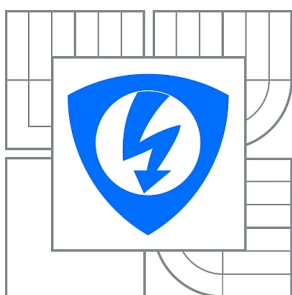




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MOŽNOSTI PŘIPOJENÍ KAMER K POČÍTAČI PC

POTENTIALS OF CAMERA CONNECTION TO PC COMPUTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ PELC

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IVO HERMAN, CSc.

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Lukáš Pelc

ID: 109708

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Možnosti připojení kamer k počítači PC

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je připojit k jednodeskovému počítači PC obsahujícím LCD displej o rozlišení 800x480 dvě a více kamer. Seznamte se s možnostmi připojení kamer k počítači PC. Navrhněte způsob, jakým na LCD displeji počítače bude možno zobrazit jeden obraz o velikosti 640x480 pixelů nebo dva až čtyři obrázky 320x240 pixelů. Určete hloubku barev a velikost datových toků obrázků nebo videa. Kamery budou k počítači připojeny pomocí ethernetu.

DOPORUČENÁ LITERATURA:

- [1] Babíček,R.: Jak na počítač - Digitální video. Computer Press, 2009.
- [2] Darwin,.F: Java - kuchařka programátora. Computer press. 2006.
- [3] Matthew,M., Stones,R.: Linux. Začínáme programovat. Computer pres. 2008

Termín zadání: 7.2.2011

Termín odevzdání: 2.6.2011

Vedoucí práce: Ing. Ivo Herman, CSc.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce se zabývá síťovými kamerami a možnostmi jejich připojení k počítači PC. Síťové kamery jsou rozebrány z pohledu vnitřní stavby, vnější konstrukce a z těchto hledisek odvíjeného využití. Komunikace s PC a uživatelem je zajištěna komunikačním rozhraním a ovládacím softwarem. Nejdůležitější funkcí při používání síťových kamer je jejich výstup – tedy obraz. Ten je složen z několika dílčích faktorů, jako je například způsob kódování, komprese a snímková frekvence. Teorie je aplikována do praxe v podobě výběru nejvhodnější kamery, způsobu jejího zapojení a naprogramování vlastní ovládací aplikace v programovacím jazyce Java.

KLÍČOVÁ SLOVA

IP kamera, Ethernet, CGI příkaz, Java, ovládací software, síťové video, komunikace

ABSTRACT

Bachelor thesis deals with the possibilities of network cameras and their connection to a PC. Network cameras are analyzed in terms of internal structure, external structure and aspects of their use. Communication with the PC and the user is provided with a communication interface and control software. The most important function in the use of network cameras is their output – a video. It is composed of several individual factors, such as the encoding method, compression and frame rate. The theory is applied in practice as proposed by the best camera, the method of connection and programming of custom control applications in the Java programming language.

KEYWORDS

IP camera, Ethernet, CGI command, Java, control software, network video, communication

PELC, L. *Možnosti připojení kamer k počítači PC*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 69 s. Vedoucí bakalářské práce Ing. Ivo Herman, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Možnosti připojení kamer k počítači PC jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Ivu Hermanovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	ix
Úvod	10
1 Cíl práce	11
2 Přehled Současného Stavu Problematiky	12
2.1 Kamera.....	12
2.1.1 Princip zachycení obrazu	12
2.2 Vnitřní stavba síťové kamery.....	12
2.2.1 Snímací čip	13
2.2.2 Objektiv	17
2.2.3 Další komponenty síťové kamery	18
2.3 Možnosti připojení k PC	19
2.3.1 Drátová komunikační rozhraní	19
2.3.2 Bezdrátová komunikační rozhraní	20
2.4 Možnosti snímání obrazu.....	20
2.4.1 Kvalita obrazu.....	20
2.4.2 Progresivní skenování.....	21
2.4.3 Kompenzace světla na pozadí.....	21
2.4.4 Snímková frekvence	21
2.5 Síťové video.....	22
2.5.1 Komprese obrazu	22
2.5.2 Standardy pro kompresi statických obrázků	23
2.5.3 Standardy pro kompresi videa	23
2.5.4 Rozlišení obrazu	24
2.6 Síťové kamery a jejich využití.....	25
2.6.1 Vnitřní kamery	26
2.6.2 Venkovní kamery.....	26
2.6.3 Ovladatelné kamery	27
2.6.4 Dome kamery.....	28

2.7	Možnosti zapojení více kamer	28
2.8	Komunikace kamery a PC	29
2.8.1	Přenos dat.....	29
2.8.2	Zabezpečení přenosu dat.....	29
2.8.3	Přenosové protokoly	29
3	Známé způsoby řešení	32
3.1	Ovládací programy kamer Edimax	32
3.1.1	IPCam Surveillance Software	32
3.1.2	Ovládání přes internetový prohlížeč	33
4	Návrh vlastního řešení	35
4.1	Výběr nejvhodnější kamery	35
4.2	Výběr nejvhodnějšího komunikačního rozhraní.....	36
4.3	Vlastní schéma zapojení	36
4.4	CGI příkazy.....	37
4.5	Technické vybavení	38
4.6	Programové vybavení	38
5	Vlastní návrh programu	39
5.1	Hlavní třída Main.....	39
5.2	Grafické rozhraní Gui	40
5.3	Zobrazovací grafické prvky ImageViewer	41
5.4	Vývojový diagram	43
6	Výsledky řešení	44
6.1	Popis programu	44
6.2	Poznámky k řešení	49
7	Závěr	50
	Literatura	51
	Seznam symbolů, veličin a zkratek	53
	Seznam příloh	55

SEZNAM OBRÁZKŮ

Obr. 2.1: Vnitřní stavba síťové kamery (zdroj: autor)	13
Obr. 2.2: Rozlišení obrazu (použito z [16])	25
Obr. 2.3: Vnitřní IP kamera AXIS 210A (použito z [13])	26
Obr. 2.4: Venkovní kamera Edimax IC-9000 (použito z [2]).....	27
Obr. 2.5: Ovladatelná kamera Edimax IC-7000PT (použito z [3])	27
Obr. 2.6: Dome kamera AXIS 209FD-R (použito z [14])	28
Obr. 3.1: IPCam Surveillance Software (zdroj: autor)	33
Obr. 3.2: Ovládání přes internetový prohlížeč (zdroj: autor).....	34
Obr. 4.1: IP kamera Edimax IC-3010 (použito z [4])	35
Obr. 4.2: Schéma zapojení čtyř kamer (zdroj: autor)	36
Obr. 5.1: Základní vývojový diagram (zdroj: autor)	43
Obr. 6.1: Výchozí okno programu (zdroj: autor).....	44
Obr. 6.2: Zobrazení obrazu jedné kamery v nižším rozlišení (zdroj: autor).....	45
Obr. 6.3: Nejvyšší komprese JPEG snímku (zdroj: autor)	46
Obr. 6.4: Otočení obrazu o 180° (zdroj: autor).....	47
Obr. 6.5: Zobrazení obrazů ze dvou kamer (zdroj: autor)	47
Obr. 6.6: Zobrazení obrazu ze tří kamer (zdroj: autor).....	48
Obr. 6.7: Zobrazení obrazu ze čtyř kamer (zdroj: autor)	48
Obr. 6.8: Nedostatky snímače kamery (zdroj: autor).....	49

SEZNAM TABULEK

Tab. 2.1: Přenosové protokoly	30
-------------------------------------	----

ÚVOD

Bakalářská práce se zabývá především možnostmi současných digitálních IP kamer. Kamera je složité zařízení, jehož výsledná funkčnost se skládá z několika aspektů. V práci je tedy popsána vnitřní a vnější stavba kamer, způsoby připojení k počítači, možnosti zobrazení obrazu a videa. Ovládací software kamer je rozebrán z pohledu uživatelského ovládání prostřednictvím počítače i schopností pro zobrazení obrazu připojených kamer.

Hlavním cílem práce je z dostupných zařízení a prostředků vytvořit komplexní celek pro obsluhu dvou a více kamer. K dalším cílům patří nalezení způsobu vhodného připojení kamer k obslužnému PC, řešení jejich ovládání a možností zobrazení jednoho či více obrazů použitých kamer. Toto řešení je nastíněno pomocí výrobcem dodaného obslužného softwaru, ale především vlastním ovládacím programem.

Bakalářská práce je uzavřena zhodnocením použitých kamer a komunikačního rozhraní. Dále je ukázkou možností vlastní aplikace pro obsluhu kamer, která odhaluje nejen funkční možnosti daných kamer, ale i jejich nedostatky.

1 CÍL PRÁCE

Cílem práce je připojit k jednodeskovému počítači PC s LCD displejem o rozlišení 800×480 dvě a více kamer a seznámit se s možnostmi připojení kamer k počítači PC. Praktická část řeší navržení způsobu, jakým na LCD displeji počítače bude možno zobrazit jeden obraz o velikosti 640×480 pixelu nebo dva až čtyři obrázky 320×240 pixelu. Navrhuje určení hloubky barev a velikost datových toků obrázku nebo videa, u toho navíc počet snímků za sekundu.

V práci bude zkoumána vnitřní stavba kamer i možnosti jejich použití v závislosti na jejich vnější stavbě. Je potřeba rozebrat také ovládací aplikace pro komunikaci s uživatelem, funkčnost a možnosti těchto aplikací.

K dispozici bude vybrána dvojice kamer splňující podmínky zadání. Jelikož budou k dispozici kamery pouze dvě, bude potřeba navrhnout nějaké řešení zobrazení obrazu z více než dvou kamer. Kamery budou k počítači připojeny pomocí ethernetu, což znamená vybrat vhodnou síťovou komponentu pro jejich připojení.

Výsledkem práce bude návrh vlastní ovládací aplikace. Tento návrh bude realizován v programovacím jazyce Java za pomoci vývojového prostředí NetBeans. Aplikace prověří možnosti použitých kamer, jejich funkce a případné nedostatky.

2 PŘEHLED SOUČASNÉHO STAVU PROBLEMATIKY

2.1 Kamera

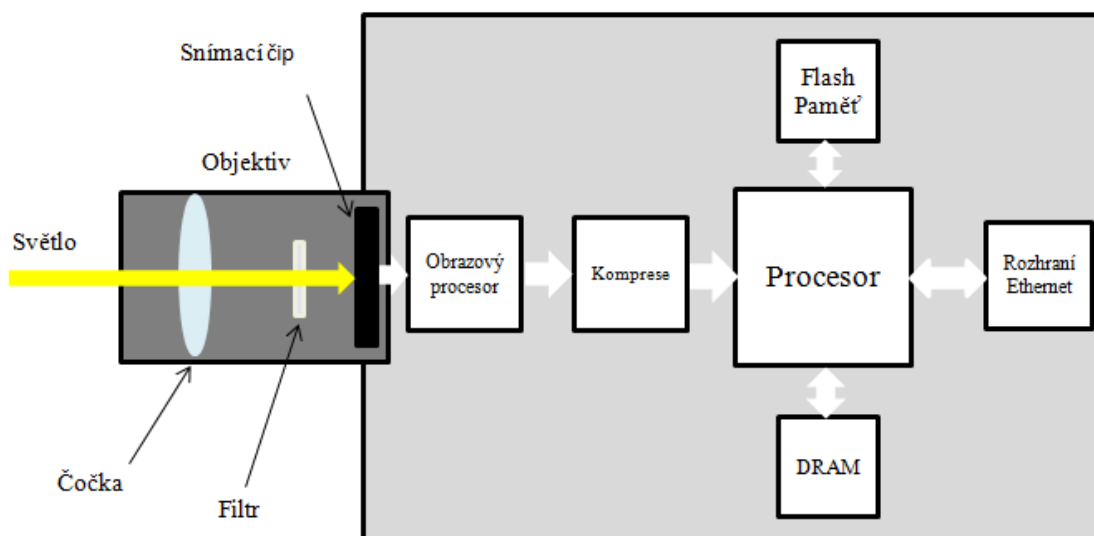
Kamera je definována jako zařízení umožňující zachycení obrazu a zvuku pro pozdější interpretaci. Základní rozdělení kamer podle principu zpracování obrazu a zvuku je na analogové a digitální. Kamery mají různé účely použití. Zmínku zaslouží profesionální studiové kamery, běžné uživatelské kamery s různými druhy záznamových médií, webkamery a v současné době i kamery v mobilních telefonech. Všechny se liší svým použitím, kvalitou obrazu, kvalitou zpracování a hlavně cenou. Tato práce je zaměřena pouze na digitální síťové kamery. Další pojednání o těchto kamerách je mimo její rozsah.

2.1.1 Princip zachycení obrazu

Základní funkcí kamery je zachycení obrazu, respektive světla o různých vlnových délkách. Toto světlo procházející objektivem kamery je zachyceno na světločivný snímací čip. Tam je přeměněno na elektrický signál a dále zpracováváno. Zpracován je signál z analogového formátu na digitální a předán výpočetní jednotce, která provede kompresi. Odtud se přes výstupní rozhraní kamery dostane signál na vedení, na jehož konci je obslužné zařízení, či software umožňující pozdější zpracování obrazu a ovládání kamery.

2.2 Vnitřní stavba síťové kamery

Síťová kamera je složena z objektivu, snímacího čipu, obrazového procesoru a hlavního procesoru, který je napojený na komunikační rozhraní a paměť (viz Obr. 2.1).



Obr. 2.1: Vnitřní stavba síťové kamery (zdroj: autor)

2.2.1 Snímací čip

Snímací čip představuje klíčovou součást digitální kamery. Je složen z několika tisíc až milionů buněk. Jedná se o analogovou součástku, jejímž výstupem je analogové napětí. Za senzorem je analogově digitální převodník, který se stará o převod na digitální signál. Problémem je, že snímací senzor je vždy barvoslepý. Barevného záznamu se docílí použitím Bayerovy masky, což je soustava barevných RGB skel uspořádaných do pravidelné struktury. Jednotlivé buňky senzoru mají nad sebou různé barevné filtry, čtvrtinu modrých, čtvrtinu červených a polovinu zelených. Každá buňka předává elektronice úroveň jasu, tedy určitou signálovou úroveň, která na ni dopadne, a to buď červeného, zeleného nebo modrého světla. Tuto signálovou úroveň reprezentuje digitální číslo z rozsahu 0 – neprochází žádné světlo, až 255 – maximální úroveň světla (úroveň oslepení). Tento rozsah má 256 úrovní, což představuje 8 bitové snímání typické pro formát JPEG. V digitálních zrcadlovkách nebo HD kamerách se můžeme setkat i s 12 bitovým zpracováním umožňující 4096 úrovní, či 14 bitovým zpracováním obrazu s 16 384 úrovněmi. Vícebitové zpracování představuje také použití odpovídajícího analogově digitálního převodníku. Vždy platí, že čím vícebitové zpracování je použito, tím je výsledný snímek jemnější. Po předání jednotlivých úrovní signálu buňkami se celkový signál z obrazového senzoru předá obrazovému procesoru, který ze signálu vypočítá výsledný snímek, se kterým je možno dále pracovat.

Dnešní kamery obsahují dva typy snímačů – CCD a CMOS. CCD snímače jsou vyvinuty speciálně pro kamerový průmysl, zatímco CMOS snímače jsou založeny na standardní technologii využívající se při výrobě paměťových čipů. CCD senzor má při stejné velikosti čipu nižší šum a vyšší kvalitu obrazu. CMOS má zase řadu jiných výhod (např. nižší cena, menší složitost výroby a možnost vysoké integrace). Technologická úroveň obou těchto čipů je téměř stejná, a proto se rozdíly stírají. Největší rozdíly nalezneme ve sběru signálu z jednotlivých buněk a v logice ovládání senzoru.

CMOS (Complementary Metal Oxide Semiconductor) senzory se dříve používaly především díky jejich velikosti a ceně do nejlevnějších síťových a webových kamer a také do mobilních telefonů za účelem co nejnižší ceny přístroje. Postupným zlepšováním se dostaly do nejvyšší sféry digitálních HD kamer či zrcadlovek a do profesionálních zařízení především díky vysokému rozlišení snímačů a jejich snadné výrobě.

CMOS obrazový senzor používá stejnou výrobní technologii jako integrované obvody typu pamětí či procesorů. Výsledkem je vyšší možnost integrace, kdy je možné přímo v senzoru integrovat několik dalších obvodů, ale také především levnější výroba těchto čipů. Každá buňka CMOS senzoru obsahuje svůj vlastní zesilovač signálu a může být tak přímo adresována i čtena pomocí svých souřadnic. Rozdíl mezi CCD a CMOS tedy není až tak v konstrukci citlivé části buňky, tedy ve fotodiodě, ale ve čtení buňky. Každá buňka je adresována pomocí svých souřadnic. Díky tomu se těmto senzorům někdy říká APS (Active Pixel Sensor), tedy senzor s aktivními pixely. Zesilovač u každé buňky má za následek snížení velikosti světločivné fotodiody, tím se snižuje fill factor. Fill factor je poměr plochy citlivé části, tedy fotodiody, k celkové ploše buňky. Malá velikost světločivné buňky vyžaduje větší zesílení mající za následek vyšší šum. Zvýšení šumu se zabraňuje použitím mikroobjektivů, které soustředí světlo přesně na světločivnou část buňky.

Díky aktivním buňkám a jejich přímému adresování je velice rychlé čtení dat ze senzoru CMOS. Lze také číst jen část obrazových dat, což se u kamer využívá k digitální stabilizaci obrazu nebo sledování pohybu objektu.

CCD (Charged Coupled Device) obrazový senzor je historicky mnohem starší než CMOS. CCD snímač je vlastně určitý paměťový registr vázaný na světlo. Má schopnost transponovat signál z jedné buňky do druhé, aniž by docházelo ke ztrátám na kvalitě signálu. Signál se tak posouvá až k okrajové buňce, která je napojena na posuvný registr. Posuvný registr poté jeden signál po druhém předá do zesilovače a odtud do analogově digitálního převodníku. Posun signálu v buňkách probíhá za pomoci nábojové vazby, což dalo těmto senzorům i jejich jméno Charged Coupled, spojeny nábojem. CCD senzory používají jinou speciální technologii výroby než ostatní integrované obvody. Je tedy složité dosáhnout velkého rozlišení snímačů a vměštění ostatní elektroniky přímo do snímače. Řídící obvody, zesilovače a analogově digitální převodníky musí být mimo senzor. Na základě toho vyžaduje CCD senzor větší rozsah různého napájení pro jednotlivé integrované obvody na desce s plošnými spoji. Tolik integrovaných obvodů má za následek mnohem větší spotřebu elektrické energie než technologie CMOS. CCD snímače jsou nasazovány do kompaktních fotoaparátů a kamer střední třídy.

Rozlišení snímačů se udává v milionech obrazových bodů neboli megapixelech. Udává se jako násobek vertikálních a horizontálních světločivných buněk na snímači, neboli kolik obrazových bodů je snímač schopen rozlišit ve vertikálním a horizontálním směru. Dnešní snímače mají u nejlevnějších webových kamer a mobilních telefonů rozlišení od desítek tisíc pixelů, u digitálních zrcadlovek až desítky megapixelů. Pokud má například síťová kamera rozlišení 640×480 bodů, po vynásobení těchto čísel získáme výsledné rozlišení snímače 307 200 pixelů. U digitální zrcadlovky se můžeme při rozlišení snímače 6000×4000 obrazových bodů dopočítat výsledného rozlišení 24 MPx. U digitálních fotoaparátů a kamer je potřeba rozlišovat dva pojmy týkající se obrazového snímače – celkový počet pixelů a efektivní počet pixelů. Celkový počet pixelů je číslo, udávající počet světločivných buněk obrazového snímače. Efektivní počet pixelů je výsledný počet obrazových bodů obrazu v podobě videa nebo fotografie. Zbylé buňky snímače se používají pro výpočet Bayerovy interpolace na okrajích senzoru a jako referenční černé pixely.

Při **tříčipovém uspořádání** prochází obraz od objektivu přes soustavu polopropustných zrcadel potažených barevnými filtry oddělující jednotlivé barvy RGB složky do tří snímačů. Každý snímač zachytává jen červenou, zelenou nebo modrou složku obrazu. Toto uspořádání čipů má za následek barevně kvalitnější obraz a méně šumu než jednočipové řešení. Použití nacházelo především u profesionálních televizních kamer. Dnes se dostalo i do vyšší spotřebitelské sféry. Vzhledem k nutnosti dosti přesného seřízení takové optické soustavy a přítomnosti tří snímačů je toto řešení dražší než jednočipové.

Dynamický rozsah čipu je rozsah odstínů od černé až po bílou barvu, kterou je schopen snímač rozlišit. Je limitován kapacitou jednotlivých buněk, tedy kolik fotonů vzniklých interakcí je schopna pojmout a dále je limitován vlastním šumem buňky. Při malém dynamickém rozsahu se v kontrastní scéně objevují přexponovaná (velmi přesvětlená místa), či podexponovaná místa (velmi tmavá místa bez detailů). S tímto problémem se lze setkat především u levnějších zařízení.

Obrazový šum je jedním z největších problémů digitálních snímačů. Základní příčinou vzniku šumu je tepelný pohyb krystalové mřížky polovodiče. Při tomto jevu se občas uvolní elektron bez přítomnosti fotonu, což má za následek přitažení elektronu k expoziční elektrodě a změnu hodnoty expozice na dané buňce. Z tohoto poznatku lze usoudit, že při vyšší teplotě snímače roste i obrazový šum. Navíc každá buňka má hodnotu šumu jinou. Není tak možné ze snímku šum na 100 % odstranit. Pro dosažení nízkého šumu je nutné mít snímač s co největšími buňkami. Ty tak mají velké kapacity a vyšší odstup signálu od šumu. Platí tedy, že malé snímače, či snímače s velkým rozlišením, produkují vždy více šumu, než snímače velké s menším rozlišením.

2.2.2 Objektiv

Objektivem nazýváme čočku nebo soustavu čoček, přes které proniká obraz na snímací čip. Objektiv tak vytváří opticky změněný obraz, aby soustředil co nejvíce světla přímo na snímač. Objektivy se liší svou konstrukcí. Jako úplně nejzákladnější poslouží spojná čočka. Pro potlačení optických vad objektivů se však používají optické soustavy několika různých druhů čoček. Takováto optická soustava pak dovoluje například měnit svoji ohniskovou vzdálenost, neboli přiblížit či oddálit obraz.

Objektivy se rozdělují podle své ohniskové vzdálenosti, což je šířka záběru, kterou jsou schopny zobrazit na snímač. Je udávána v mm a zpravidla k přepočtu na 35 mm kinofilm. V praxi se ujal především širokoúhlý objektiv (28 mm a méně), které poskytnou velkou šířku záběru a pokryjí tak záběrem třeba celou místnost. Na opačném konci jsou to teleobjektivy (100 mm a výše), jejichž záběr je sice užší, ale předměty zvětšují. Je tedy možné přiblížení předmětu. Lidské oko má ohniskovou vzdálenost přibližně 50 mm, proto objektivy s touto ohniskovou vzdáleností poskytují obraz nejvěrohodnějšímu reálnému vnímání právě lidským okem.

Dalším důležitým faktorem je světelnost objektivu, neboli míra světla, která projde přes objektiv na snímací čip. Je to bezrozměrná hodnota udávaná jako číslo jmenovatele ve zlomku s čitatelem 1. Objektiv se světelností 2,0 propustí polovinu na něj dopadajícího světla.

Možnost výměny objektivu respektive čočky nabídnou především profesionální kamery. Jedná se o C mount nebo CS mount čočky, které přinášejí vysokou kvalitu obrazu. V případě přítomnosti automatické kontroly clony mají i zvýšený dynamický rozsah obrazu.

Vinětace se u kamer neprojevuje tolik jako u fotoaparátů, ale i přesto se s ní můžeme setkat. Světločivné buňky snímače zachytí nejvíce světla v kolmém směru dopadu. Jakékoliv světlo pod jiným úhlem je zaznamenáno s nižší intenzitou. Z běžného objektivu dopadají na snímač kolmé paprsky jen ze středu objektivu. Proto mají kvalitnější kamery i kvalitnější optiku, která splňuje vyšší nároky na kolmý dopad paprsků.

2.2.3 Další komponenty síťové kamery

Obrazový procesor se stará především o zpracování dat ze snímače a jejich interpretaci hlavnímu procesoru. Jeho úkolem je zvolit vhodnou dobu expozice, hodnotu jasu, kontrastu, vyvážení barevných úrovní, ostrosti a jiných parametrů obrazu. Obrazový procesor obsahuje i komponentu starající se o kompresi obrazu.

Hlavní procesor řídí veškeré funkce kamery. Je propojen s obrazovým procesorem, pamětí i komunikačním rozhraním. Je to mozek kamery, který zpracovává data z okolních obvodů a přes komunikační rozhraní i příkazy softwaru. Tyto příkazy poté interpretuje do jednotlivých obvodů a zajišťuje tak danou funkci. Jeho rychlost určuje možnosti a rychlost reakce na příkazy a rychlost snímání obrazu.

Paměť kamery většinou dělí na dva typy. Jednou je operační paměť procesoru, kam si při výpočtech procesor odkládá data. Jedná se o velmi rychlou paměť, která je závislá na napájení, jinak se z ní uložené informace ztratí. Druhým typem paměti je flash paměť, která slouží např. k uložení uživatelského nastavení kamery. Tato paměť je schopna uchovávat informace i při vypnutí kamery.

Komunikační rozhraní se stará o přenos dat na vedení. V případě bezdrátového rozhraní se stará o vyzáření informace do okolního prostředí prostřednictvím antény. Podrobnosti o komunikačním rozhraní jsou uvedeny v kapitole 2.3 Možnosti připojení k PC.

2.3 Možnosti připojení k PC

2.3.1 Drátová komunikační rozhraní

USB (Universal Serial Bus) je univerzální sériová sběrnice s možností připojení periferních zařízení k počítači. Propojovací kabel k zařízením obsahuje 4 vodiče. První pár vodičů je kroucený a slouží k přenosu dat. Druhý pár je nekroucený a slouží k napájení. Délka segmentu od zařízení k zařízení je 5 m. Existuje více verzí USB lišící se především přenosovou rychlostí. Sběrnice USB umožňuje připojení rozbočovačů, k dispozici je tedy více portů. Pokud je rozbočovač napájen a regeneruje signál, tak můžeme připojit zařízení i na větší vzdálenost. Celkem je možno připojit až 127 zařízení. Budoucnost této sběrnice patří technologii verzi USB 3.0 umožňující přenosovou rychlost až 5 Gb/s.

Ethernet je komunikační technologie pro vytváření počítačových sítí. Výhodou je jednoduchost, snadná implementace a instalace. Ethernet je normalizován jako standart IEEE 802.3. Původní ethernet používal sběrniceovou topologii, která je dnes už neefektivní z důvodů, že vysílat mohla jen jedna stanice, čím při vyšším zatížení sítě vznikalo mnoho kolizí. Ethernet je dnes realizován spíše hvězdicovou topologií, kdy jsou jednotlivé stanice připojovány k prepínačům či rozbočovačům. Přenosovým médiem může být koaxiální kabel, kroucená dvojlinka nebo optické vlákno. Výhodná je i délka jednoho segmentu pohybující se u kroucené dvojlinky do 100 m. Existuje několik verzí ethernetu lišících se především v přenosové rychlosti.

PoE (Power over Ethernet) je technologie umožňující napájení zařízení přes síťový datový kabel bez nutnosti připojení samostatného kabelu pro napájení. Výhodou je použití pouze jednoho kabelu pro data i napájení, čímž šetříme kabely při budování sítě. PoE zdroje jsou zálohované, je tedy ošetřena funkčnost zařízení i při výpadku elektrického proudu. Umožněn je i dálkový restart zařízení odpojením a připojením k napájení pomocí příkazu na portu prepínače. Využití najdeme právě pro připojení IP kamer, telefonů, či bezdrátových přístupových bodů. Zjednodušeno z [8].

2.3.2 Bezdrátová komunikační rozhraní

Wi-Fi (Wireless Fidelity) je standard pro bezdrátové lokální sítě založený na specifikaci IEEE 802.11. Z několika standardů Wi-Fi se dnes používají především dva. IEEE 802.11g vysílá ve frekvenčním pásmu 2,4 GHz s maximální přenosovou rychlostí 54 Mb/s. Druhý je IEEE 802.11n s frekvenčním pásmem 2,4 nebo 5 GHz a maximální teoretickou přenosovou rychlostí 300 Mb/s. Z původního záměru zajišťovat vzájemné propojení různých zařízení do lokální sítě se Wi-Fi začalo používat i pro připojení k internetu v rámci větších lokalit pomocí hotspotů. Rozšíření Wi-Fi je dnes obrovské. Nalezneme jej v každém notebooku, v mobilních telefonech a jiných přenosných zařízeních. Může za to především provozování těchto sítí v bezlicenčním pásmu, což má ovšem za následek velké rušení v rámci příslušného frekvenčního spektra. Výhodou je také dosah těchto sítí, který je u základních zařízení přibližně 300 m ve volném prostoru. V případě střešních antén pro připojení k internetu je dosah v řádu kilometrů. Samozřejmě tento dosah snižují určité překážky, jako například zdi budov, či vlivy počasí. Poslední dobou se rozmáhá i technologie MIMO. Pro přenos informací se používá více antén, což má za důsledek hlavně rychlejší komunikaci a menší zahlcení sítě.

2.4 Možnosti snímání obrazu

2.4.1 Kvalita obrazu

Kvalita obrazu je nejdůležitější vlastností kamery, která nás při výběru kamery zajímá nejvíce. Je závislá především na těchto parametrech a funkcích:

- typ obrazového snímače;
- schopnost zobrazení špatně osvětlené scény;
- možnost výměny objektivů;
- rozlišení videa;
- velikost souboru a propustnost;
- kompenzace světla na pozadí;
- schopnost zachycení pohybujících se objektů;
- další vylepšení obrazu.

2.4.2 Progresivní skenování

S příchodem digitálních kamer a LCD monitorů se začalo využívat progresivního skenování obrazu. Pokud obraz rozdělíme na jednotlivé řádky, tak při progresivním skenování se přenáší vždy celý snímek najednou. Je to rozdíl oproti prokládanému skenování, kdy se přenášejí a zobrazují nejdříve liché řádky a v druhém snímku sudé řádky. Prokládaný režim se používá u analogových CCTV kamer a televizních přijímačů. Hlavní nevýhodou prokládání je špatné zachycení pohybu v obraze. V jednom okamžiku pohyb zachytí pouze polovina řádků, na druhé polovině je zaznamenán až v dalším snímku. To má za následek rozmazání pohybujících se objektů v obraze. Zjednodušeno z [11].

Progresivní skenování má tak velkou výhodu právě v zachycení celého snímku najednou. Počítačové monitory nemusejí obraz prokládat, čímž nenastává efekt, kdy obrazovky blikají. Nedochozí zde k rozmazání pohyblivých objektů v obraze, což je důležité například u dohledových kamerových systémů. Přechzení SPZ značky rychle jedoucího auta by jinak nebylo možné [11].

2.4.3 Kompenzace světla na pozadí

Kompenzace světla na pozadí souvisí s citlivostí kamery na světlo. Kamera se ve snaze přizpůsobit úroveň světlosti nejlépe pro lidské oko může někdy dostat do problémů. Vše lze demonstrovat na příkladu, kdy do kamery umístěné na parkovišti posvítí světla automobilu. Kamera zachytí intenzivnější zdroj světla a kontrola expozice obraz přizpůsobí jeho ztmavením. Výsledkem je tmavý obraz a vidět budou jen zmíněná světla automobilu. Aby se tomuto jevu zabránilo, používá se kompenzace světla na pozadí. Kamera se snaží v obraze ignorovat malé oblasti zvýšeného zdroje světla, nezahrnuje je tak do celkové expozice, která tak zůstává stále stejná. Pokud tedy tuto funkci bude kamera obsahovat, bude pak možné například přechíst SPZ značku vozidla, což by z původního záběru díky jeho tmavosti možné jistě nebylo.

2.4.4 Snímková frekvence

Snímková frekvence udává, kolik jednotlivých snímků kamera zachytí za jednu sekundu. Vyšší snímková frekvence více zatěžuje kameru. Vytváří i větší objemy dat, které je nutno po síti přenášet a snižuje tak propustnost sítě, hlavně pak v případě

použití více kamer. Je proto vhodné zvolit dostačující snímkovou frekvenci pro zachycení obrazu. Pro plynulé video je dobré použít snímkovou frekvenci nad 25 snímků/s. Ovšem za předpokladu, že takovéto video budeme i nahrávat, musíme zabezpečit dostatečné kapacity pro objem dat v ukládacích zařízeních. Když je to možné můžeme přistoupit i k nižší snímkové frekvenci (např. jen 1 snímek/s). Příkladem může být zabezpečení hlídaného parkoviště, kdy je jedna sekunda dostatečná pro reakci hlídače.

2.5 Síťové video

Pod pojmem síťové video rozumíme systém pro sledování a zaznamenávání videa přes IP síť. Síťové video může zahrnovat kromě obrazu i zvuk. Pro přenos dat využívá běžnou síťovou infrastrukturu. Obraz kamery jsou digitalizované přenosy, které můžeme sledovat kdekoliv v síti nebo pomocí sítě internet, třeba i na druhém konci světa. Síťové video můžeme využít v těchto dvou základních kategoriích:

- **Zabezpečovací dohled**

Pokročilé funkce síťového videa z něj dělají skvělý nástroj pro zabezpečovací dohled. Pružnost digitální technologie zlepšuje schopnosti zabezpečovacích pracovníků chránit majetek a osoby. Proto jsou systémy síťového videa zvláště přitažlivé pro firmy, které v současné době používají CCTV, jak je vysvětleno v [7].

- **Vzdálený monitoring**

Síťové video umožňuje uživatelům shromažďovat informace ze všech klíčových míst a sledovat je v reálném čase. Proto je tato technologie ideální pro lokální i vzdálené monitorování vybavení, lidí a míst. Použití se různí od monitorování dopravy a výrobní linky až po monitorování několika skladišť na různých místech, vysvětluje [7].

Hlavními oblastmi použití jsou doprava, obchody, průmysl, státní sektor, bankovní sektor a vzdělávání.

2.5.1 Komprese obrazu

Hlavní myšlenkou komprese je snížení datové velikosti statického obrázku či videa při změně obrazu v podobě vynechání informace nepostřehnutelné pro lidské oko.

Základní rozdělení kompresních metod je na ztrátové a bezztrátové. Při bezztrátové kompresi je každý pixel nezměněn a po dekompresi je obraz v původním stavu. Nevýhodou je omezený kompresní poměr, tedy i vysoká datová náročnost, z čehož plyne nevhodnost použití pro síťové video. Opakem je ztrátová komprese, kdy se z obrazu vypouští informace, které lidské oko nepostřehne. Výrazně se tak zvýší kompresní poměr a klesne datová náročnost.

U kamerových systémů se setkáváme především s kompresí ztrátovou. Pro digitální video přenášené po síti je nutná jeho komprese. Komprimované video se rychleji přenese po síti a také zabírá méně místa na pevném disku. Existuje několik standardů pro kompresi videa i statických obrázků. Zjednodušeno z [10].

2.5.2 Standardy pro kompresi statických obrázků

JPEG (Joint Photographic Experts Group international) je nejrozšířenější standard pro kompresi obrázků. Má širokou podporu ze strany programů i zařízení. Umožňuje různé nastavení úrovně komprese, na které závisí poměr kvality a datové velikosti obrazu. Na úroveň kompresního poměru má vliv i barevnost snímané scény. Čím více barev v obraze, tím menší kompresní poměr může být použit pro zachování určité kvality obrazu, z čehož plyne i vyšší datová náročnost [10].

JPEG2000 je následovník JPEG. Má omezenější podporu ze strany programů a zařízení. Při vyšším poměru komprese vykazuje lepší výsledky než formát JPEG [10].

2.5.3 Standardy pro kompresi videa

Motion JPEG je video poskládané ze sekvence JPEG snímků. Síťová kamera zachytává například 30krát za sekundu jednotlivé snímky, které se komprimují do formátu JPEG. Pokud je těchto obrázků za sekundu dostatečné množství, tak je vnímáme jako video, tedy Motion JPEG. Každý jednotlivý obrázek má kvalitu podle zvolené komprese. Díky své jednoduchosti je to velmi používaný formát pro síťové video. Umožňuje další zpracování obrazu, jako je detekce pohybu či sledování objektu. Disponuje širokou možností výběru rozlišení a zaručuje neměnnou kvalitu obrazu bez ohledu na složitost snímaného záběru. Možnost volby úrovně komprese, a tím i velikosti datového toku umožní přizpůsobení propustnosti sítě. Však díky nepřítomnosti video komprese je tento datový tok přenášený po síti poměrně velký [10].

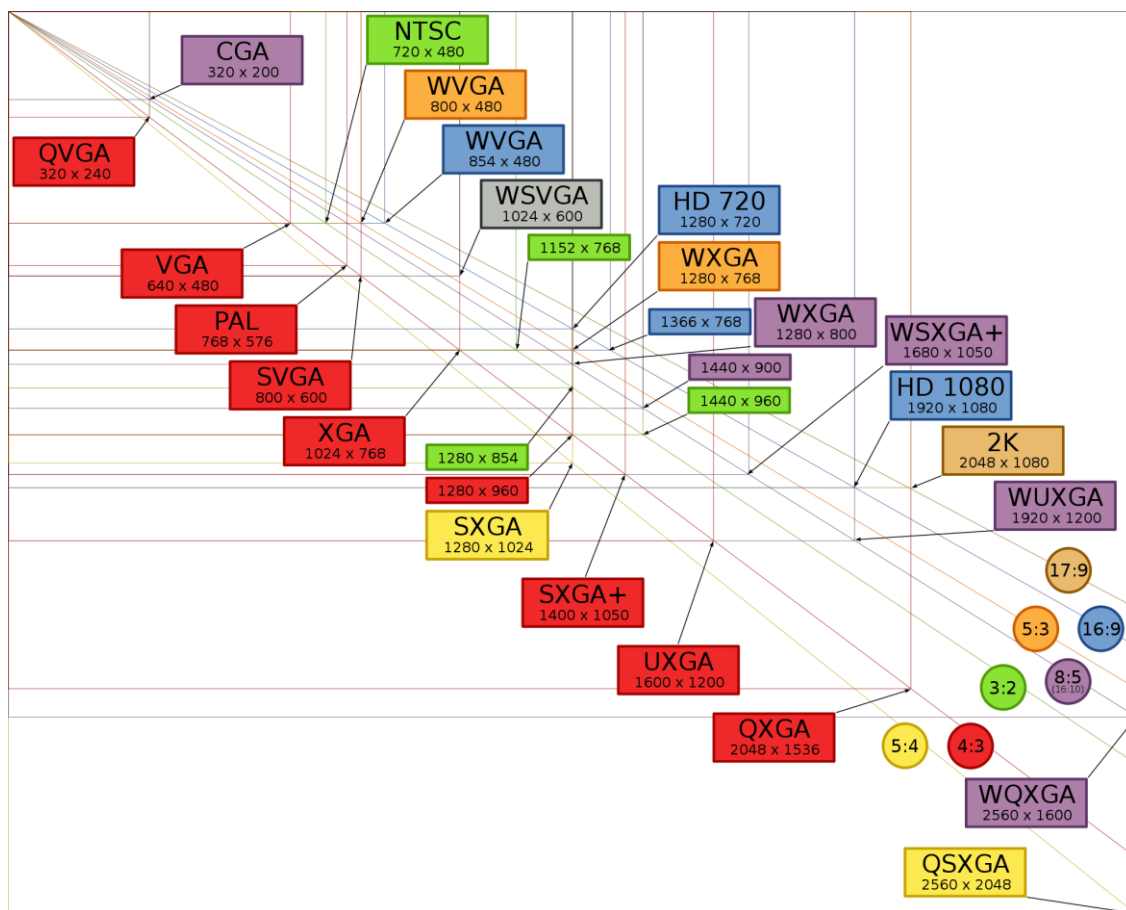
MPEG (Motion Picture Experts Group) je jedním z nejpoužívanějších standardů pro audio a video kompresi. Základním principem je porovnávání jednotlivých snímků mezi sebou. První komprimovaný snímek je referenční. U dalších snímků se použijí pouze data z částí obrazu, které se změnily. Výsledný obraz je složen z prvního snímku a dat, ve kterých jsou zaznamenány rozdíly v obraze. Takto je i výsledný obraz přehráván. Tento standard zahrnuje ovšem i několik dalších technik, jako je identifikace předmětů a předvídání jejich pohybu v obraze. Výhodou je především menší datová náročnost výsledných souborů než při použití Motion JPEG. Dobrá je kvalita obrazu při použití nižší bitové rychlosti při vyšším počtu snímků za sekundu. Menší datové nároky jsou však kompenzovány složitostí kódování a dekódování, což má za důsledek vyšší latenci, nebo-li dobu od zachycení po zpřístupnění záznamu [10].

MPEG existuje v několika verzích. Za verze MPEG-2 a MPEG-4 je nutno platit licenční poplatky. U síťových kamer se dnes používá MPEG-4, především pro snížení bitové rychlosti při stejné kvalitě obrazu.

H.263 je standard určený především pro přenosy videa se stálou bitovou rychlostí. Je to standard původně navržený pro video hovory, kde je konstantní rychlost přenosu důležitá. V oblasti síťového videa je jeho nevýhodou především snížení kvality obrazu při pohybu objektu, kdy by bylo potřeba bitovou rychlost zvýšit [10].

2.5.4 Rozlišení obrazu

Rozlišení obrazu, který je schopna kamera zachytit, se uvádí jako poměr horizontálních a vertikálních pixelů. U digitálních IP kamer je to počet efektivních pixelů snímače. Z analogových kamer, kde se obraz skládal z řádků, je známo PAL rozlišení 704×576 řádků. U digitálních kamer převládá rozlišení 640×480 pixelů zvané VGA (Video Graphics Array). V dnešní době se používají jeho násobky, např. QVGA či XVGA. Jednotlivá rozlišení jsou uvedena v Obr. 2.2.



Obr. 2.2: Rozlišení obrazu (použito z [16])

2.6 Síťové kamery a jejich využití

Síťová kamera vysílá zachycené záběry skrze datovou síť. Umožňuje lokální nebo vzdálený přístup, správu, či nahrávání záznamů za pomoci standardní síťové infrastruktury. Tyto kamery jsou nejčastěji do sítě připojeny pomocí technologie Ethernet nebo bezdrátové Wi-Fi. Kromě zobrazování a zaznamenávání obrazu dokáže síťová kamera přenášet i jiná data. Mezi další funkce patří například detekce pohybu v obraze, přenos zvuku, mechanismy pro ovládání kamery (pohyb, zoom), ale také digitální vstupy a výstupy. Ty slouží pro spuštění alarmu nebo rozsvícení světel. Kamera může umět poslat i obrazová data při detekci pohybu na email nebo mobilní telefon. V následujícím rozdělení kamer je vždy uveden i obrázek ilustrující typického zástupce dané kategorie.

2.6.1 Vnitřní kamery

Vnitřní kamery jsou určeny především pro použití ve vnitřních prostorech (kanceláře, chodby, obchody, sklady a garáže). Pro venkovní nasazení nejsou určeny především díky jejich konstrukci. Plastové materiály, z nichž jsou většinou vyrobeny, nejsou odolné vůči slunci, dešti, vysokým či nízkým teplotám. Snímání kamery je nastaveno tak, aby poskytovala dobrý obraz za denního světla i za umělého osvětlení. Speciální kamery s odnímatelným infračerveným filtrem umožňují zachycení černobílého obrazu i za velmi špatných světelných podmínek. Kamery s infračervenou přisvětlovací diodou jsou schopny snímat i za úplné tmy. Zástupce těchto kamer – AXIS 210A na Obr. 2.3.



Obr. 2.3: Vnitřní IP kamera AXIS 210A (použito z [13])

2.6.2 Venkovní kamery

Venkovní kamery se od vnitřních kamer liší především použitím krycí konstrukce (viz Obr. 2.4). Některé vnitřní kamery po dokoupení krytu lze používat jako venkovní. Pro kryty kamer jsou použity více odolné materiály (speciální technický polymer, kov) odolávající prachu, silnému dešti, UV záření, vysokým a nízkým teplotám nebo i zničení od různých živočichů. Snímání je nastaveno pro přirozené světlo. Zpravidla jsou použity čočky s automatickou clonou. Objektiv kamery by měl být co nejbližší sklu krytu, aby nedocházelo k odleskům. Kamera by měla být umístěna tak, aby v záběru nebylo přímé sluneční světlo, které může poškodit infračervený filtr. Neměla by snímat oblohu, aby nedocházelo k přílišnému zvýšení kontrastu. Pokud by bylo nutné používat kameru i v noci, přídatné infračervené světlo musí být umístěno tak, aby žádným negativním způsobem neovlivňovalo obraz kamery. Opět musí být zabezpečeno krytem.



Obr. 2.4: Venkovní kamera Edimax IC-9000 (použito z [2])

2.6.3 Ovladatelné kamery

Ovladatelné kamery, včetně zástupce na Obr. 2.5, přinášejí možnosti otáčení kamery do osmi základních směrů. Otáčení je možné například až v rozsahu 360° vertikálně a 120° horizontálně. Mezi další funkce patří přibližování či oddalování obrazu. Některé kamery umožňují i předem naprogramovat, jaký pohyb budou vykonávat, čímž lze jednoduše rozšířit možnost snímání větší plochy okolí. Vše je jednoduše ovladatelné pomocí aplikace nebo internetového prohlížeče ze vzdáleného místa přes datovou síť.



Obr. 2.5: Ovladatelná kamera Edimax IC-7000PT (použito z [3])

2.6.4 Dome kamery

Dome kamery jsou diskrétní IP kamery pro dohledové systémy uvnitř budov, kde jsou určeny především pro montáž na strop. Další možnosti použití nacházejí v dopravních prostředcích. Vyznačují se především malou výškou a větší šířkou své konstrukce. Tyto typy kamer navíc poskytují základní odolnost proti vlhkosti a prachu. Zástupcem je kamera AXIS 209FD-R z Obr. 2.6.



Obr. 2.6: Dome kamera AXIS 209FD-R (použito z [14])

2.7 Možnosti zapojení více kamer

Možnosti zapojení více kamer lze využít především pro zabezpečení většího objektu. Při zapojení více kamer do síťového kamerového systému je potřeba síťových přepínačů. Jednotlivé kamery jsou připojeny datovým kabelem přes rozhraní Ethernet do portů přepínače. Na port přepínače je připojeno i obslužné PC nebo server.

Výhodou použití tohoto řešení je jeho škálovatelnost. Je jednoduché přidat další kameru či obslužné PC. Omezení jsme jen počtem portů přepínače. V případě potřeby lze do systému přidat i další přepínač a rozšiřování může pokračovat.

Systém má i další výhody. V některých případech je možno pro síťový kamerový systém použít stávající infrastrukturu datových kabelů. To umožňuje napojení i na jiné systémy pro více funkcí a jejich snadné monitorování. Příkladem může být systém pro správu budov, který při detekci pohybu v místnosti sám zapíná klimatizaci nebo topení.

Pro zajištění funkčnosti rozsáhlého systému je nutno zajistit dostatečnou datovou propustnost pro požadované video přenosy. Datová úložiště musejí mít dostatečné kapacity pro ukládání videí. Jednotlivé komponenty zabezpečit proti krádeži a poškození. Je vhodné použít technologii zabezpečení proti výpadku elektrického proudu, která spolu s technologií PoE zajistí funkčnost i při nenadálé události.

2.8 Komunikace kamery a PC

2.8.1 Přenos dat

Pro připojení kamery do sítě je nutné, aby měla svojí vlastní IP adresu. IP adresu tvoří čtyři čísla v rozsahu 0 – 255 oddělená tečkou. Na základě této adresy je kamera v síti identifikovatelná a můžeme přistupovat k funkcím této kamery.

Je také několik možností jak data do sítě vyslat [9]:

- unicast – data od odesílatele jsou adresovány pouze jednomu příjemci;
- multicast – komunikace mezi jedním odesílatelem a několika příjemci;
- broadcast – odesílatel posílá data všem bodům v síti.

2.8.2 Zabezpečení přenosu dat

Zabezpečení přenosu dat je velmi obsáhlé téma přesahující rozsah této práce. Stručně jde o znemožnění přístupu k datům uživateli, pro kterého nejsou data určena. Příkladem v síti IP kamer je použití hesla pro přístup do systému kamer, do obslužných programů a monitorovacích zařízení. Vysoce zabezpečené systémy mohou používat šifrování přenosu dat z kamer.

Základem při začlenění bezpečnostních kamer do provozu je také zabezpečení samotných zařízení. Je nutné zabezpečit jednotlivé komponenty v systému proti krádeži, poškození i přírodním vlivům. V případě nahrávání obrazu je vhodná datová záloha systému. Dobré je použití i zabezpečení systému proti výpadku elektrického proudu.

2.8.3 Přenosové protokoly

Dnes nejpoužívanějším síťovým protokolem je IP (Internet Protokol). Tento protokol se používá v datových packetových sítích a funguje jako zastřešující protokol pro ostatní protokoly.

IP používá dva standardní protokoly: TCP (Transmission Control Protocol) a UDP (User Datagram Protocol), které se liší především poskytovanými službami. TCP poskytuje spolehlivý přenos, velké objemy dat rozdělí do menších packetů a ručí za jejich doručení od vysílače k příjemci. UDP protokol negarantuje doručení všech dat a kontrola chyb v přenosu je závislá na aplikaci.

Při komunikaci kamery, která vystupuje v této komunikaci jako server a počítače, zastupujícím klienta, bude hlavním protokolem HTTP. Kamera přes tento protokol přijímá CGI příkazy, vysvětleny níže v kapitole 4.4. Na základě těchto příkazů vrací klientovy odpovědi v podobě vyžádaných dat. Případně jinak reaguje na daný příkaz, například natočením kamery do určitého směru. Základní přenosové protokoly lze najít v Tab. 2.1. V této kapitole bylo čerpáno z [9].

Tab. 2.1: Přenosové protokoly

Protokol	Přenosový protokol	Port	Použití
DDNS	TCP	N	Protokol, který umožňuje v reálném čase aktualizovat záznamy o internetové doméně DNS serveru. V případě síťových kamer se jedná o připojení ke kameře přes známou DNS adresu, ne přes IP adresu, složitou na zapamatování.
FTP	TCP	20/21	Protokol určený pro přenos souborů přes datovou síť. Používá se pro přenos obrázků a video nahrávek z IP kamer do úložného prostoru FTP serveru, popřípadě do úložného prostoru vyhrazeného pro obslužnou aplikaci.
HTTP	TCP	80	Základní použití tohoto protokolu je pro prohlížení webových stránek, tedy pro jejich příjem z webového serveru. V případě síťového videa kamera funguje jako webový server a zpřístupňuje video, případně i ovládací prvky přes webový prohlížeč.
HTTPS	TCP	443	Jedná se o zabezpečenou verzi protokolu HTTP, která se používá pro zabezpečený přístup k webovým stránkám. Používá šifrovací technologii pro zabezpečený přenos videa ze síťové kamery. Je možno použít i autentifikaci pomocí digitálních certifikátů.

PPPoE	TCP	N	Protokol umožňující vytvořit virtuální spoj mezi dvěma síťovými zařízeními. Vytvoří tak v packetové síti zabezpečený virtuální okruh.
RTP	TCP/UDP	N	Standardizovaný protokol pro poskytování videa a zvuku v packetových datových sítích. Používá se pro přenos digitálního videa a zvuku z kamer, ale také pro různé sdílení (steam) médií, popřípadě pro videokonference ve VOIP telefonii.
SMTP	TCP	25	Protokol pro odesílání emailových zpráv. V případě, že kamera umožňuje tento typ komunikace, je možno uživatele upozorňovat emailem na změny v obraze, případně zachycený hluk. Lze posílat JPEG obrázky či nahrané sekvence videa.
UPnP	TCP/UDP	5000/ 1900	Tento protokol umožňuje co nejjednodušší připojení síťových periférií k počítači. Umožňuje automatickou konfiguraci síťových periférií a jejich rychlé fungování bez složitého nastavování.

3 ZNÁMÉ ZPŮSOBY ŘEŠENÍ

Mezi známé způsoby řešení obsluhy a komunikace se síťovou kamerou patří obslužný software. Software pro komunikaci kamery a PC je program zpřístupňující funkce kamery na počítači. V programu je možno zobrazit video z kamery, popřípadě je zde možnost jeho nahrávání a ukládání na pevný disk. Nejčastěji obsahuje i několik funkčních tlačítek pro obsluhu kamery. Program tak zpřístupňuje nastavení jasu, kontrastu, přiblížení, oddálení nebo pohyb kamery. Pro kamerové systémy s větším počtem kamer je zde možnost sledování více obrazů najednou. Například pro systém čtyř kamer se použije matice 2×2 . V praxi může být i systém s mnohem více kamerami. Tato kapitola se věnuje především originálnímu softwaru dodávanému ke kamerám Edimax.

3.1 Ovládací programy kamer Edimax

Ovládání kamer výrobce Edimax je řešeno dvěma způsoby. Jedním z nich je ovládací software pro zabezpečovací systémy IPCam Surveillance Software. Druhý způsob představuje ovládání přes internetový prohlížeč.

3.1.1 IPCam Surveillance Software

IPCam Surveillance Software je ovládací program, jehož využití najdeme především v dohledovém PC. Program je přiložen ke kameře na CD, případně ke stažení z internetových stránek výrobce. Aplikace umožňuje ovládání až šestnácti kamer. Splňuje základní požadavky pro záznam obrazu a fotografií. Detekuje pohyb v obraze. Významnou možností je i plánování nahrávání videa z kamer a jejich nastavení. Program dále umožňuje konfiguraci automatického pohybu kamer. Halvní okno programu je zobrazeno na Obr. 3.1.

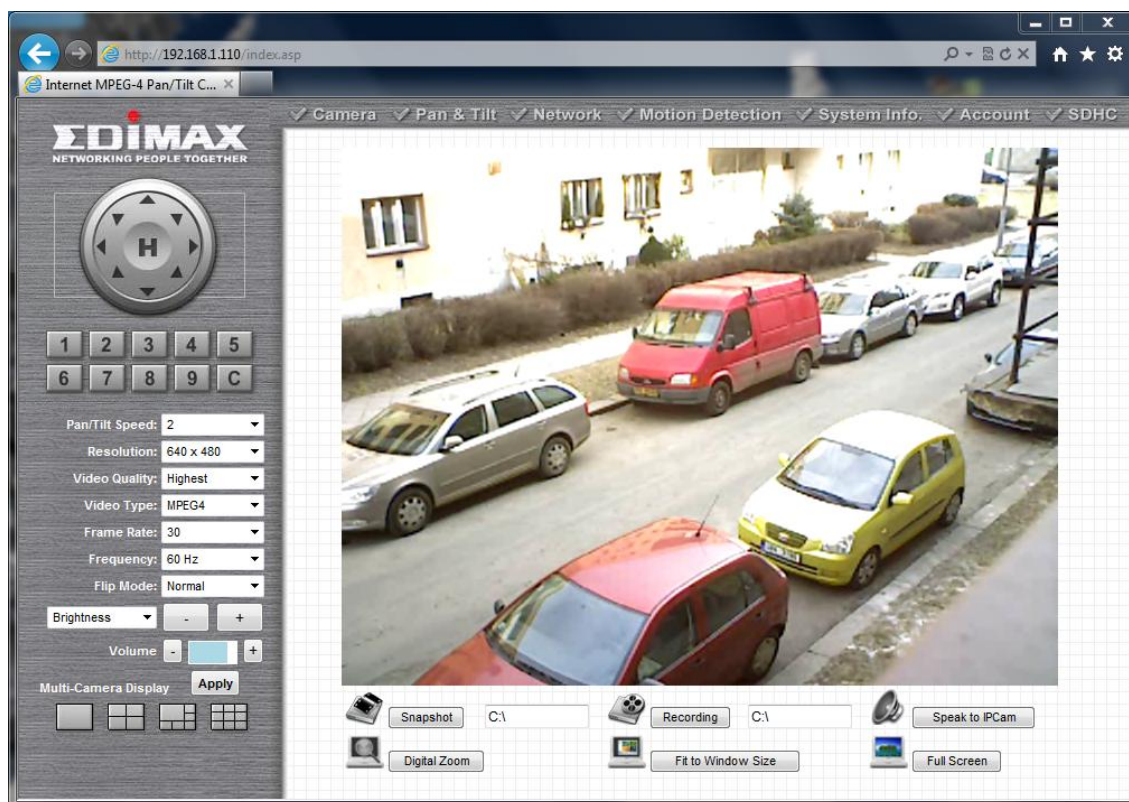


Obr. 3.1: IPCam Surveillance Software (zdroj: autor)

Základní okno programu umožňuje na ovládacím prvku imitujícím dálkový ovladač přepínat počet zobrazených obrazů kamer. Na obrázku je vybrán režim osmi kamer s jednou hlavní, zvětšenou. Ve spodní části ovladače lze s vybranou kamerou pohybovat, případně přibližovat a oddalovat obraz. Ve spodní části programu jsou čtyři tlačítka sloužící pro nahrávání, nastavení, přehrávání a pořízení fotografie. V levém dolním rohu jsou dvě tlačítka pro ukončení a minimalizaci programu.

3.1.2 Ovládání přes internetový prohlížeč

Kamery Edimax nabízí i možnost ovládání pomocí internetového prohlížeče. Při volbě této možnosti je použito přihlášení pomocí uživatelského jména a hesla. Poté se z kamery, jež splňuje funkci serveru, začne načítat internetová stránka s použitím appletu. V internetovém prohlížeči lze poté pracovat se všemi funkcemi kamery. Je přenášén i video stream obrazu z kamery, případně z více kamer. Velmi dobře je zde vyřešeno i pohybové ovládání kamery. Samozřejmostí je možnost volby rozlišení obrazu, standartu MPEG a M-JPEG, komprese obrazu. Mezi další možnosti patří nastavení emailové komunikace kamery, například při detekci pohybu v obraze.



Obr. 3.2: Ovládání přes internetový prohlížeč (zdroj: autor)

Na Obr. 3.2 je vidět internetová stránka, kterou kamera zobrazí po zadání její IP adresy do internetového prohlížeče. Hlavním obsahem této stránky je video z kamery, případně z více kamer. Nad videem je lišta s přístupem k jednotlivým funkcím a nastavením kamery. Pod obrazem najdeme možnost uložení fotografie, nahrání videa, ovládání přiblížení a volbu velikosti obrazu kamery. V levé části se nachází ovladač směrového pohybu kamery a čísla pro volbu jednotlivých kamer, pokud jich je zapojeno více. Pod čísly nalezneme nastavení videa. Lze nastavit rozlišení, kvalitu, formát, snímkovou frekvenci, otočení videa a také úpravu světlosti, saturace a kontrastu. V levém dolním rohu se nachází možnost volby zobrazení matice více kamer.

4 NÁVRH VLASTNÍHO ŘEŠENÍ

4.1 Výběr nejvhodnější kamery

Výběr nejvhodnější kamery závisí především na zadání práce. Pro tuto práci je však důležité, aby kamera splňovala i jiné podmínky, ke kterým patří především dobrá dokumentace od výrobce, hlavně co se týče CGI příkazů. CGI příkazy slouží k ovládání kamery přes protokol HTTP. Důležité jsou pro vlastní ovládací program, bez těchto příkazů by bylo velmi obtížné ovládat kameru ve vlastní aplikaci. Z tohoto ohledu je nejvhodnější výrobce kamer firma Edimax. Je jedním z mála výrobců, který dokumentaci příkazů nabízí volně ke stažení ze svých internetových stránek.

Mimo jiné je nutné brát ohled na poměr cena/výkon. V případě zakoupení dvou kamer to platí dvojnásob. Tento požadavek opět splňuje výrobce Edimax, jelikož vyrábí kamery levné, ale s velkým množstvím funkcí. Požadované rozlišení videa 640×480 pixelů splňují i nejlevnější kamery, to tedy nebude problém.

Z naznačených požadavků byly vybrány jako vhodné dvě IP kamery. Jako první model kamera Edimax IC-3010, která splňuje pouze základní požadavky na funkce kamery pro tuto práci s cenou 1 500 Kč. Pokročilejší kamera Edimax IC-7000PT nabízí mnohem více funkcí. Především možnost motorizovaného otáčení kamerou. Cena je však vyšší, přibližně 4 000 Kč. První z kamer je zobrazena na Obr. 4.1, druhá kamera byla zmiňována výše jako zástupce ovládacích kamer v kapitole 2.6.3 na Obr. 2.5.



Obr. 4.1: IP kamera Edimax IC-3010 (použito z [4])

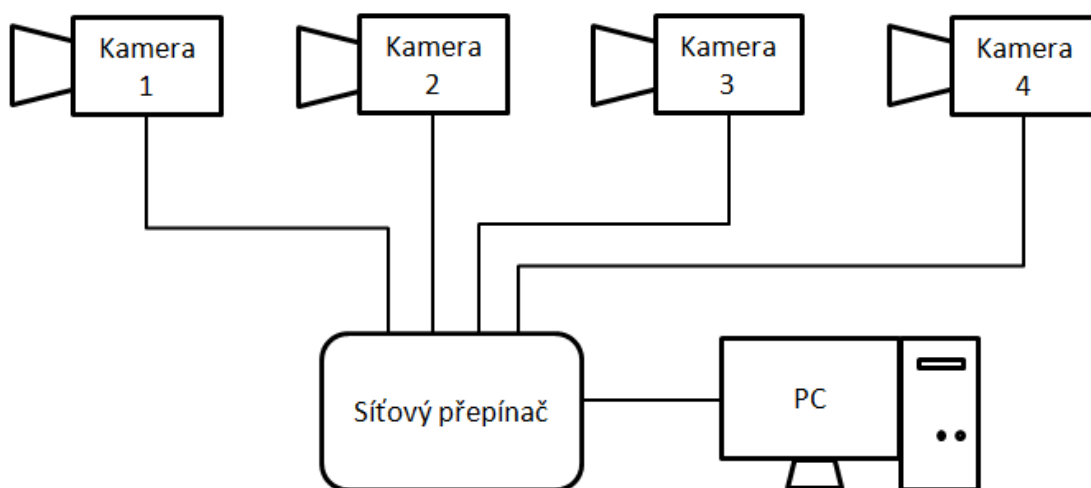
Po zvážení funkcí jednotlivých kamer byla jako nejlepší kamera zvolena Edimax IC-7000PT především díky množství funkcí, které nabízí. Nejzajímavější funkcí je možnost motorizovaného pohybu kamerou v rozsahu 355° horizontálně a 120° vertikálně. Kamera umožňuje přenos a záznam obrazu ve formátech MPEG4, M-JPEG v rozlišení vyšším než 640×480 pixelů. Mezi další funkce patří detekce pohybu v obraze, na které reaguje například zasláním emailové zprávy s fotografií. Z okrajových funkcí lze zmínit obousměrné audio a podporu záznamu fotografií a videa na paměťové SD karty.

4.2 Výběr nejvhodnějšího komunikačního rozhraní

Jako nejvhodnější pro tuto práci lze shledat komunikaci kamery a počítače přes Internet Protokol. Komunikačním rozhraním tak bude jistě Ethernet díky jeho rozšíření mezi počítači a jednotlivými periferiemi. Toto rozhraní pak nabídne jednoduchou možnost připojení kamery k počítači, ale také k rozšiřujícím síťovým prvkům, jako je například přepínač. Ten mimo jiné zajistí případné připojení více kamer.

4.3 Vlastní schéma zapojení

Schéma zapojení naznačuje možný postup při řešení připojení jednotlivých kamer k PC. Možné schéma zapojení čtyř kamer naznačuje Obr. 4.2.



Obr. 4.2: Schéma zapojení čtyř kamer (zdroj: autor)

4.4 CGI příkazy

CGI (Common Gateway Interface) příkazy jsou nejdůležitější pro komunikaci kamery přes vlastní ovládací program. CGI je protokol, který umožňuje propojení externích aplikací a webového serveru. Přenos příkazů a následná reakce ze strany kamery probíhá přes protokol HTTP na portu 80. Příkazy tak lze zadávat i v internetovém prohlížeči. Na tomto principu funguje originální ovládací program kamery. CGI příkazy jsou popsány v dokumentaci kamery Edimax IC-7000PT, kterou lze najít na přiloženém DVD této práce.

Příkladem příkazu bez autentizace může být:

```
Method: Get
URL: HTTP://camera_ip/loginfree.jpg
Example: HTTP://192.168.1.110/loginfree.jpg
```

Reakce kamery na tento příkaz je, že přenese ke klientovi jeden JPEG obrázek. Při tomto příkazu není potřeba být přihlášen pomocí jména a hesla.

Dalším příkladem je příkaz pro otočení kamery:

```
Method: Get
URL: http://camera_ip/camera-cgi/com/ptz.cgi?move=direction
Parameters:
Direction: left, right, up, down, upleft, upright, downleft,
downright, center
Example:http://192.168.1.1/camera-cgi/com/ptz.cgi?move=left
```

Příkaz slouží k otočení kamery v osmi směrech a vrácení kamery do základní polohy (center). Směr určuje parametr direction podle zvoleného příkazu. V příkladu je použit parametr move=left, který posune kameru mírně vlevo. Kamera však tento příkaz provede až po přihlášení jménem a heslem, což bylo nutné v aplikaci ošetřit. Ošetření bylo provedeno příkazem, který kameře pošle přihlašovací jméno a heslo, kdykoliv si jej kamera vyžádá.

4.5 Technické vybavení

Vývoj vlastního programu probíhal na počítači s procesorem Intel Core 2 Solo na frekvenci 1,3 GHz (jedná se o jednojádrový procesor). Počítač má k dispozici 3 GB paměti RAM typu DDR3. Hlavní komponentou byly dvě kamery Edimax IC-7000PT, včetně potřebné kabeláže. K dispozici byl dále síťový router TP-Link TL-WR340G s integrovaným čtyř-portovým přepínačem. Daný přepínač obsahující pouze 4 porty. Byl v tomto případě dostačující, ale pro zapojení 4 kamer by musel být nahrazen víceportovým.

Po připojení kamer Edimax IC-7000PT k routeru se nabízely dvě možnosti přiřazení IP adresy. První možností bylo použití DHCP, kdy router přiřadí kameře volnou adresu z daného rozsahu, nebo přiřadit adresu pevnou. Router přiřazuje adresy zařízením v pořadí, ve kterém se přihlásili. Kamery by tak měly pokaždé adresu jinou. To by znemožňovalo ovládání kamery vlastním programem. Stále by se musela zjišťovat IP adresa kamery a v programu ji znovu a znovu nastavovat. Kameře je přiřazena pevná IP adresa v ovládacím programu kamery i v konfiguraci routeru.

4.6 Programové vybavení

Vlastní ovládací software bude napsán v programovacím jazyce Java. Programovací jazyk Java je objektově orientovaný, vyvinutý firmou Sun Microsystems. Jedná se o jeden z nejvíce používaných programovacích jazyků na světě hlavně díky přenositelnosti mezi různými operačními systémy. Tento programovací jazyk je používán pro několik různých platforem.

Jako vývojové prostředí je zvoleno NetBeans ve verzi 6.8. Vývojové prostředí NetBeans je koncipováno jako open source projekt. Za jeho vývojem stojí především společnost SUN Microsystems. Vývojové prostředí NetBeans slouží především k programování v jazyce Java. Nabízí však možnost programování v jiných jazycích jako je C, C++, PHP a další. NetBeans existuje pro operační systémy Windows, Linux, Mac OS i Solaris.

5 VLASTNÍ NÁVRH PROGRAMU

Prvním krokem vývoje programu je navržení zobrazení obrazu na displeji počítače o rozlišení 800×480 pixelů. Pro jednu kameru stačí obraz o rozlišení 640×480 pixelů, pro dvě či více kamer bude potřeba rozlišení snímaného videa snížit na 320×240 pixelů. Zbývající místo kolem obrazu z kamer bude možno využít pro ovládací prvky (především pro pohybové funkce kamery). Bude potřeba ošetřit i hardwarové prostředky PC (hlavně procesorový výkon bude ovlivňovat plynulost zobrazování při zapojení více kamer). V případě, že nebude možné zachytávat z kamer plynulý obraz, bude se muset omezit především počet zobrazených snímků za sekundu, popřípadě jejich barevná hloubka (komprese).

Program je navržen pro obsluhu čtyř kamer. K dispozici však byly kamery pouze dvě. Bylo tedy nutné zbylé dvě kamery simulovat, proto se v programu zobrazuje obraz každé kamery dvakrát. Kamera číslo jedna je zároveň kamerou číslo čtyři. Kamera dvě je zároveň kamerou číslo tři. Toto číslování je zvoleno proto, aby nebyly v matici zobrazení obrazů ze čtyř kamer dva stejné obrazy vedle sebe.

Při programování byla zvolena angličtina před češtinou jako vhodnější jazyk pro veškeré proměnné, popisky, třídy i metody. Především díky kratším slovům a slovním spojení, které bylo i tak problémové do některých komponent zapsat.

Vlastní ovládací program je rozdělen do několika tříd. Hlavní třída je `Main`, ve které jsou řešeny nejdůležitější příkazy pro ovládání kamery. V grafickém rozhraní `Gui` je navrženo rozmístění ovládacích a zobrazovacích prvků. Dále sem patří čtyři zobrazovací panely `ImageViewer` starající se o vykreslování obrazu.

Ve zdrojovém kódu, který je součástí přílohy, je použito mnoha stručných vysvětlivek, jež objasňují fungování daného příkazu.

5.1 Hlavní třída `Main`

V hlavní třídě `Main` je řešeno především předávání CGI příkazů kamerám, základní nastavení grafického rozhraní `Gui` a metoda `MyAuthenticator` pro komunikaci kamery zabezpečenou přihlašovacím jménem a heslem.

Hlavní třída zahrnuje úvodní implementaci potřebných knihoven a deklaraci proměnných. V základní metodě `main()` je vytvořen nový objekt `Authenticator` a `Gui`, tedy grafické uživatelské rozhraní. Dále je zde definováno chování tohoto grafického rozhraní, velikost okna, viditelnost a akce při ukončení.

V metodě `Action` je řešeno předání různých CGI příkazů kameře a odpověď kamery na tyto příkazy. Toho je dosaženo pomocí vnořeného bloku `try-catch`. Blok obsahuje vytvoření objektů a jejich metod, pomocí nichž se zašle kameře URL adresa. Následně se pomocí objektu `InputStream` a jeho metody získá prostor pro provedení příkazu kamerou. Důležitá je zde proměnná `http` typu `String`, do které je při stisku tlačítka zapsána URL adresa interpretující CGI příkaz. V této metodě je řešen především směrový pohyb kamerou, rychlost pohybu kamerou, velikost rozlišení a volba zrcadlových otočení obrazu.

Třída `NumberOfCameras` se stará o změnu rozlišení obrazu více kamer. Pokud je tedy zvoleno zobrazení z více než jedné kamery je volána tato metoda pro změnu rozlišení na 320×240 bodů. Pokud je přepnuto zobrazení pouze na obraz z jedné kamery jsou připojené kamery přepnuty na rozlišení 640×480 bodů.

Ve třídě `MyAutheticator` je řešeno vyplnění přihlašovacího jména a hesla při komunikaci s kamerou. Heslo je zvoleno jako neměnné. Pro účely této práce nebylo potřeba ke komunikaci využívat heslo vůbec, avšak kamera nemá funkci vypnutí požadování jména a hesla při začátku komunikace. Metoda `MyAutheticator` se tedy stará o vyplnění jména hesla kdykoliv si jej kamera vyžádá.

5.2 Grafické rozhraní Gui

Při návrhu grafického rozhraní bylo nutné počítat s vymezenou plochou 800×480 pixelů. Pokud od této plochy odečteme velikost snímků videa 640×480 pixelů, zbývá pro rozmístění ovládacích prvků pouze 160×480 pixelů. Tato plocha se z počátku jevila jako nedostačující. Avšak po zpracování dostupných příkazů a vhodném zvolení ovládacích tlačítek a přepínačů se podařilo všechny ovládací prvky vměstnat. Bylo nutné zkracovat popisky ovládacích prvků, veškeré funkce však budou vysvětleny níže.

Třída `Gui` byla vytvořena pomocí komponenty `JFrame`. Problém vznikl i při nastavení grafického rozhraní `Gui` na velikost 800×480 . Komponenta `JFrame` při nastavení `Absolute Layout` počítá i s okolním systémovým oknem. Vnitřní okno rozhraní tak nemělo požadovanou velikost. Nastaly dvě možnosti řešení: přizpůsobit ovládací a zobrazovací prvky vnitřní velikosti okna nebo zvětšit velikost celého okna tak, aby vnitřní prostor měl velikost 800×480 pixelů. Jako lepší rozhodnutí se jevílo zvětšit celé okno na 806×508 pixelů. Velikost systémových oken je totiž u operačních systémů různá, a proto bylo lepší zachovat vnitřní velikost, která neomezuje v zobrazení videa.

V třídě `Gui` je implementována pouze jedna knihovna `java.awt.event.*` starající se o funkčnost reakce na stisk tlačítka. Následuje nezbytná deklarace proměnných a vytvoření objektů `ImageViewer`, tedy zobrazovacích panelů. Tyto zobrazovací panely jsou tvořeny komponentou `JPanel`, je zde i nastavení jejich velikosti a viditelnosti. Pro celou třídu `Gui` je implementována metoda `ItemListener`. Ta se stará o zachycení volby z komponent `JComboBox`. Tato volba se ukládá do proměnné `combo` a předává se do proměnných `compres` nebo `fps` podle toho, jestli se mění komprese obrazu nebo snímková frekvence. V konstruktoru `Gui` jsou přiřazeny přepínací tlačítka `JRadioButton` do skupin, aby bylo možné v každé skupině zvolit pouze jeden přepínač.

Hlavním úkolem `Gui` je zajistit komunikaci programu s uživatelem. Stará se o reakce na stisk tlačítek, přepínačů a roletových seznamů. Jako reakci na stisk komponenty se nejčastěji mění proměnná `http`, ale samozřejmě i jiné proměnné. Stiskem tlačítka se většinou volá i metoda z třídy `Main` nebo z tříd `ImageViewer`. Zavoláním metod se provede s kamerou pohyb nebo zobrazení obrázku.

5.3 Zobrazovací grafické prvky `ImageViewer`

Zobrazovací prvek `ImageViewer` se stará o vykreslování obrázků z jednotlivých kamer. Každá ze čtyř kamer má svůj vlastní zobrazovací panel číslovaný podle čísla kamery. Každý `ImageViewer` je řešen pomocí komponenty `JPanel`. `ImageViewer` opět obsahuje import použitých knihoven a úvodní deklaraci proměnných. Nejdůležitější jsou proměnné `image` interpretující obrázek a `time` vysvětlena níže.

Hlavním úkolem grafické komponenty `ImageViewer` je vykreslování obrázků. K tomu je potřeba, aby obsahovala pět metod. Metodu `setImage` pro uložení obrázku do `Gui` a metodu `paint` pro vykreslení. Dále samozřejmě metodu, která získá obrázek z kamery. To bylo řešeno opět CGI příkazem pro zobrazení jednoho JPEG obrázku. Pro načtení obrázku bylo nutné importovat knihovnu `javax.imageio.ImageIO` a použít příkaz `image=ImageIO.read(url)`, který se stará o uložení obrázku z kamery do proměnné `image`.

Největší problém však nastal při načítání videa, řešeným periodickým načítáním JPEG obrázku. Při implementaci cyklu `while`, který opakoval sérii příkazů v bloku `try-catch`, došlo k zamrznutí programu. Program nereagoval na žádné tlačítko, ani na vypnutí. Postupem času bylo jasné, že program je natolik zaměstnán načítáním obrázků z kamer, že nedokáže provést už nic jiného. Řešením této situace bylo začlenění více vláken `Thread`. Bylo tedy nutné pro každý `ImageViewer` vytvořit vlastní vlákno, které se spustí při periodickém načítání obrázků. To vyžadovalo implementaci rozhraní `Runnable` s metodou `run()`. Byla tedy vytvořena nová metoda `Video`, která je volaná při stisku tlačítka `VIDEO` a stará se o vytvoření a start nového vlákna. Při startu nového vlákna je volaná metoda `run()`, která provádí periodické načítání JPEG snímku.

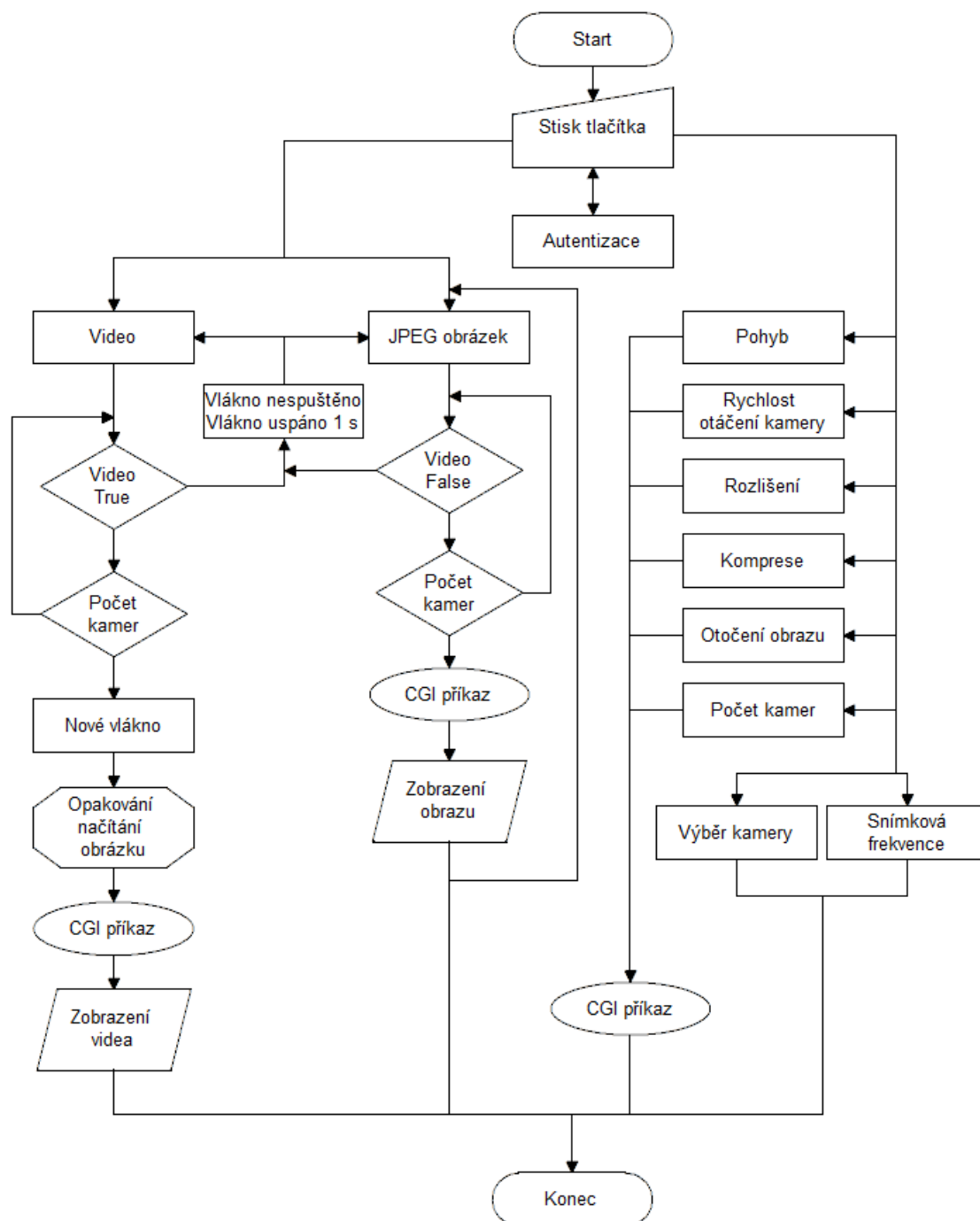
Periodické načítání snímku bylo potřeba ošetřit omezením počtu snímků za sekundu. Toho bylo dosaženo pomocí usnutí vlákna příkazem `Thread.sleep(time)`. Proměnná `time` je závislá na výběru počtu snímků z rozbalovacího seznamu `FPS` (např. při volbě snímkové frekvence 10 snímků za sekundu se vlákno uspává na 100 ms).

Dalším problémem bylo pozastavení vlákna při jeho nevyužívání. Příkazem `wait()` totiž vlákno vytěžovalo procesor na 100 %. Řešením bylo při nevyužití vlákna toto vlákno uspat na 1000 ms a po probuzení neprovádět žádný příkaz. Tímto způsobem je procesor od nevyužívaných vláken vytížen na 7 %.

Komponenty `ImageViewer` s čísly 2, 3 a 4 fungují stejně jako první. Obsahují stejné metody a proměnné. Pouze se metody, jež obsahují, volají při použití zobrazení více kamer. Stejně tak se využívají, respektive nevyžívají, v nich spouštěná vlákna.

5.4 Vývojový diagram

Základní a velmi zjednodušený vývojový diagram programu zobrazuje Obr. 5.1.

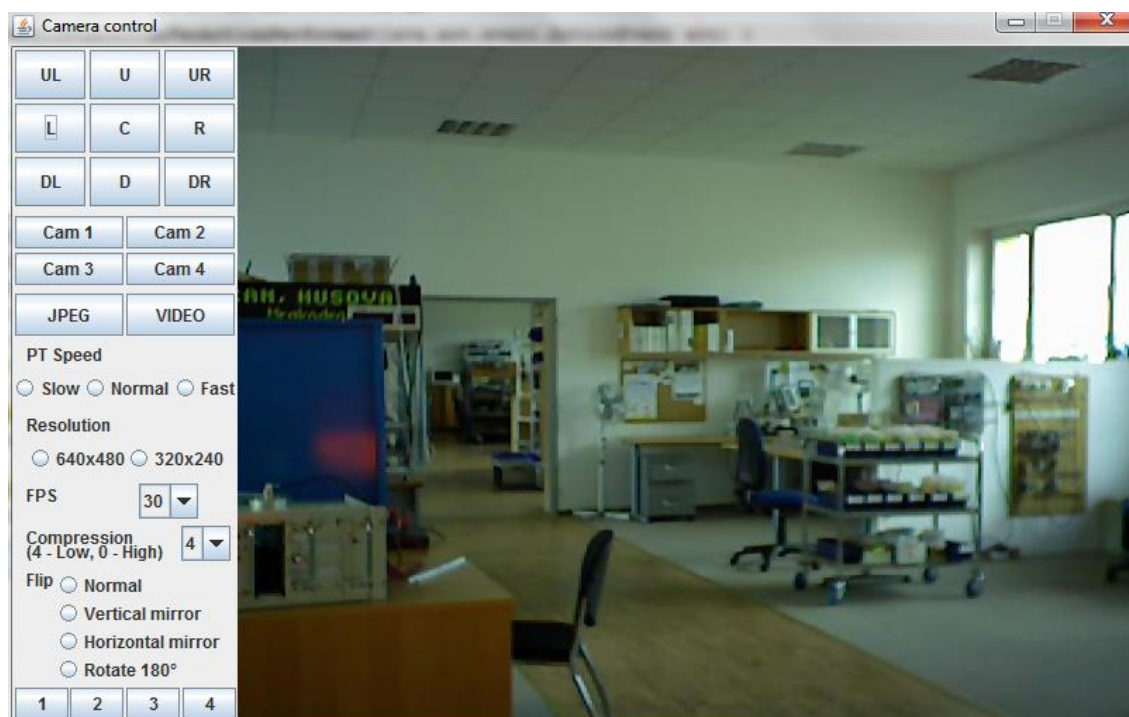


Obr. 5.1: Základní vývojový diagram (zdroj: autor)

6 VÝSLEDKY ŘEŠENÍ

6.1 Popis programu

Funkčnost programu můžeme rozdělit na dvě části: část ovládací a část zobrazovací. Výchozí okno programu s obrazem jedné kamery je ukázáno na Obr. 6.1.



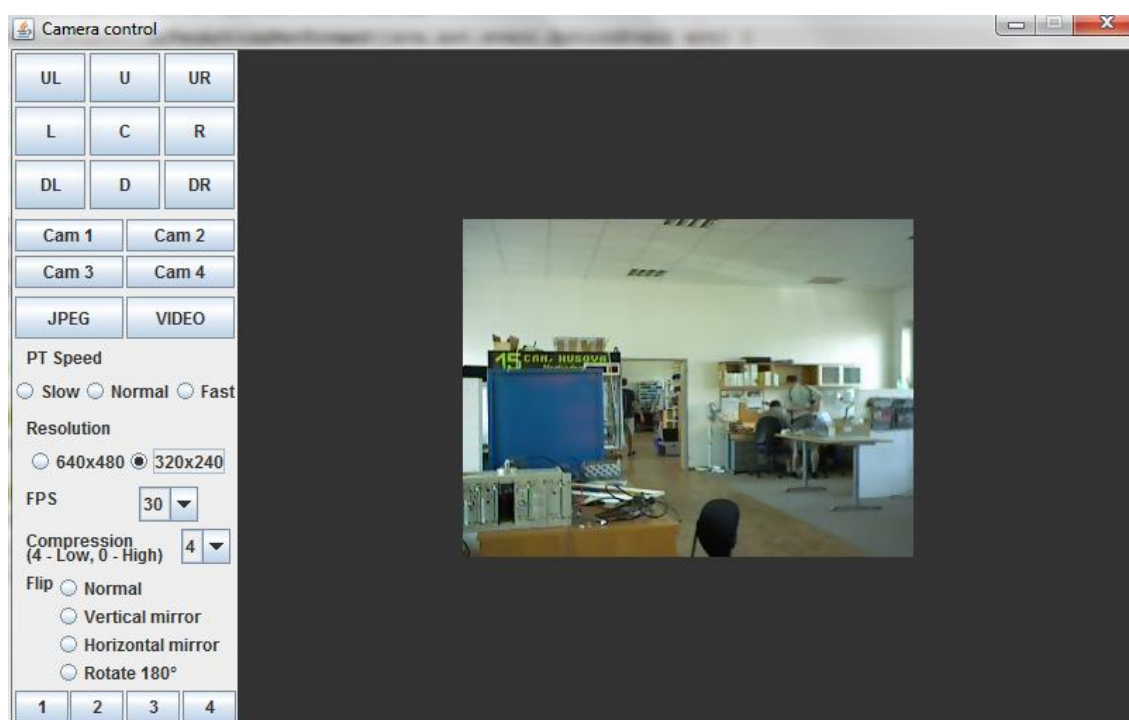
Obr. 6.1: Výchozí okno programu (zdroj: autor)

V levé ovládací části se nachází matice 3×3 tlačítek ovládajících směr otočení kamery. Zkratky znamenají směr natočení kamery, prostřední tlačítko Center vrátí kameru do výchozí polohy. Pod těmito tlačítky se nachází blok čtyř tlačítek označených jako Cam 1 až 4. Tyto tlačítka slouží k výběru ovládané kamery.

Tlačítka JPEG a VIDEO slouží k zachytávání obrázku, respektive videa. Program umožňuje zachycení statického snímku i při právě probíhajícímu načítání videa, video je v tu chvíli přerušeno, tedy neběží na pozadí. Je načten statický obrázek a po stisku tlačítka VIDEO se zase obnoví periodické snímání. V zobrazení více kamer lze i pro jednotlivé kamery získat obrázek, tedy pozastavit obraz i znovu obnovit video.

Tři přepínače PT Speed slouží k rychlosti pohybu kamery, tedy jak rychle se kamera otočí to zvolené pozice. To ovlivní i slyšitelnost tohoto pohybu. V programu jsou zapracovány jen tři nejrychlejší. Kamera však nabízí i pomalejší. Tyto pohyby trvají několik sekund, proto jsou nepoužitelné, není problém je do programu později začlenit.

Následující dva přepínače umožňují nastavení rozlišení obrazu z kamery. Tyto tlačítka fungují pouze při zobrazení obrazu z jedné kamery, tak jak to ukazuje Obr. 6.2. Při zobrazení obrazů z více kamer jsou kamery automaticky nastaveny na rozlišení 320×240, aby bylo možné do dané plochy více obrazů vůbec vměstnat.

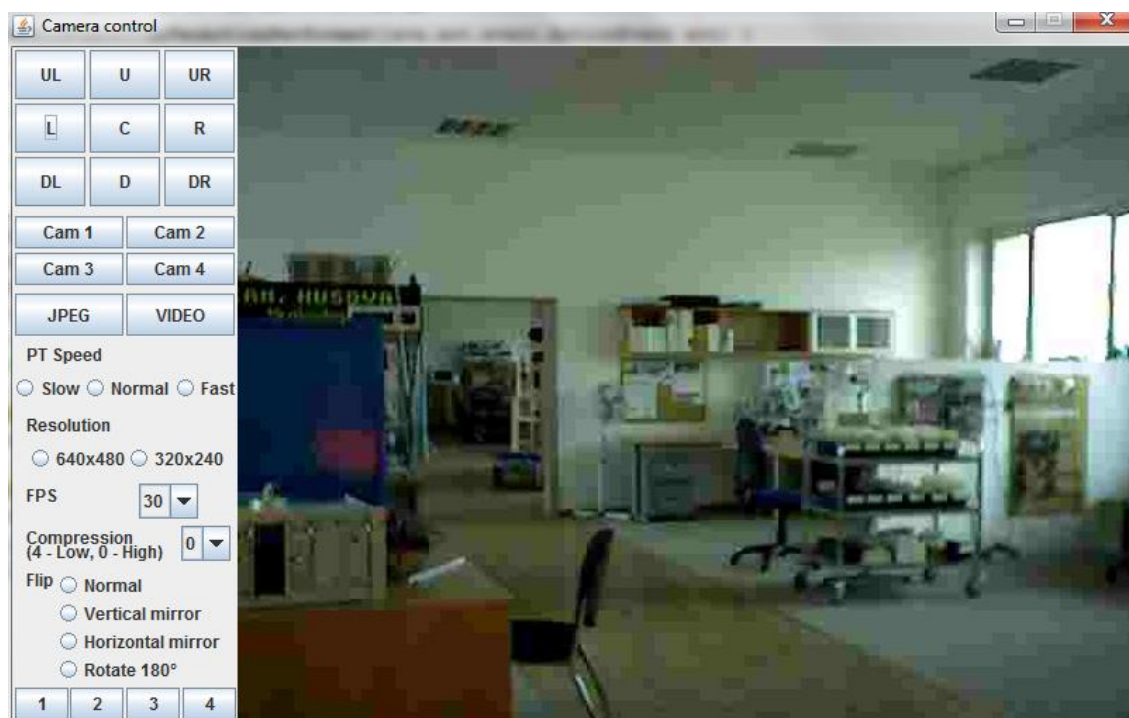


Obr. 6.2: Zobrazení obrazu jedné kamery v nižším rozlišení (zdroj: autor)

Rozbalovací seznam FPS (Frame per second) slouží pro nastavení snímkové frekvence videa. Snímková frekvence je měněna vždy pro všechny kamery najednou. Ovlivňuje rychlost vyčítání JPEG obrázku z kamery a jeho zobrazení. Zvolené hodnotě odpovídá čas v milisekundách, na který se uspí vlákno programu starající se o vyčítání snímku z kamery, tak aby byl zobrazen právě daný počet snímků. V programu bylo dosaženo zobrazení 30 snímků z jedné kamery v rozlišení 640×480. Stejně hodnoty bylo dosaženo i při zobrazení videa ze dvou kamer. Při volbě tří nebo čtyř kamer jsou

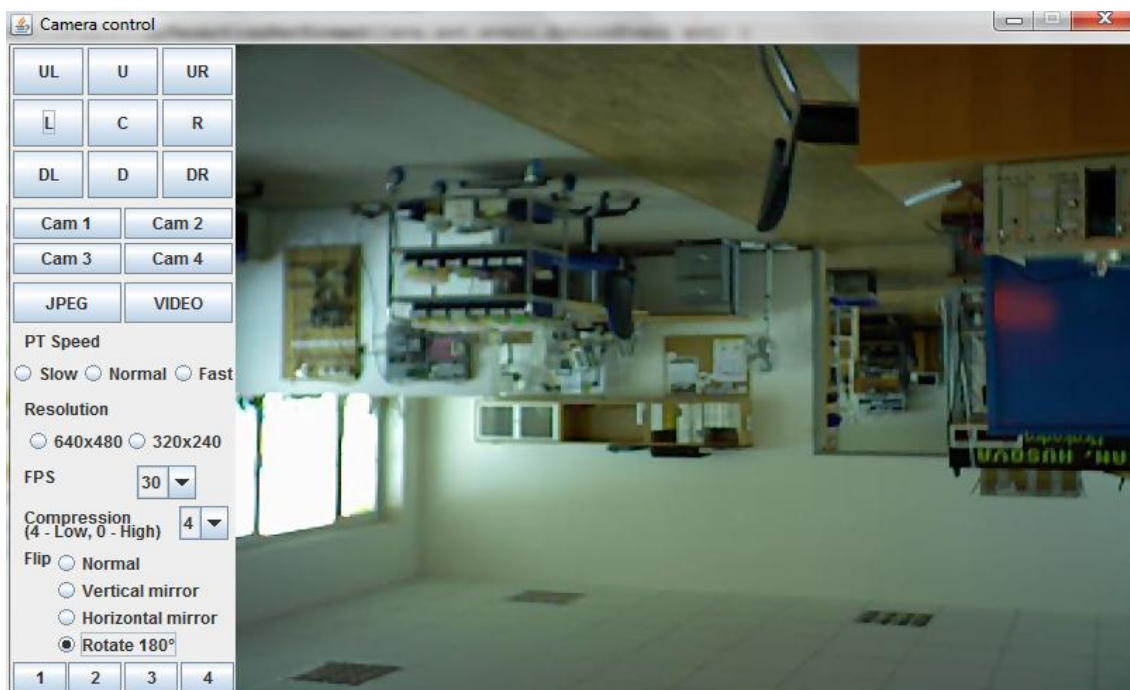
další obrazy načítané pouze jedné respektive dvou kamer. Snímkovou frekvenci tedy nebylo možné zjistit, jelikož se vždy dvě vlákna podílejí o vyčítání jednoho snímku jedné kamery. Snímková frekvence tak teoreticky spadne na 15 snímků pro jednotlivé obrazy. Pro ověření možností nastavení snímkové frekvence by bylo potřeba čtyř kamer.

Kompresi JPEG snímků se nastavuje vybráním hodnoty z rozbalovacího seznamu. Je možno vybrat pět hodnot od nejnižší po nejvyšší kompresi JPEG snímku. Nejnižší hodnoty pod čísly 4 a 3 jsou téměř k nerozeznání a obraz je na dobré úrovni kvality. Naopak při nejvyšší kompresi je obraz už velmi nekvalitní, což je ukázáno na Obr. 6.3. Ovlivnění datového přenosu se na frekvenci snímků či vytížení procesoru počítače neprojevovalo.



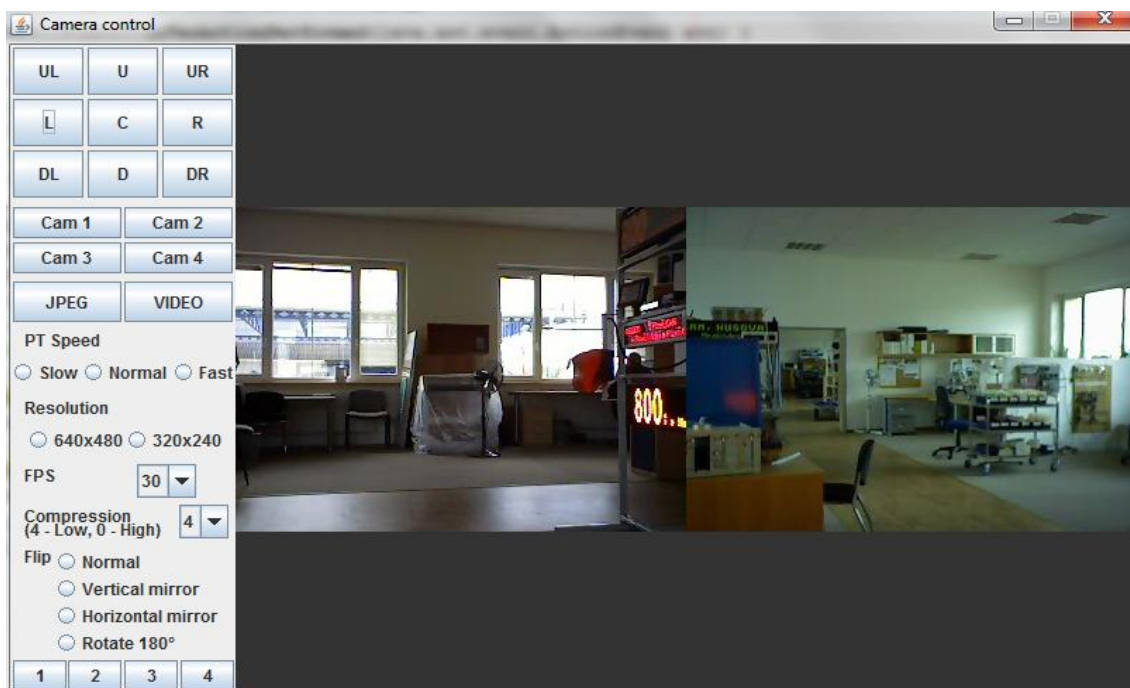
Obr. 6.3: Nejvyšší komprese JPEG snímku (zdroj: autor)

Poslední čtyři přepínače umožňují otočení obrazu. V režimu Normal je obraz zobrazen tak jak jej kamera ve skutečnosti ukazuje. Další dvě možnosti slouží pro zrcadlové otočení obrazu podle vertikální nebo horizontální osy. Poslední možnost je otočení obrazu o 180°, což ukazuje Obr. 6.4. Tato možnost je využitelná především při montáži kamery na strop místnosti.

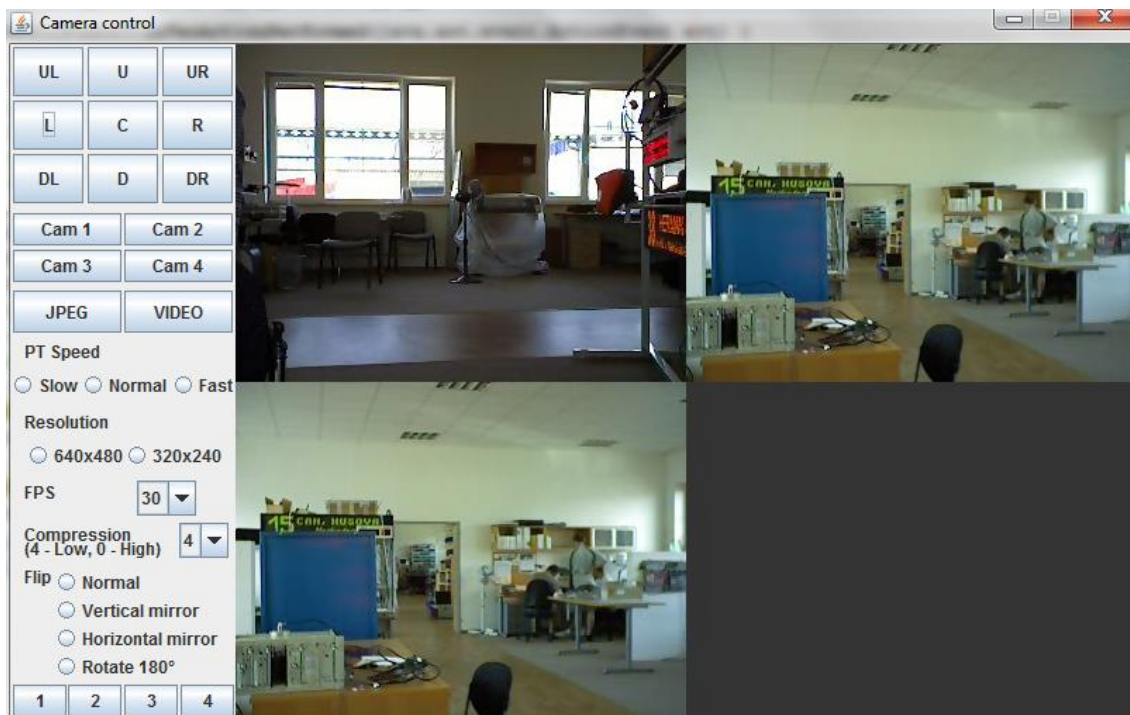


Obr. 6.4: Otočení obrazu o 180° (zdroj: autor)

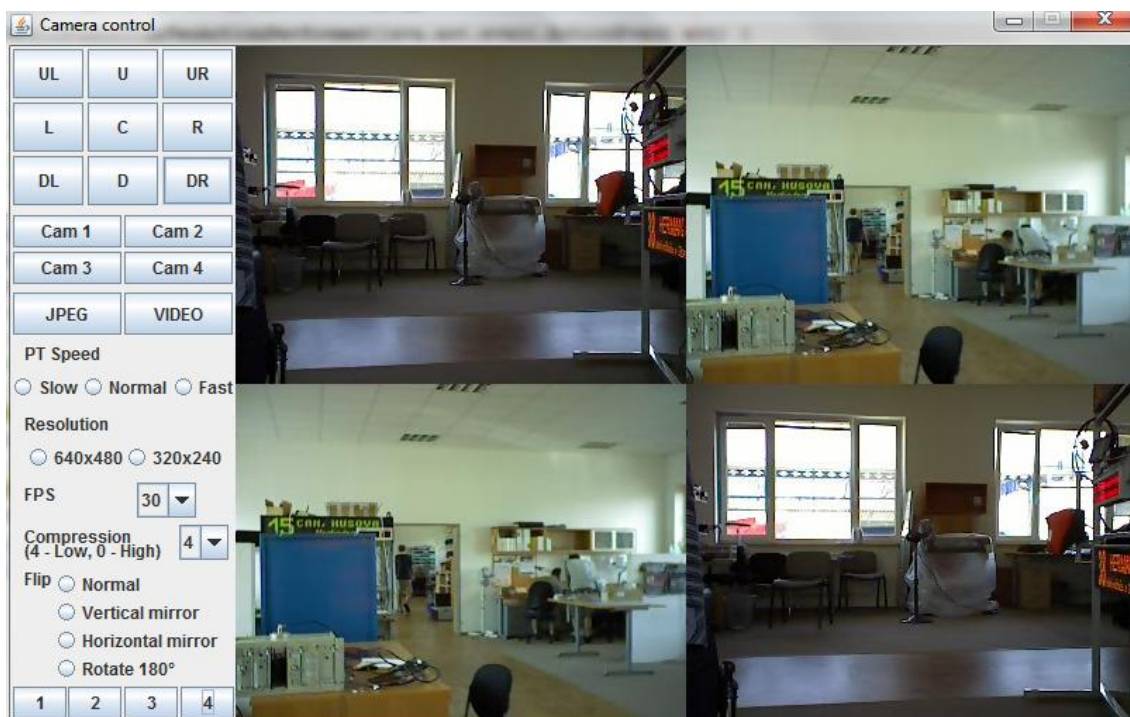
V levém dolním rohu programu se nachází poslední čtveřice tlačítek. Ty slouží pro volbu, z kolika kamer se bude zobrazovat obraz. Ukázky voleb zobrazení jsou na Obr. 6.5, Obr. 6.6 a Obr. 6.7.



Obr. 6.5: Zobrazení obrazů ze dvou kamer (zdroj: autor)



Obr. 6.6: Zobrazení obrazu ze tří kamer (zdroj: autor)



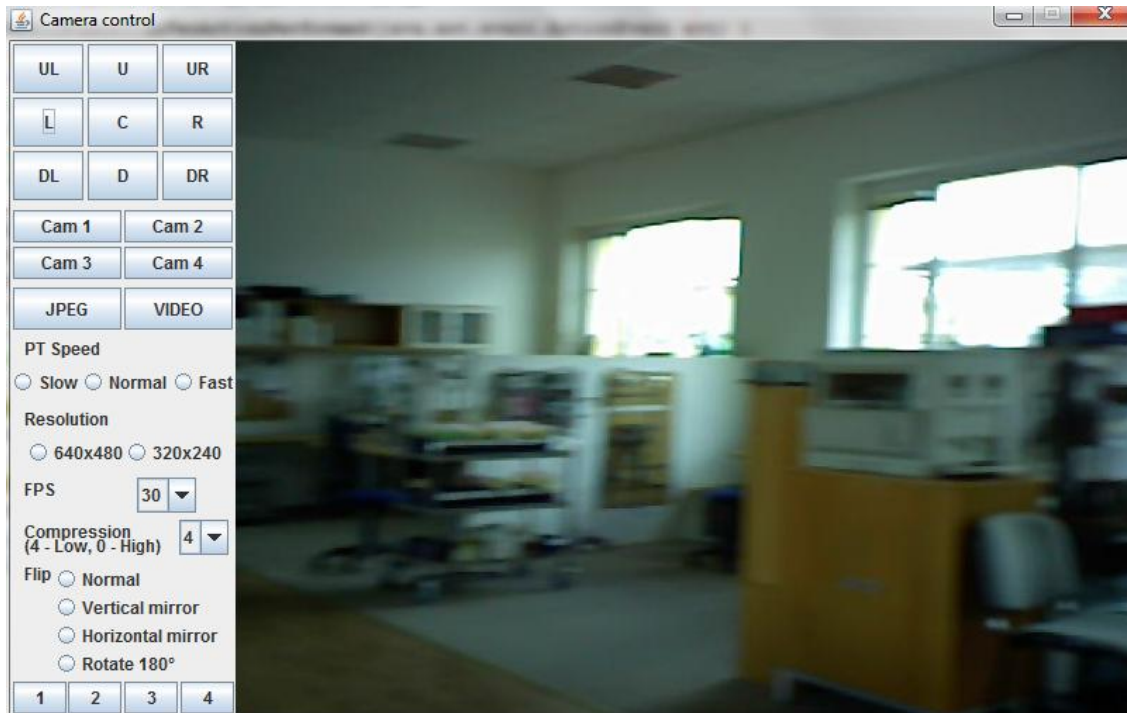
Obr. 6.7: Zobrazení obrazu ze čtyř kamer (zdroj: autor)

6.2 Poznámky k řešení

Výrobce Edimax neposkytuje kompletní seznam CGI příkazů, zejména možnost volby jasu, případně i kontrastu a sytosti by mohla v některých situacích chybět. Přitom v ovládání pomocí internetového prohlížeče tyto možnosti nastavení kamery jsou.

Dále není v programu zpracováno řešení se zachytáváním videa, jelikož výrobce dodává pouze příkaz pro multipart MJPG streamu, který by bylo nutné v programu dekodovat. Ve výsledku by toto řešení mohlo být při připojení dvou a více kamer náročnější než pouhé zachytávání JPEG obrázků. Ze stejného důvodu není použito i příkazů pro přenos zvuku.

Vlastní ovládací aplikace pomohla i při odhalení nedostatků použitého CMOS snímače kamery. Při čtení informace ze snímacího čipu dochází k tzv. „kácení hran“, které je způsobeno řádkovým čtením informace ze snímače. Tohoto jevu je možné si všimnout při snímání videa a otočení kamery vlevo či vpravo. Čtení informace je tak pomalé, že než se začne číst další řádek, kamera už je na jiném místě a dochází tedy ke zkosení obrazu, které je vidět na Obr. 6.8.



Obr. 6.8: Nedostatky snímáče kamery (zdroj: autor)

7 ZÁVĚR

Bakalářská práce se zabývá především síťovými kamerami. Je zde ukázána vnitřní stavba kamery s popisem jednotlivých bloků. IP kamery jsou rozebrány i z pohledu vnější stavby a od toho se odvíjejícího využití kamery. Práce pomáhá v orientaci při výběru vhodné síťové kamery a komunikačního rozhraní. Stranou nezůstávají ani možnosti připojení více kamer k obslužnému PC včetně jednoduchého schéma zapojení.

V teoretické části lze nalézt odpovědi na otázky okolo možností síťového videa, volby komunikačních protokolů a také funkčního vybavení kamer. Hlavním výstupem z kamery je video a jeho aspekty v podobě komprese, počtu snímků za sekundu, vyvážení barev, jasu a dalších.

Kapitola Známé způsoby řešení ukazuje možnosti výrobci dodávaných obslužných programů. Je zde program pro obsluhu kamerového systému i applet do internetového prohlížeče.

Návrh vlastního řešení obsahuje výčet dostupného hardwaru i softwaru, včetně ukázky CGI příkazů. Programování vlastního obslužného programu v programovacím jazyce Java. Rozebráno je zde vše od vývojového prostředí až po jednotlivé použité třídy, objekty, metody a proměnné ve vlastní aplikaci.

Výsledkem této práce je aplikace, která ukazuje funkce, ale i nedostatky dané IP kamery. Bylo dosaženo možnosti zobrazení obrazů ze čtyř kamer, včetně všech dostupných možností jejich ovládání. Některé funkce jsou však od výrobce nedostatečně popsány anebo nedostupné, proto nejsou v aplikaci zapracovány. Na základě toho je například nedostupná volba nastavení jasu a kontrastu. Snímání videa je řešeno pomocí periodického vyčítání JPEG obrázku.

Možné pokračování vývoje aplikace lze shledat v získávání MJPEG video streamu z kamery. Případné ukládání na disk počítače či sledování pohybu v obraze. Zajímavé pak bude sledovat vytížení procesoru počítače a jeho porovnání s dosaženým periodickým vyčítáním JPEG obrázků. Jako další možnosti zapracování se nabízejí: možnost výchozího počtu připojených kamer včetně nastavení jednotlivých IP adres či zasílání fotografií pomocí emailových zpráv.

LITERATURA

- [1] BÁBÍČEK, L. *Jak na počítač: Digitální video*. ČR: Computer Press, 2006. 90 s.
- [2] Edimax.com. *Bezpečnostní IP kamery* [online]. 2008 [cit. 2011-2-25]. Dostupné z WWW: <http://www.edimax.pl/images/Image/productimage/ipcamara/217x205/IC-9000_217X205.jpg>.
- [3] Edimax.com. *Bezpečnostní IP kamery* [online]. 2008 [cit. 2011-2-25]. Dostupné z WWW: <<http://www.edimax.pl/images/Image/imagecenter/ProductPictures/Internet%20IP%20Camera/IC-7000PT/IC-7000PT.jpg>>.
- [4] Edimax.com. *Bezpečnostní IP kamery* [online]. 2008 [cit. 2011-2-25]. Dostupné z WWW: <<http://www.edimax.pl/images/Image/productimage/ipcamara/IC-3010/IC-3010h.jpg>>.
- [5] CHURÝ, L. *Programujte.com* [online]. 2004. Dostupné z WWW: <<http://programujte.com>>. ISSN 1801-1586.
- [6] KEOGH, J. *JAVA bez předchozích znalostí*. ČR: Computer Press, 2005. 282 s. ISBN 80-251-0839-2.
- [7] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006 [cit. 2010-02-5]. Co je síťové video. Dostupné z WWW: <<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/co-je-sitove-video.php>>.
- [8] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006 [cit. 2010-02-2]. Power Over Ethernet. Dostupné z WWW: <<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/power-over-ethernet.php>>.
- [9] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006 [cit. 2010-02-10]. Metody přenosu dat. Dostupné z WWW: <<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/metody-prenosu-dat.php>>.
- [10] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006 [cit. 2010-03-3]. Standardy komprese videa. Dostupné z WWW: <<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/standardy-komprese-video.php>>.
- [11] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006 [cit. 2010-04-6]. Progresivní skenování. Dostupné z WWW: <<http://www.netcam.cz/encyklopedie-ip-zabezpeceni/progresivni-skenovani.php>>.
- [12] Netcam.cz. *Encyklopedie síťového videa* [online]. 2006. Dostupné z WWW: <<http://www.netcam.cz/reseni.php#encyklopedieH>>.

- [13] Netcam.cz. *Produkty* [online]. 2008 [cit. 2009-12-13]. Dostupné z WWW: <<http://www.netcam.cz/produkty/ip-kamery/ip-kamery-pro-indoor-210.php>>.
- [14] Netcam.cz. *Produkty* [online]. 2008 [cit. 2009-12-13]. Dostupné z WWW: <<http://www.netcam.cz/produkty/ip-kamery/ip-kamery-pro-autobusy-vlaky-209fd-r.php>>.
- [15] Oracle.com [online]. 1995. The Java Tutorials. Dostupné z WWW: <<http://download.oracle.com/javase/tutorial/>>.
- [16] Wikipedie. *Video Graphic Array* [online]. 2009 [cit. 2009-12-13]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Video_Graphics_Array>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CCD	Charged Coupled Device, snímací čip
CCTV	Closed Circuit Television, kamerový systém
CD	Compact Disc, optické záznamové médium
CGI	Common Gateway Interface, příkaz kamery
CMOS	Complementary Metal Oxide Semiconductor, snímací čip
DDNS	Dynamic Domain Name Server, síťový protokol
DDR	typ operační paměti
DHCP	Dynamic Host Configuration Protocol, aplikační protokol
DVD	Digital Versatile Disk, optické záznamové médium
FTP	File Transfer Protokol, síťový protokol
H.263	kódování videa
HD	High Definition, vysoké rozlišení
HTTP	Hyper Text Transfer Protocol, síťový protokol
HTTPS	Hyper Text Transfer Protocol over Secure Socket Layer), síťový protokol
IEEE	The Institute of Electrical and Electronics Engineers, asociace
IP	Internet Protokol, síťový protokol
ISO	International Organization for Standardization, Mezinárodní organizace pro normy
JPEG	Joint Photographic Experts Group international, formát obrazu
LCD	Liquid Crystal Display, displej z tekutých krystalů
MPEG	Motion Picture Experts Group, kódování videa
MPx	MegaPixel, milion obrazových bodů
Např.	například

PAL	Phase Alternating Line, standard kódování obrazu
PC	Personal Computer, osobní počítač
Pixel	Obrazový bod
PoE	Power over Ethernet, technologie pro napájení přes rozhraní Ethernet
PPPoE	Point - to - Point Protocol over Ethernet, síťový protokol
RAM	Random Access Memory, operační paměť
RGB	Red Green Blue, barevný prostor
RTP	Real - time Transfer Protocol, síťový protokol
SD	Secure Digital, typ paměťových karet
SMTP	Send Mail Transfer Protocol, síťový protokol
SPZ	Státní poznávací značka, identifikace automobilu
TCP	Transmission Control Protocol, internetový protokol
UDP	User Datagram Protocol, internetový protokol
UPnP	Universal Plug and Play, technologie připojení zařízení
USB	Universal Serial Bus, univerzální sériová sběrnice
UV	Ultraviolet, ultrafialové světlo
VGA	Video Graphics Array, standart pro zobrazení obrazu
Wi-Fi	Wireless Fidelity, bezdrátová síť

SEZNAM PŘÍLOH

A	Zdrojový kód programu	56
A.1	Hlavní třída Main.....	56
A.2	Grafické rozhraní Gui	58
A.3	Zobrazovací prvek ImageViever	64
A.4	Zobrazovací prvek ImageViewer2.....	67
B	Obsah přiloženého DVD	69

A ZDROJOVÝ KÓD PROGRAMU

A.1 Hlavní třída Main

```
package bp;

import javax.swing.*;
import java.io.*;
import java.net.*;
import java.net.HttpURLConnection;

public class Main {

    Gui gui;
    ImageViewer imageViewer;
    ImageViewer2 imageViewer2;
    ImageViewer3 imageViewer3;
    ImageViewer4 imageViewer4;

    String ipadr;
    String http;
    String resolution;
    //proměnné

    public static void main(String[] args) {

Authenticator.setDefault (new MyAuthenticator());
//authenticator
        Gui gui = new Gui();
        //graficke rozhrani
        gui.setTitle("Camera control");
        //titulek okna
        gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //chování po zavření
        gui.setSize(806, 508);
        //velikost okna
        gui.setVisible(true);
        //viditelnost
    }

    public void Action() {
        try {
            URL url = new URL(http);
            //CGI příkaz
            HttpURLConnection urlConnection =
                (HttpURLConnection) url.openConnection();
            //otevření URL, připojení
            InputStream inputStream = urlConnection.getInputStream();
            //vstupní stream, odpověď kamery
            inputStream.close();
            //uzavření streamu

        } catch (IOException e) {
            // vyjímka IO zpracování
        }
    }
}
```

```

public void NumberOfCameras() {
    try {
        URL url = new URL
            ("http://192.168.1.110/camera-cgi/admin/param.cgi?
            action" + "=update&Image.IO.Appearance.Resolution="
            +resolution); //CGI příkaz
        HttpURLConnection urlConnection =
            (HttpURLConnection) url.openConnection();
        //otevření URL, připojení
        InputStream inputStream = urlConnection.getInputStream();
        //vstupní stream, odpověď kamery
        inputStream.close();
        //uzavření streamu

    } catch (IOException e) {
        // vyjímka IO zpracování
    }

    try {
        URL url = new URL
            ("http://192.168.1.111/camera-cgi/admin/param.cgi?
            action" + "=update&Image.IO.Appearance.Resolution="
            +resolution);
        //CGI příkaz
        HttpURLConnection urlConnection =
            (HttpURLConnection) url.openConnection();
        //otevření URL, připojení
        InputStream inputStream = urlConnection.getInputStream();
        //vstupní stream, odpověď kamery
        inputStream.close();
        //uzavření streamu

    } catch (IOException e) {
        // vyjímka IO zpracování
    }
}

public static class MyAuthenticator extends Authenticator{
    // metoda volaná pro zabezpečené URL
    protected PasswordAuthentication getPasswordAuthentication() {
        // informace o přijetí
        String promptString = getRequestingPrompt();
        String hostname = getRequestingHost();
        InetAddress ipaddr = getRequestingSite();
        int port = getRequestingPort();
        String username = "admin"; // jméno
        String password = "1234"; // heslo
        return new PasswordAuthentication
            (username,password.toCharArray());
        // vložení informací zpět
    }
}
}

```

A.2 Grafické rozhraní Gui

Z důvodů úspory místa je vynechán automaticky generovaný kód a z každé funkce je uveden pouze jeden (první) zástupce. Kompletní zdrojový kód je na přiloženém DVD.

```
package bp;

import java.awt.event.*;

public class Gui extends javax.swing.JFrame implements
ItemListener{

    Main main;
    ImageViewer imageViewer;
    ImageViewer2 imageViewer2;
    ImageViewer3 imageViewer3;
    ImageViewer4 imageViewer4;

    String combo;
    String compres;
    String fps;
    String ipadr= "192.168.1.110";
    Integer cam = 1;
    Integer numberCam = 1;
    //proměnné

    public Gui() {
        initComponents();

        main = new Main();

        imageViewer = new ImageViewer();
        imageViewer.setSize(640, 480);
        imageViewer.setVisible(true);
        jPanel1.add(imageViewer);
        //nastavení ImageViewer

        imageViewer2 = new ImageViewer2();
        imageViewer2.setSize(320, 240);
        imageViewer2.setVisible(true);
        jPanel2.add(imageViewer2);
        //nastavení ImageViever2

        imageViewer3 = new ImageViewer3();
        imageViewer3.setSize(320, 240);
        imageViewer3.setVisible(true);
        jPanel3.add(imageViewer3);
        //nastavení ImageViever3

        imageViewer4 = new ImageViewer4();
        imageViewer4.setSize(320, 240);
        imageViewer4.setVisible(true);
        jPanel4.add(imageViewer4);
        //nastavení ImageViever4
```

```

        buttonGroupSpeed.add(jRadioButtonSpeedSlow);
        buttonGroupSpeed.add(jRadioButtonSpeedNormal);
        buttonGroupSpeed.add(jRadioButtonSpeedFast);
        buttonGroupResolution.add(jRadioButtonResHigh);
        buttonGroupResolution.add(jRadioButtonResLow);
        buttonGroupFlip.add(jRadioButtonFlipNormal);
        buttonGroupFlip.add(jRadioButtonFlipVertical);
        buttonGroupFlip.add(jRadioButtonFlipHorizontal);
        buttonGroupFlip.add(jRadioButtonFlip180);
        //přidání tlačítek do skupin
    }

    private void jButtonUpActionPerformed(java.awt.event.ActionEvent evt){
        main.http = "http://" +ipadr+ "/camera-cgi/com/ptz.cgi?move=up";
        //změna proměnné http ve třídě Main
        main.Action();
        //volání metody Action ve třídě Main
    }

    private void jButtonCameralActionPerformed(java.awt.event.ActionEvent
    evt) {
        cam = 1;
        //změna proměnné cam
        ipadr = "192.168.1.110";
        //změna proměnné ipadr
        imageViewer.ipadr = "192.168.1.110";
        //změna proměnné ipadr ve třídě ImageViewer
    }

    private void jRadioButtonSpeedSlowActionPerformed
    (java.awt.event.ActionEvent evt) {
        main.http = "http://" +ipadr+ "/form/enet?
           enet_source=cp.asp&enet_avs_pt_speed=3";
        //změna proměnné http ve třídě Main
        main.Action();
        //volání metody Action ve třídě Main
    }

    private void jRadioButtonResHighActionPerformed
    (java.awt.event.ActionEvent evt) {
        if (numberCam == 1) {
            imageViewer.setBounds(0, 0, 640, 480);
            //velikost a umístění ImageViewer
            main.http =
                "http://" +ipadr+ "/camera-cgi/admin/param.cgi?
                action=update&Image.IO.Appearance.Resolution=640x480";
            //změna proměnné http ve třídě Main
            main.Action();
            //volání metody Action ve třídě Main
        }
    }

    private void jComboBoxCompressionActionPerformed
    (java.awt.event.ActionEvent evt) {
        jComboBoxCompression.addItemListener(this);
        //přidání ItemListener
    }

```

```

        compres = combo;
        //zapsání proměné combo do compres
        main.http = "http://" + ipadr + "/camera-cgi/admin/param.cgi?
            action=update&Image.I0.Appearance.Compression=" + compres;
        //změna proměnné http ve třídě Main
        main.Action();
        //volání metody Action ve třídě Main
    }

private void jComboBoxFpsActionPerformed
(java.awt.event.ActionEvent evt) {
    jComboBoxFps.addItemListener(this);
    //přidání ItemListener
    if (combo == "1") {
        imageView1.time = 1000;
        imageView2.time = 1000;
        imageView3.time = 1000;
        imageView4.time = 1000;
    }
    if (combo == "3") {
        imageView1.time = 333;
        imageView2.time = 333;
        imageView3.time = 333;
        imageView4.time = 333;
    }
    if (combo == "5") {
        imageView1.time = 200;
        imageView2.time = 200;
        imageView3.time = 200;
        imageView4.time = 200;
    }
    if (combo == "10") {
        imageView1.time = 100;
        imageView2.time = 100;
        imageView3.time = 100;
        imageView4.time = 100;
    }
    if (combo == "15") {
        imageView1.time = 67;
        imageView2.time = 67;
        imageView3.time = 67;
        imageView4.time = 67;
    }
    if (combo == "30") {
        imageView1.time = 33;
        imageView2.time = 33;
        imageView3.time = 33;
        imageView4.time = 33;
    }
    //změna proměnné time
}

private void jRadioButtonFlipNormalActionPerformed
(java.awt.event.ActionEvent evt) {
    main.http = "http://" + ipadr + "/form/enet?
        enet_source=cp.asp&enet_avs_flip_mode=0";
    //změna proměnné http ve třídě Main
    main.Action();
    //volání metody Action ve třídě Main
}

```

```

private void jButtonJpegActionPerformed
(java.awt.event.ActionEvent evt) {
    if (cam == 1) {
        imageView.Jpeg();
        //volání metody Jpeg v ImageViewer
        imageView.video = false;
        //změna proměnné video ve třídě ImageViewer
    } else {
        imageView.video = true;
    }
    if (cam == 2) {
        imageView2.Jpeg();
        imageView2.video = false;
    } else {
        imageView2.video = true;
    }
    if (cam == 3) {
        imageView3.Jpeg();
        imageView3.video = false;
    } else {
        imageView3.video = true;
    }
    if (cam == 4) {
        imageView4.Jpeg();
        imageView4.video = false;
    } else {
        imageView4.video = true;
    }
}

```

```

private void jButtonVideoActionPerformed
(java.awt.event.ActionEvent evt) {
    imageView.video = true;
    imageView2.video = true;
    imageView3.video = true;
    imageView4.video = true;
    //změna proměnné video ve třídě ImageViewer
    imageView.Video();
    imageView2.Video();
    imageView3.Video();
    imageView4.Video();
    //volání metody Video v ImageViewer
}

```

```

private void jButtonNumberCamOneActionPerformed
(java.awt.event.ActionEvent evt) {
    numberCam = 1;
    imageView.numberCam = 1;
    imageView2.numberCam = 1;
    imageView3.numberCam = 1;
    imageView4.numberCam = 1;
    //změna proměnné numberCam
    jPanel2.setVisible(false);
    jPanel3.setVisible(false);
    jPanel4.setVisible(false);
    //viditelnost komponent jPanel
    jPanel1.setBounds(160, 0, 640, 480);
    jPanel1.setSize(640, 480);
    imageView.setSize(640, 480);
}

```

```

        //velikost a umístění oken
        main.resolution = "640x480";
        //změna proměnné resolution ve třídě Main
        main.NumberOfCameras();
        //volání metody NumberOfCameras ve třídě Main
    }

private void jButtonNumberCamTwoActionPerformed
(java.awt.event.ActionEvent evt) {
    numberCam = 2;
    imageView.numberCam = 2;
    imageView2.numberCam = 2;
    imageView3.numberCam = 2;
    imageView4.numberCam = 2;
    //změna proměnné numberCam
    jPanel2.setVisible(true);
    jPanel3.setVisible(false);
    jPanel4.setVisible(false);
    //viditelnost komponent jPanel
    jPanel1.setBounds(160, 120, 320, 240);
    jPanel2.setBounds(480, 120, 320, 240);
    imageView.setSize(320, 240);
    //velikost a umístění oken
    main.resolution = "320x240";
    //změna proměnné resolution ve třídě Main
    main.NumberOfCameras();
    //volání metody NumberOfCameras ve třídě Main
    imageView2.Video();
    //volání metody Video v ImageViewer2
}

private void jButtonNumberCamThreeActionPerformed
(java.awt.event.ActionEvent evt) {
    numberCam = 3;
    imageView.numberCam = 3;
    imageView2.numberCam = 3;
    imageView3.numberCam = 3;
    imageView4.numberCam = 3;
    //změna proměnné numberCam
    jPanel2.setVisible(true);
    jPanel3.setVisible(true);
    jPanel4.setVisible(false);
    //viditelnost komponent jPanel
    jPanel1.setBounds(160, 0, 320, 240);
    jPanel2.setBounds(480, 0, 320, 240);
    imageView.setSize(320, 240);
    //velikost a umístění oken
    main.resolution = "320x240";
    //změna proměnné resolution ve třídě Main
    main.NumberOfCameras();
    //volání metody NumberOfCameras ve třídě Main
    imageView2.Video();
    imageView3.Video();
    //volání metod Video v ImageViews
}

private void jButtonNumberCamFourActionPerformed
(java.awt.event.ActionEvent evt) {
    numberCam = 4;

```

```

        imageView.numberCam = 4;
        imageView2.numberCam = 4;
        imageView3.numberCam = 4;
        imageView4.numberCam = 4;
        //změna proměnné numberCam
        jPanel2.setVisible(true);
        jPanel3.setVisible(true);
        jPanel4.setVisible(true);
        //viditelnost komponent jPanel
        jPanel1.setBounds(160, 0, 320, 240);
        jPanel2.setBounds(480, 0, 320, 240);
        imageView.setSize(320, 240);
        //velikost a umístění oken
        main.resolution = "320x240";
        //změna proměnné resolution ve třídě Main
        main.NumberOfCameras();
        //volání metody NumberOfCameras ve třídě Main
        imageView2.Video();
        imageView3.Video();
        imageView4.Video();
        //volání metod Video v ImageViews
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Gui().setVisible(true);
            } });
    }

    public void itemStateChanged (ItemEvent change) {
        combo = change.getItem().toString();
        //zapsání hodnoty comboBoxu do proměnné combo
    }
}

```

A.3 Zobrazovací prvek ImageViewer

```
package bp;

import java.awt.Graphics;
import java.awt.Image;
import java.io.*;
import java.net.*;
import java.net.HttpURLConnection;
import javax.imageio.ImageIO;

public class ImageViewer extends javax.swing.JPanel implements
Runnable {

    Boolean video;
    Image image;
    Integer numberCam = 1;
    Integer time = 33;
    String ipadr = "192.168.1.110";
    Thread animator = null;
    //proměnné
    public void Video() {
        animator = new Thread(this); //nové vlákno
        animator.start();           //start vlákna
    }

    public void run() {
        while(animator != null) {    //start cyklu while
            if (video == true){      //podmínka video
                if (numberCam == 1){  //podmínka počtu kamer
                    try {
                        URL url = new URL("http://" + ipadr + "/jpg/image.jpg");
                        //CGI příkaz
                        HttpURLConnection urlConnection =
                            (HttpURLConnection) url.openConnection();
                        //otevření URL, připojení
                        image=ImageIO.read(url);
                        //načtení obrázku z URL
                        InputStream inputStream = urlConnection.getInputStream();
                        //vstupní stream, odpověď kamery
                        inputStream.close();
                        //uzavření streamu

                        Thread.sleep(time);
                        //uspání vlákna (ms)

                    } catch (InterruptedException e) {
                        //vyjímka přerušení
                    } catch (IOException e) {
                        //vyjímka IO zpracování
                    }
                    setImage(image); //vlození obrázku do setImage
                    repaint();       //překreslení
                }
            }
        }
    }
}
```

```

if (numberCam == 2 | numberCam == 3 | numberCam == 4){
//podmínka počtu kamer
try {
    URL url = new URL("http://192.168.1.110/jpg/image.jpg");
    //CGI příkaz
    HttpURLConnection urlConnection =
        (HttpURLConnection) url.openConnection();
    //otevření URL, připojení
    image=ImageIO.read(url);
    //načtení obrázku z URL
    InputStream inputStream = urlConnection.getInputStream();
    //vstupní stream, odpověď kamery
    inputStream.close();
    //uzavření streamu

    Thread.sleep(time);
    //uspání vlákna (ms)

} catch (InterruptedException e) {
    //vyjímka přerušení
} catch (IOException e) {
    //vyjímka IO zpracování
}

setImage(image); //vložení obrázku do setImage
repaint();      //překreslení

}
} else {
    try {
        Thread.sleep(1000);
        //uspání při nečinnosti
    } catch (InterruptedException ex) {
    }
}
}

public void Jpeg() {
    if (video == false){ //podmínka video
        if (numberCam == 1){ //podmínka počtu kamer
            try {
                URL url = new URL("http://" + ipadr + "/jpg/image.jpg");
                //CGI příkaz
                HttpURLConnection urlConnection =
                    (HttpURLConnection) url.openConnection();
                //otevření URL, připojení
                image=ImageIO.read(url);
                //načtení obrázku z URL
                InputStream inputStream = urlConnection.getInputStream();
                //vstupní stream, odpověď kamery
                inputStream.close();
                //uzavření streamu

            } catch (IOException e) {
                //vyjímka IO zpracování
            }
        }
    }
}

```

```

        setImage(image); //vložení obrázku do setImage
        repaint();      //překreslení
    }
    if (numberCam == 2 | numberCam == 3 | numberCam == 4){
//podmínka počtu kamer
        try {
            URL url = new URL("http://192.168.1.110/jpg/image.jpg");
            //CGI příkaz
            HttpURLConnection urlConnection =
                (HttpURLConnection) url.openConnection();
            //otevření URL, připojení
            image=ImageIO.read(url);
            //načtení obrázku z URL
            InputStream inputStream = urlConnection.getInputStream();
            //vstupní stream, odpověď kamery
            inputStream.close();
            //uzavření streamu

            } catch (IOException e) {
                //vyjímka IO zpracování
            }
            setImage(image); //vložení obrázku do setImage
            repaint();      //překreslení
        }
    }
}

public void setImage (Image image){
    this.image = image;
    //uložení obrázku v GUI a prenesení pro vykreslení
}

public void paint (Graphics g) { //grafické vykreslení obrázku
    super.paintComponent (g);
    g.drawImage(image,0 , 0, this);
}
}

```

A.4 Zobrazovací prvek ImageViewer2

Zobrazovací prvek ImageViewer2 je téměř stejný jako zbylé dva prvky ImageViewer3 a ImageViewer4. Nejsou tak z důvodů úspory místa zobrazeny. Kompletně na DVD.

```
package bp;

import java.awt.Graphics;
import java.awt.Image;
import java.io.*;
import java.net.*;
import java.net.HttpURLConnection;
import javax.imageio.ImageIO;

public class ImageViewer2 extends javax.swing.JPanel implements
Runnable {

    Boolean video;
    Image image2;
    Integer numberCam = 1;
    Integer time = 33;
    Thread animator2 = null;
    //proměnné

    public void Video() {
        animator2 = new Thread(this); //nové vlákno
        animator2.start();           //start vlákna
    }

    public void run(){
        while(animator2 != null) {    //start cyklu while
            if (video == true){        //podmínka video
                if (numberCam == 2 | numberCam == 3 | numberCam == 4){
                    //podmínka počtu kamer
                    try {
                        URL url = new URL("http://192.168.1.111/jpg/image.jpg");
                        //CGI příkaz
                        HttpURLConnection urlConnection =
                            (HttpURLConnection) url.openConnection();
                        //otevření URL, připojení
                        image2=ImageIO.read(url); //načtení obrázku z URL
                        InputStream inputStream = urlConnection.getInputStream();
                        //vstupní stream, odpověď kamery
                        inputStream.close(); //uzavření streamu

                        Thread.sleep(time); //uspání vlákna (ms)
                    } catch (InterruptedException e) {
                        //vyjímka přerušení
                    } catch (IOException e) {
                        //vyjímka IO zpracování
                    }
                    setImage(image2); //vložení obrázku do setImage
                    repaint();        //překreslení
                }
            }
        }
    }
}
```

```

    } else {
        try {
            Thread.sleep(1000); //uspání při nečinnosti
        } catch (InterruptedException ex) {
        }
    }
} else {
    try {
        Thread.sleep(1000); //uspání při nečinnosti
    } catch (InterruptedException ex) {
    }
}
}
}

public void Jpeg() {
    if (video == false) { //podmínka video
        if (numberCam == 2 | numberCam == 3 | numberCam == 4){
            //podmínka počtu kamer
            try {
                URL url = new URL("http://192.168.1.111/jpg/image.jpg");
                //CGI příkaz
                HttpURLConnection urlConnection =
                    (HttpURLConnection) url.openConnection();
                //otevření URL, připojení
                image2=ImageIO.read(url);
                //načtení obrázku z URL
                InputStream inputStream = urlConnection.getInputStream();
                //vstupní stream, odpověď kamery
                inputStream.close();
                //uzavření streamu

                } catch (IOException e) {
                    //vyjímka IO zpracování
                }
                setImage(image2); //vložení obrázku do setImage
                repaint();        //překreslení
            }
        }
    }

    public void setImage (Image img){
        this.image2 = img;
        //uložení obrázku v GUI a prenesení pro vykreslení
    }
    public void paint (Graphics g) { //graficke vykreslení obrázku
        super.paintComponent (g);
        g.drawImage(image2,0 , 0, this);
    }
}

```

B OBSAH PŘILOŽENÉHO DVD

Základní struktura DVD:

- **Kódy_programu:** textové soubory s kompletními zdrojovými kódy všech částí programu
- **Manuál:** manuál ke kameře, CGI příkazy, Quick Instalation Guide
- **Program_Camera_Control:** obsahuje program spustitelný souborem BP4.jar
- **Projekt_BP4:** projekt programu ve vývojové prostředí Netbeans 6.8
- vlastní práce v souboru PDF