1 (4 prvky), pole2 (3 prvky), | | |
|--|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 26 | 25 |
| Běžná chůze | 19 | 20 |
| Rychlá chůze (běh) | 12 | 12 |

Tab. 6 Naměřené hodnoty (bez ref. snímku – malé pole)

| Nastavení2: pole1 (10 prvků), pole2 (9 prvků), | | |
|---|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 20 | 19 |
| Běžná chůze | 18 | 19 |
| Rychlá chůze (běh) | 7 | 6 |

Tab. 7 Naměřené hodnoty (bez ref. snímku – středně velké pole)

| Nastavení3: pole1 (16 prvků), pole2 (15 prvků), | | |
|--|----------|---------|
| Typ chůze | Příchody | Odchody |
| Pomalá chůze | 19 | 20 |
| Běžná chůze | 17 | 19 |
| Rychlá chůze (běh) | 0 | 0 |

Tab. 8 Naměřené hodnoty (bez ref. snímku – velké pole)



Obr. 51 Měřící pracoviště

4.4 Zhodnocení dosažených výsledků a návrh úprav

Z naměřených hodnot vyplývá, že obě dvě metody porovnávání jsou přibližně stejně spolehlivé. Z měření vyplývá, že ideální velikost polí pro monitorování počtu osob v laboratoři C304 by byla asi stejná nebo o něco menší než je v případě **Nastavení2**. S tímto nastavením pracoval vyhodnocovací algoritmus nejlépe a nejspolehlivěji.

Nastavení1 mělo pole pro vyhodnocování příliš malá. Proto jsme se při opravdu pomalé chůzi dobrali výsledků nad 20 příchodů a odchodů. Po překročení meze se obě pole sice vynulovala, ale protože vyhodnocovací algoritmus počítá s překročením dané meze na ose x o 20 bodů dříve, může se stát, že po překročení této hodnoty se obě pole vlivem extrémně pomalé chůze ještě jednou naplní a započítá se jeden člověk navíc. Běžnou rychlost chůze dokázal algoritmus s tímto nastavením vyhodnotit docela dobře. Za nezapočítané osoby mohou v tomto případě odlesky z lesklých ploch a hra stínů, čímž se zaznamenal pohyb v jiné části snímané scény a data v polích neodpovídala podmínkám k vyhodnocení průchodu. Nejhůře dopadlo vyhodnocení rychlého pohybu (běhu). Na jeho výsledcích se podepsaly velkou mírou dva nepříznivé faktory. Prvním byla již výše zmíněná hra stínů a odlesky lesklých ploch. Druhým faktorem je rychlý pohyb sám o sobě. Jak jsem již napsal dříve, algoritmus vyhodnotí průchod, když střed sledovaného objektu protne určitou mez. Ta byla v tomto případě vzdálená 20 bodů od okraje. Kvůli nízké snímkovací frekvenci (cca 4 snímky za vteřinu) se sledovaný bod nemusí do této hranice vůbec dostat.

Nastavení2 mělo takřka ideálně velká pole pro vyhodnocování. Pomalá chůze a běžná chůze byla správně vyhodnocována jen s malými odchylkami, způsobenými výše zmíněnými problémy se stíny a odlesky lesklých ploch. Rychlá chůze (běh) trpí stejnými problémy jako v předchozím případě a navíc se k nim přidává další problém. Pole pro vyhodnocování jsou již docela velká a při snímkové frekvenci 4 snímků za vteřinu se nemusí stihnout ani všechna naplnit, díky čemuž algoritmus průchod nezapočítá.

Nastavení3 mělo pole tvořena největším počtem prvků pro vyhodnocování. Pomalá a běžná chůze byla i zde vyhodnocována relativně správně. Průchody a odchody při rychlé chůzi (běhu) byly kvůli velikosti polí již neměřitelné. Jednotlivá pole se nestačila během pohybu osoby naplnit údaji, a proto nebylo možné průchody vyhodnotit.

Další nastavení jsem již netestoval, protože pro ověření funkce detekce pohybu a vyhodnocovacího algoritmu stačila tato měření. Z naměřených výsledků mohu odhadovat, že běžná chůze by při zvyšování počtu prvků ve vyhodnocovacích polích dopadla stejně jako rychlá chůze (běh) v **Nastavení3**.

Z měření dále vyplynulo, že je velmi důležitá šířka scény, v níž sledujeme pohyb a zaznamenáváme průchody. Kdyby námi sledovaná scéna v laboratoři C304 byla širší, bylo by možné započítat průchody i u rychlého pohybu (běhu) v **Nastavení3**.

O tom, že při měření musí být zapnuta volba **Eroze/dilatace** jsem se již zmínil. Další postřeh o problémech se stíny a odrazy lesklých povrchů zde byl také již zmíněn. Poslední připomínka se týká metody detekce porovnáváním aktuálního snímku se snímkem předchozím. Pokud se sledovaný objekt zastaví ve sledované scéně, tak se po chvíli stane součástí pozadí, dokud se opět nepohne. U metody s referenčním snímkem toto nehrozí, ale můžeme se u ní setkat s jinou

ZÁVĚR

Tato diplomová práce měla za cíl seznámit čtenáře s různými druhy přístupových systémů, počítačovým zpracováním obrazu a metodami pro detekci pohybujících se objektů ve video sekvenci. Na základě znalostí dané problematiky byla vytvořena výuková aplikace umožňující detekci, sledování a vyhodnocení pohybu osob ve sledované scéně. Aplikace byla úspěšně odzkoušena v laboratoři a její funkčnost byla potvrzena provedeným měřením.

Navrhnutá aplikace je plně funkční, ale musíme počítat s určitým omezením jejího použití. Pro laboratorní využití, ověření principu fungování detekce pohybujících se objektů, sledování jejich pohybu a vyhodnocování průchodů je plně dostačující. Pro laboratorní účely by bylo potřeba zobecnit práci s poli, aby nebylo nutné zasahovat do zdrojového kódu. V hlavním ovládacím panelu by měla přibýt volba, díky které by bylo možné měnit velikost vyhodnocovacích polí. Možností jak celou aplikaci vylepšit je více. Rozhodně doporučuji v práci na zlepšování aplikace dále pokračovat.

Pro využití v praxi by musel algoritmus projít většími změnami. Předně by musel být více optimalizován, jelikož rychlost snímkování pohybující se kolem 4 snímků za vteřinu nemusí být pro některé účely plně dostačující. Potvrdila to i laboratorní měření a pro lepší funkčnost by se mělo snímkování pohybovat někde kolem 10 snímků za vteřinu. Další nevýhodou je možnost sledování a vyhodnocování pohybu pouze jedné osoby. Pro využití ve statistice nebo bezpečnostních aplikacích, by bylo potřeba sledovat a vyhodnocovat pohyb většího množství osob. Úpravám by se v tomto případě určitě nevyhnul ani vyhodnocovací algoritmus. Ten by měl být mnohem komplexnější a musel by se umět vypořádat s větším množstvím pohybujících se osob. Po provedení výše zmíněných úprav by mohla aplikace sloužit ke statistickým potřebám. Další možností je využít aplikaci pouze k detekci změn ve sledované scéně a použít ji jako standardní alarm.

Metody detekce pohybujících se objektů za pomoci kamerových systémů jsou neustále vylepšovány a stále více využívány v praxi. Některé pokročilé algoritmy již dnes dokáží sledovat pohybující se objekty, zaznamenávat jejich trasu nebo na základě rozměrů sledovaného objektu určit, zda se jedná o zvíře, člověka nebo automobil. Do budoucna bude určitě zajímavé sledovat vývoj těchto technologií a jejich uplatňování v praxi.

ZÁVĚR V ANGLIČTINĚ

The aim of this graduation thesis was to make the readers acquainted with various types of access systems, computer processing of image and methods for detection of moving objects in a video sequence. The educational application, which enables detecting, monitoring and evaluation of people's movement on the watched scene, was created on the basis of knowledge of the given problems. This application was successfully tested in a laboratory and its functionality was confirmed by performed measurement.

Suggested application is fully functional but we have to count with certain restriction of its usage. It is fully sufficient for laboratory use, to verify the principle of detecting moving objects, observe their movements and evaluate the throughputs. For laboratory purposes would be required to generalize the work with arrays, so that it wouldn't be necessary to intervene in the source code. An option should be added to the main control panel, thanks to which it would be possible to change the evaluating array's size. There are more possibilities how to improve the whole application. I definitely recommend carrying on with the work on improving the application.

The algorithm would have to come through larger changes for the practical use. First of all it would have to be more optimised because the frame rate about 4 frames in a second does not have to be fully sufficient for some purposes. This fact has been proved also by laboratory measurements and for better functionality should the frame rate move somewhere about 10 frames in a second. Another disadvantage is the possibility of detecting and evaluating the motion of only one person. For use in statistics or security applications it would be necessary to detect and evaluate the movement of larger amount of people. Some modifications would surely be needed also in the evaluating algorithm. This algorithm should be more complex and it would have to deal with larger number of moving people. After performing the above mentioned modifications, the application should serve for statistics purposes. Another possibility is to make use of this application just for detecting the changes in the observed scene and so to use it as a standard alarm.

The methods of detecting the moving objects with the help of camera systems are still being improved and more frequently practically used. Nowadays there are some advanced algorithms that are able to observe the moving objects, record their track or to specify, on the basis of the object's size, whether it is an animal, a man or a car. It will

surely be interesting to follow the development of these technologies and their practical use in the future.

SEZNAM POUŽITÉ LITERATURY

- [1] KŘEČEK, Stanislav. *Příručka zabezpečovací techniky*. Cricetus, 2002. 350 s. ISBN 80-902938-2-4
- [2] ČANDÍK, Marek. *Objektová bezpečnost II*. Zlín : UTB ve Zlíně, 2004. 100 s. ISBN 80-7318-217-3
- [3] ZAPLATÍLEK, Karel; DOŇAR, Bohuslav. *MATLAB – tvorba uživatelských aplikací*. Praha : BEN, 2004. 215 s. ISBN 80-7300-133-0
- [4] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. Praha : BEN, 2008. 143 s. ISBN 978-80-7300-233-6
- [5] JAVIDI, Bahram. *Image recognition and classification, algorithms, systems and application*. Marcel Dekker, Inc., 2002. 520 s. ISBN 978-0-8247-0783-5
- [6] CAPPELLINI, V. *Time-varying image processing and moving object recognition*. Elsevier Science, 1997. 346 s. ISBN 0-444-82307-7
- [7] GONZALES, R., C.; WOODS, R., E.; EDDINS, S., L. *Digital image processing using Matlab*. Prentice Hall, 2004. 827 s. ISBN 0-13-008519-7
- [8] JAIN, A., K. *Fundamentals of digital image processing*. Prentice Hall, 1988. 569 s. ISBN 0-13-336165-9
- [9] VOŘÍŠEK, Josef. *Význam kontroly pohybu osob v chráněné zóně z hlediska průmyslu komerční bezpečnosti*. Zlín, 2007. Bakalářská práce. UTB ve Zlíně
- [10] AXIS. *Axis 221 camera* [online]. 2009 [cit. 2010-01-21]. Dostupný z WWW: <http://www.axis.com/files/manuals/um_221_32817_20080901.pdf>
- [11] ŠMIRAUS, Michal. *Vývoj a nové trendy počítačově podporovaných technologií identifikace*. Zlín, 2008. 2 s. Bakalářská práce. UTB ve Zlíně
- [12] Elvi [online]. 2009 [cit. 2010-03-14]. Turnikety. Dostupné z WWW: <<http://www.elvi.cz/trn.html>>
- [13] HÁJEK, Jan. *Automatizace* [online]. 2005 [cit. 2010-03-20]. Optické kódy a RFID. Dostupné z WWW: <<http://www.automatizace.cz/article.php?a=777>>
- [14] Soom [online]. 2007 [cit. 2010-03-28]. Bezpečnost magnetických karet. Dostupné z WWW: <<http://www.soom.cz/index.php?name=articles/show&aid=427>>

- [15] *CardHouse* [online]. [cit. 2010-03-28]. Vše o kartách - karta jako datový nosič. Dostupné z WWW: <<http://cardhouse.cz/produkty/plastove-karty/vse-o-kartach/karta-jako-datovy-nosic/>>
- [16] ROSOL, Ivo. *Systemonline* [online]. 2007 [cit. 2010-03-30]. Čipové karty. Dostupné z WWW: <<http://www.systemonline.cz/it-security/cipove-karty.htm>>
- [17] *International Biometric Group* [online]. 2008 [cit. 2010-04-07]. Biometrics Market and Industry Report 2009-2014. Dostupné z WWW: <http://www.biometricgroup.com/reports/public/market_report.php>
- [18] *Krimi-spik.sweb* [online]. [cit. 2010-04-12]. Daktyloskopie. Dostupné z WWW: <http://krimi-spik.sweb.cz/02_exper/expertiz/02a_dakt/02a_hlav.htm>
- [19] *Tretipol* [online]. [cit. 2010-04-14]. Biometrické metody v bezpečnostní praxi. Dostupné z WWW: <<http://www.tretipol.cz/480/print>>
- [20] *Krimi-spik.sweb* [online]. [cit. 2010-06-14]. Obrazce a znaky kůže. Dostupné z WWW: <http://krimi-spik.sweb.cz/02_exper/expertiz/02a_dakt/02a_kuze.htm>
- [21] DRAHANSKÝ, Martin. *Přehled biometrických systémů a testování jejich spolehlivosti* [online]. VUT v Brně, 2007 [cit. 2010-04-14]. Dostupné z WWW: <http://data.security-portal.cz/clanky/113/odborne_prednasky/Prezentace.pdf>
- [22] *Cleverandsmart* [online]. [cit. 2010-04-20]. Autentizace: přilož prst. Dostupné z WWW: <<http://www.cleverandsmart.cz/autentizace-priloz-prst>>
- [23] KOLÁČEK, Michal. *Svethardware* [online]. 2009 [cit. 2010-04-22]. Šifrování a biometrie pod drobnohledem. Dostupné z WWW: <http://www.svethardware.cz/art_doc-D460E7813803821AC125755C00404E69.html>
- [24] *Strade.fit.vutbr* [online]. 2009 [cit. 2010-04-28]. Snímání žil. Dostupné z WWW: <<http://strade.fit.vutbr.cz/biometrie/snimanizil.html>>
- [25] PIHAL, Roman. *Fotografování* [online]. 2009 [cit. 2010-04-30]. Obrazové problémy digitální fotografie IV. - Záznam barev. Dostupné z WWW: <http://www.fotografovani.cz/art/fotech_df/rom_trouble4.html>
- [26] *Humusoft* [online]. [cit. 2010-05-20]. Matlab. Dostupné z WWW: <<http://www.humusoft.cz/produkty/matlab/matlab/>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|-------|--|
| IZS | Integrovaný záchranný systém |
| ACS | Access system |
| OCR | Optical Character Recognition |
| IATA | International Air Transportation Association |
| ABA | American Bankers Association |
| ATM | Automated Teller Machine |
| SIM | Subscriber Identity Module |
| ISO | International Organization for Standardization |
| IEC | International Electrotechnical Commission |
| USB | Universal Serial Bus |
| RFID | Radio Frequency Identification |
| PIN | Personal identification number |
| AFIS | Automatic Fingerprint Information System |
| RGB | Red-Green-Blue |
| LBP | Local Binary Patterns |
| VGA | Video Graphics Array |
| CCD | charge-coupled device |
| JPEG | Joint Photographic Experts Group |
| MPEG | Moving Picture Experts Group |
| MJPEG | Motion JPEG |
| PC | Personal Computer |
| IP | Internet Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obr. 1 Tripodový turniket [12] | 10 |
| Obr. 2 Plnoprůchodový turniket [12] | 10 |
| Obr. 3 Schéma principu činnosti systému OCR [11] | 12 |
| Obr. 4 Jednorozměrný čárový kód [13] | 13 |
| Obr. 5 Dvourozměrný čárový kód [13] | 13 |
| Obr. 6 Rozdělení stop na magnetickém proužku ID karty [11] | 15 |
| Obr. 7 Schéma kontaktní plošky čipové karty dle ISO/IEC 7816-12 | 17 |
| Obr. 8 Schéma principu činnosti systému RFID | 18 |
| Obr. 9 Anténa a čip na fólii transpodéru RFID [13] | 19 |
| Obr. 10 Princip verifikace [11] | 22 |
| Obr. 11 Princip identifikace [11] | 22 |
| Obr. 12 Graf výnosů jednotlivých biometrických systémů za rok 2009 [17] | 22 |
| Obr. 13 Hlavní identifikační vzory – smyčka, přeslen, oblouk [11] | 23 |
| Obr. 14 Individuální znaky papírárních linií [20] | 24 |
| Obr. 15 Osy měření geometrie ruky [21] | 24 |
| Obr. 16 Měření vzdáleností mezi specifickými body obličeje [23] | 25 |
| Obr. 17 Rozpoznávání termosnímků obličeje [21] | 26 |
| Obr. 18 Duhovka lidského oka [21] | 27 |
| Obr. 19 Sítnice lidského oka [23] | 28 |
| Obr. 20 Čtečka žilního řečiště prstu [24] | 29 |
| Obr. 21 Porovnání dvou zvukových signálů [23] | 29 |
| Obr. 22 Speciální podložka s perem [23] | 30 |
| Obr. 23 Řetězec DNA [21] | 31 |
| Obr. 24 Rozsah vnímání barev lidským okem proti rozsahu RGB [25] | 32 |
| Obr. 25 Obrázek obsahující šum | 34 |
| Obr. 26 Obrázek s odfiltrovaným šumem | 34 |
| Obr. 27 Aktuální snímek | 34 |
| Obr. 28 Použití prahovacího filtru | 34 |
| Obr. 29 Použití erosion filtru | 35 |
| Obr. 30 Histogram | 36 |
| Obr. 31 Referenční snímek | 39 |
| Obr. 32 Aktuální snímek | 39 |

| | |
|---|----|
| Obr. 33 Rozdílový snímek..... | 39 |
| Obr. 34 Prahování rozdílového snímku..... | 39 |
| Obr. 35 Použití erosion filtru na upravený rozdílový snímek | 40 |
| Obr. 36 Použití filtru zvýraznění hran..... | 41 |
| Obr. 37 Schéma zapojení měřicího pracoviště | 44 |
| Obr. 38 Axis 221 Network Camera [10]..... | 45 |
| Obr. 39 Axis IP Utility | 46 |
| Obr. 40 Windows Media Encoder | 47 |
| Obr. 41 Axis MJPEG Capture Driver | 48 |
| Obr. 42 Pracovní prostředí Matlabu..... | 51 |
| Obr. 43 Editor m-souborů..... | 52 |
| Obr. 44 Princip fungování navrženého algoritmu | 53 |
| Obr. 45 Ovládací panel detektoru pohybu..... | 54 |
| Obr. 46 Zobrazení osového kříže..... | 55 |
| Obr. 47 Vyznačení mezí..... | 55 |
| Obr. 48 Snímaná scéna v laboratoři C304..... | 57 |
| Obr. 49 Šum (vypnutá Eroze/dilatace)..... | 58 |
| Obr. 50 Průchod dvou osob současně | 58 |
| Obr. 51 Měřicí pracoviště..... | 60 |

SEZNAM TABULEK

| | |
|--|----|
| Tab. 1 Porovnání vlastností optických systémů [13] | 13 |
| Tab. 2 Porovnání vlastností čipových systémů [13] | 20 |
| Tab. 3 Naměřené hodnoty (ref. snímek – malé pole)..... | 58 |
| Tab. 4 Naměřené hodnoty (ref. snímek – středně velké pole)..... | 58 |
| Tab. 5 Naměřené hodnoty (ref. snímek – velké pole)..... | 59 |
| Tab. 6 Naměřené hodnoty (bez ref. snímku – malé pole) | 59 |
| Tab. 7 Naměřené hodnoty (bez ref. snímku – středně velké pole) | 59 |
| Tab. 8 Naměřené hodnoty (bez ref. snímku – velké pole) | 59 |

SEZNAM PŘÍLOH

P I: Zdrojový kód aplikace pro monitorování počtu osob v budovách (Matlab)

PŘÍLOHA PI: ZDROJOVÝ KÓD APLIKACE PRO MONITOROVÁNÍ POČTU OSOB V BUDOVÁCH (MATLAB)

```
% Program pro detekci pohybujících se objektů ve snímané scéně
% Vytvořeno 10.5.2010
% Autor: Bc. Martin Bobek
% -----

function detekce_pohybu(vstpar)

if nargin==0 % jestliže je funkce volána bez vstupního parametru
    % ----- definice grafických objektů Figure a Uicontrol -----
    % vykreslení objektu Figure1 - Ovládací panel detektoru pohybu
    f1=figure('units','pixels','name','Ovládací panel detektoru pohybu',...
        'position',[40 100 300 600],'tag','Figure1',...
        'numbertitle','off','resize','on','menubar','none');
    % tlačítko Start - spuštění algoritmu pro detekci pohybu
    h1=uicontrol('units','normalized','style','Push',...
        'position',[0.1 0.91 0.3 0.07],'tag','start',...
        'string','Start','foregroundcolor','black',...
        'callback','detekce_pohybu start','fontweight','bold',...
        'fontsize',16);
    % tlačítko Stop - zastavení algoritmu pro detekci pohybu
    h2=uicontrol('units','normalized','style','Push',...
        'position',[0.65 0.91 0.3 0.07],'tag','stop',...
        'string','Stop','foregroundcolor','black',...
        'callback','detekce_pohybu stop','fontweight','bold',...
        'fontsize',16);
    % tlačítko Konec - ukončení programu
    h3=uicontrol('units','normalized','style','Push',...
        'position',[0.65 0.02 0.3 0.07],'tag','konec',...
        'string','Konec','foregroundcolor','black',...
        'callback','detekce_pohybu konec','fontweight','bold',...
        'fontsize',16);
    % zapne metodu porovnávání aktuálního snímku se snímkem referenčním
    h4=uicontrol('units','normalized','style','Check',...
        'position',[0.1 0.85 0.6 0.05],'tag','referencni',...
        'string','Referenční snímek','foregroundcolor','black',...
        'backgroundcolor',get(gcf,'color'),...
        'fontweight','bold','fontsize',12);
    % počet snímků, po kterých se automaticky obnoví referenční snímek
    h5=uicontrol('units','normalized','style','Edit',...
        'position',[0.2 0.74 0.3 0.05],'tag','snimky',...
        'string','0','fontsize',12,'fontweight','bold');
    % zapnutí filtru eroze a dilatace
    h6=uicontrol('units','normalized','style','Check',...
        'position',[0.1 0.68 0.6 0.05],'tag','eroze',...
        'string','Eroze/dilatace','foregroundcolor','black',...
        'backgroundcolor',get(gcf,'color'),...
        'fontweight','bold','fontsize',12);
    % rozbalovací menu pro výběr směru příchoďů a odchodů
    h7=uicontrol('units','normalized','style','Popup',...
        'position',[0.55 0.57 0.4 0.1],'tag','smer',...
        'string','z prava do leva|z leva do prava',...
        'callback','detekce_pohybu smer');
    % tlačítko Reset - ukončení programu
    h8=uicontrol('units','normalized','style','Push',...
        'position',[0.1 0.02 0.3 0.07],'tag','reset',...
        'string','Reset','foregroundcolor','black',...
```



```

        'callback','detekce_pohybu reset','fontweight','bold',...
        'fontsize',16);
% kolonka pro počet příchodů
h9=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.24 0.3 0.05],'tag','prichody',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
% kolonka pro počet odchodů
h10=uicontrol('units','normalized','style','Text',...
    'position',[0.6 0.24 0.3 0.05],'tag','odchody',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
% kolonka pro celkový počet lidí v místnosti
h11=uicontrol('units','normalized','style','Text',...
    'position',[0.35 0.12 0.3 0.05],'tag','celkem',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
% ----- kolonky pro výpis z pole -----
h12=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.50 0.25 0.05],'tag','stred1',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
h13=uicontrol('units','normalized','style','Text',...
    'position',[0.4 0.50 0.25 0.05],'tag','stred2',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
h14=uicontrol('units','normalized','style','Text',...
    'position',[0.7 0.50 0.25 0.05],'tag','stred3',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
% ----- kolonky pro výpis z pole2 -----
h15=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.38 0.25 0.05],'tag','rozdil1',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
h16=uicontrol('units','normalized','style','Text',...
    'position',[0.4 0.38 0.25 0.05],'tag','rozdil2',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
h17=uicontrol('units','normalized','style','Text',...
    'position',[0.7 0.38 0.25 0.05],'tag','rozdil3',...
    'foregroundcolor','white','backgroundcolor','black',...
    'string',0,'fontweight','bold','fontsize',12);
% text - Příchody
h18=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.29 0.3 0.05],'tag','text1',...
    'string','Příchody','foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',14);
% text - Odchody
h19=uicontrol('units','normalized','style','Text',...
    'position',[0.6 0.29 0.3 0.05],'tag','text2',...
    'string','Odchody','foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',14);
% text - Celkový počet lidí v místnosti
h20=uicontrol('units','normalized','style','Text',...
    'position',[0.05 0.17 0.9 0.05],'tag','text3',...
    'string','Celkový počet lidí v místnosti',...
    'foregroundcolor','black','fontweight','bold',...
    'fontsize',14,'backgroundcolor',get(gcf,'color'));

```

```

% text - Obnovení referenčního snímku po
h21=uicontrol('units','normalized','style','Text',...
    'position',[0.05 0.79 0.9 0.05],'tag','text4',...
    'string','Obnovení referenčního snímku po',...
    'foregroundcolor','black','fontweight','bold',...
    'fontsize',12,'backgroundcolor',get(gcf,'color'));

% text - snímcích
h22=uicontrol('units','normalized','style','Text',...
    'position',[0.55 0.73 0.3 0.05],'tag','text5',...
    'string','snímcích','foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',12);

% text - Směr příchodů
h23=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.62 0.4 0.05],'tag','text6',...
    'string','Směr příchodů','foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',12);

% text - Souřadnice bodů pohybu
h24=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.55 0.8 0.05],'tag','text6',...
    'string','Souřadnice bodů pohybu',...
    'foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',12);

% text - Rozdíly mezi body pohybu
h25=uicontrol('units','normalized','style','Text',...
    'position',[0.1 0.43 0.8 0.05],'tag','text6',...
    'string','Rozdíly mezi body pohybu',...
    'foregroundcolor','black',...
    'backgroundcolor',get(gcf,'color'),'fontweight','bold',...
    'fontsize',12);

% ----- reakce na stisk tlačítka -----
else % jestliže je funkce volána se vstupním parametrem (volá sama sebe)
    switch(vstpar) % přepínač odezev objektu Uicontrol
        case('start') % reakce na stisk tlačítka Start
            % hlavní část programu vyhodnocující pohyb ve sledované scéně
            clear all;
            global active;
            active=1;
            clc;
            % vykreslení objektu Figure2 - Detektor pohybu
            f2=figure('units','pixels','name','Detektor pohybu',...
                'position',[400 100 640 480],'tag','Figure2',...
                'visible','off','numbertitle','off','resize','on',...
                'menubar','none');
            % zakázání interakce některým objektům ve Figure1
            set(findobj('tag','referencni'),'enable','off');
            set(findobj('tag','snimky'),'enable','off');
            set(findobj('tag','smer'),'enable','off');
            set(findobj('tag','reset'),'enable','off');
            % nastavení počátečních hodnot
            pom=0;
            s=0;
            pole1=zeros(1,7);
            pole2=zeros(1,6);
            snim=str2num(get(findobj('tag','snimky'),'string'));
            prich=str2num(get(findobj('tag','prichody'),'string'));
            odch=str2num(get(findobj('tag','odchody'),'string'));
            celk=str2num(get(findobj('tag','celkem'),'string'));

```

```

% zobrazení prvního snímku a nastavení mezí ve sledované scéně
vid=videoinput('winvideo',2);
obr1=getsnapshot(vid);
f2=findobj('tag','Figure2'); % zjištění HANDLE Figure2
set(f2,'visible','on'); % zviditelnění Figure2
figure(f2); % Figure2 bude aktivní
imshow(obr1);
g=ginput(2);
bound_left=g(1,1);
bound_right=g(2,1);
if bound_left>bound_right
    % ošetření správného umístění mezí
    pom=bound_left;
    bound_left=bound_right;
    bound_right=pom;
end
obr1=rgb2gray(obr1);
obr2=rgb2gray(getsnapshot(vid));
while (active==1)
    ref=get(findobj('tag','referencni'),'value');
    if ref==1
        % porovnávání aktuálního snímku s referenční smímkem
        if snim==0
            orig=getsnapshot(vid);
            obr2=rgb2gray(orig);
            obr=(abs(obr1-obr2));
        else
            s=s+1;
            if ((s>=snim) && (pole2(1,4)==0) && (pole2(1,5)==0)...
                && (pole2(1,6)==0))
                obr1=rgb2gray(getsnapshot(vid));
                s=0;
            end
            orig=getsnapshot(vid);
            obr2=rgb2gray(orig);
            obr=(abs(obr1-obr2));
        end
    else
        % porovnávání aktuálního snímku s předchozím snímkem
        orig=getsnapshot(vid);
        obr3=rgb2gray(orig);
        obr1=obr2;
        obr2=obr3;
        obr=(abs(obr1-obr2));
    end
    % převod rozdílového snímku na monochromatický
    bw=im2bw(obr,0.1);
    erode=get(findobj('tag','eroze'),'value');
    if erode==1
        % filtry eroze a dilatace
        bw=bwmorph(bw,'erode');
        bw=bwmorph(bw,'dilate');
    end
    f2=findobj('tag','Figure2'); % zjištění HANDLE Figure2
    set(f2,'visible','on'); % zviditelnění Figure2
    figure(f2); % Figure2 bude aktivní
    imshow(orig);
    % nalezení krajních bodů pohybujícího se objektu
    [x1,y1,x2,y2]=zjisti_okraje(bw);
    hold on
    % vykreslení mezních bodů pro detekci pohybu

```

```

vert=[0 240];
line([bound_left bound_left],vert,'color','yellow')
line([bound_right bound_right],vert,'color','yellow')
% pomocné výpočty pro určení rozměrů pohybujícího se objektu
w=x2-x1;
h=y2-y1;
% určení pozice středu
width=w/2;
heig=h/2;
tezw=x1+width;
tezh=y1+heig;
% vykreslení obdélníku kolem pohybujícího se objektu
if((w>0) && (h>0))
    rectangle('Position',[x1 y1 w h],'EdgeColor','r');
    % vykreslení středu obdélníku
    plot(tezw,tezh,'r*');
    drawnow;
    hold off
end
% posunutí prvků v poli1
for i=1:1:(6)
    pole1(1,i)=pole1(1,i+1);
end
% dosazení hodnoty do posledního místa v poli1
pole1(1,7)=tezw;
% rozdělí jednotlivých prvků v poli1 uložené do pole2
for j=1:1:(6)
    pole2(1,j)=pole1(1,j)-pole1(1,j+1);
end
% pohyb=get(findobj('tag','smer'),'value');
if pohyb==1
    % vyhodnocení pohybu (příchod je z prava)
    if ((pole2(1,1)>0) && (pole2(1,2)>0) && (pole2(1,3)>0)...
        && (pole2(1,4)>0) && (pole2(1,5)>0)...
        && (pole2(1,6)>0) && (pole1(1,7)<(bound_left+20))...
        && (pole1(1,7)>(bound_left-20)))
        prich=prich+1;
        pole1=zeros(1,7);
        pole2=zeros(1,6);
    end
    if ((pole2(1,1)<0) && (pole2(1,2)<0) && (pole2(1,3)<0)...
        && (pole2(1,4)<0) && (pole2(1,5)<0)...
        && (pole2(1,6)<0) && (pole1(1,7)>(bound_right-20))...
        && (pole1(1,7)<(bound_right+20)))
        odch=odch+1;
        pole1=zeros(1,7);
        pole2=zeros(1,6);
    end
    celk=prich-odch;
else
    % vyhodnocení pohybu (příchod je z leva)
    if ((pole2(1,1)<0) && (pole2(1,2)<0) && (pole2(1,3)<0)...
        && (pole2(1,4)<0) && (pole2(1,5)<0)...
        && (pole2(1,6)<0) && (pole1(1,7)>(bound_right-20))...
        && (pole1(1,7)<(bound_right+20)))
        prich=prich+1;
        pole1=zeros(1,7);
        pole2=zeros(1,6);
    end
    if ((pole2(1,1)>0) && (pole2(1,2)>0) && (pole2(1,3)>0)...
        && (pole2(1,4)>0) && (pole2(1,5)>0)...

```

```

        && (pole2(1,6)>0) && (pole1(1,7)<(bound_left+20))...
        && (pole1(1,7)>(bound_left-20)))
    odch=odch+1;
    pole1=zeros(1,7);
    pole2=zeros(1,6);
end
celk=prich-odch;
end
% předání výsledných hodnot do objektů ve Figure1
set(findobj('tag','prichody'),'string',prich);
set(findobj('tag','odchody'),'string',odch);
set(findobj('tag','celkem'),'string',celk);
set(findobj('tag','stred1'),'string',pole1(1,5));
set(findobj('tag','stred2'),'string',pole1(1,6));
set(findobj('tag','stred3'),'string',pole1(1,7));
set(findobj('tag','rozdil1'),'string',pole2(1,4));
set(findobj('tag','rozdil2'),'string',pole2(1,5));
set(findobj('tag','rozdil3'),'string',pole2(1,6));
end
close 2
case('stop') % reakce na stisk tlačítka Stop
% zastavení vykreslování snímků ve Figure2
global active;
active=0;
% zrušení zákazu interakce některým objektův ve Figure1
set(findobj('tag','referencni'),'enable','on');
set(findobj('tag','snimky'),'enable','on');
set(findobj('tag','smer'),'enable','on');
set(findobj('tag','reset'),'enable','on');
case('reset') % reakce na stisk tlačítka Reset
% vynulování všech údajů ve Figure1
set(findobj('tag','prichody'),'string',0);
set(findobj('tag','odchody'),'string',0);
set(findobj('tag','celkem'),'string',0);
set(findobj('tag','stred1'),'string',0);
set(findobj('tag','stred2'),'string',0);
set(findobj('tag','stred3'),'string',0);
set(findobj('tag','rozdil1'),'string',0);
set(findobj('tag','rozdil2'),'string',0);
set(findobj('tag','rozdil3'),'string',0);
case('konec') % reakce na stisk tlačítka Konec
% ukončení práce celého programu
close all
end
end
end

```

```

% Funkce pro nalezení krajních bodů pohybujícího se objektu
% Vytvořeno 17.5.2010
% Autor: Bc. Martin Bobek
% -----

```

```
function [x1,y1,x2,y2] = zjisti_okraje(img)
```

```

[h w]=size(img);
x1=0;
y1=0;
x2=0;
y2=0;

```

```
% nalezení levého krajního bodu pohybujícího se objektu
for x=1:1:w
    s=sum(img(:,x));
    if (s~=0)
        x1=x;
        break
    end
end

% nalezení pravého krajního bodu pohybujícího se objektu
for x=w:-1:1
    s=sum(img(:,x));
    if (s~=0)
        x2=x;
        break
    end
end

% nalezení horního krajního bodu pohybujícího se objektu
for y=1:1:h
    s=sum(img(y,:));
    if (s~=0)
        y1=y;
        break
    end
end

% nalezení dolního krajního bodu pohybujícího se objektu
for y=h:-1:1
    s=sum(img(y,:));
    if (s~=0)
        y2=y;
        break
    end
end
```