

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra fyziky

Konverze grafických souborových formátů

Bakalářská práce

Vedoucí práce: Ing. Michal Šerý

Vypracoval: David Hromádko

Konverze grafických souborových formátů

V této práci jsem se zaměřil na základní popis bitmapových a vektorových formátů. Je zde popsána struktura grafických souborů jejich výhody a vhodné použití. Jsou zde vysvětleny základní pojmy počítačové grafiky. Popis video formátů, video kontejnerů a kodeků. Rozdíl mezi jednotlivými pojmy, jejich parametry a popsány některé konkrétní typy. Dále je zde popis jak postupovat při převodu animace do video souboru a popsány konkrétní příklady převodu.

Klíčová slova

Pixel, barevné systémy, komprese, bitmapový soubor, vektorový soubor, video

Conversion graphic formats

In this dissertation I have focused on basic characterization bitmapped and vector graphics. Here are described structure graphic files, their benefits and appropriate application. There are explained basic concepts of computer graphics. Description of video formats, video containers and codec. Differences between single definitions, their characteristics and description some concrete types. Further there is characterization how to proceed to conversion of an animation to the video file and the description of concrete example of conversion.

Keywords

Pixel, color models, compression, bitmapped graphic file, vector graphic file, video

Prohlašuji, že bakalářskou práci jsem vypracoval samostatně a uvedl všechnu použitou literaturu.

V Českých Budějovicích dne 22.11.2006

David Hromádko

1	Úvod a základní pojmy	6
1.1	Úvod.....	6
1.2	Pixel	6
1.2.1	Fyzické pixely	7
1.2.2	Logické pixely	7
1.2.3	Hloubka pixelu a displeje	7
1.3	Palety	8
1.4	Barvy.....	9
1.4.1	Vnímání barev.....	9
1.4.2	Aditivní a subtraktivní modely	10
1.4.3	Nejrozšířenější barevné systémy	11
1.4.4	Překrývání a průhlednost	12
1.5	Komprese	13
1.5.1	Fyzická a logická komprese.....	13
1.5.2	Symetrická a asymetrická komprese.....	13
1.5.3	Ztrátová a neztrátová komprese.....	14
2	Bitmapové soubory	15
2.1	Organizace bitmapových souborů	15
2.1.1	Hlavička	15
2.1.2	Bitmapová data	16
2.1.3	Data zapsaná do vzorkovacích řádků.....	16
2.1.4	Planární data	17
2.1.5	Organizace bitmapových dat	17
2.1.6	Paleta.....	18
3	Vektorové soubory.....	20
3.1	Organizace vektorového souboru	20
3.2	Metasoubory	21
4	Video a animované formáty.....	22
4.1	Animace	22
4.2	Video.....	22
4.2.1	Digitalizace obrazu	23
4.2.2	Datový tok.....	24
4.2.3	Komprese dat při práci s video a audio soubory.....	25
4.2.4	Parametry videosouboru	26
4.2.5	Kompresní systémy.....	28
5	Převody animací	33
6	Závěr	40
	Seznam literatury	41

1 Úvod a základní pojmy

1.1 Úvod

Problematika prezentace informací má své kořeny v dávné minulosti, v době klasických psacích a kreslicích nástrojů a v době klasických technologií tisku. Teprve s nástupem výpočetní techniky a díky jejímu obrovskému rozšíření došlo ke vzniku a rozvoji grafických formátů.

Obrovské množství informací, které lidstvo shromáždilo, se nachází převážně v textové a grafické formě a výpočetní technika se stává jediným nástrojem, který je schopen řešit problém zpracování a využití těchto informací. Tvorbu dokumentů si dnes nelze představit bez reprodukcí fotografií, kreseb, počítačově vytvořených obrázků, nejrůznějších grafických symbolů, značek apod.

V minulosti sloužily obrazové prvky převážně jako ilustrace, jako doplňkový prvek k textu dokumentu. Dnes se situace u mnoha typů dokumentů obrací. Obraz, fotografie nebo kresba se často stává hlavním nositelem informace, protože může být snadno přijímán jako ucelený a univerzální způsob komunikace se čtenářem. Kvalita a druh obrazových prvků v dokumentech jsou daleko více sledovány, a jsou na ně kladeny mnohem vyšší nároky než dříve. Někdy je nutné, aby obrazové prvky měly silnou a jasnou vypovídací schopnost. Jen tehdy je člověk schopen přijímat a rozlišovat informační obsah obrazových prvků.

Formát grafického souboru je formát, v kterém jsou grafická data, data popisující grafickou předlohu, uložena ve formě souboru. Tyto soubory byly zavedeny z toho důvodu, že postupně vyvstávala potřeba ukládat, organizovat a znovu obnovovat grafická data efektivně a logicky.

1.2 Pixel

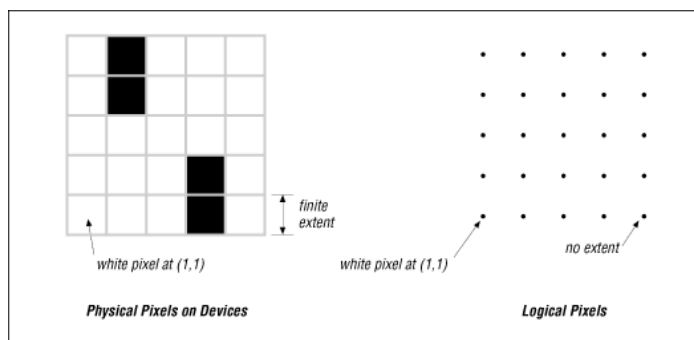
Pixel (picture element) je jeden ze základních pojmů počítačové grafiky. V počítačové grafice jsou pozice dat uloženy jako matematické souřadnice. Zobrazovací plocha výstupního zařízení je však fyzický objekt. Proto je nutné si uvědomit rozdíly mezi pixely (body) fyzickými a logickými.

1.2.1 Fyzické pixely

Fyzické pixely (obrázek 1-1) jsou body, které používá k zobrazování výstupní zařízení. Tyto prvky jsou přímo ovládány hardwarem výstupního zařízení. V praxi obrazové zařízení překrývá nebo skládá vedle sebe několik menších bodů tak, aby vytvořily jeden barevný bod.

1.2.2 Logické pixely

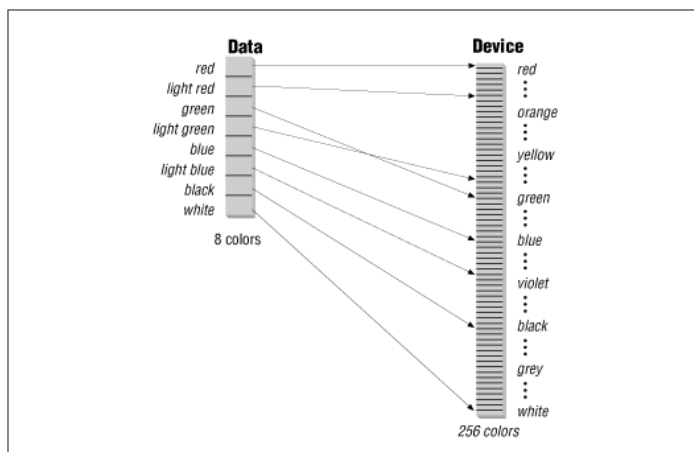
Na rozdíl od fyzických pixelů jsou logické pixely matematické body. Specifikují polohu, ale fyzicky nezabírají žádnou plochu. Mapování mezi logickými pixely v bitmapových datech musí odpovídat skutečné velikosti a uspořádání fyzických pixelů. Husté a jasně barevné bitmapy se mohou začít chvět při zobrazení na příliš velkém monitoru, protože pixely se musí rozprostřít po celé ploše zobrazovací plochy. Ukázka fyzických a logických pixelů je na obrázku 1-1.



Obrázek 1-1 Fyzické a logické pixely

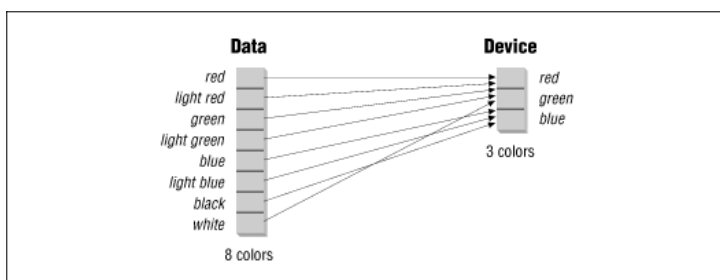
1.2.3 Hloubka pixelu a displeje

Počet bitů, které reprezentují daný pixel, rovněž určuje počet barev, kterých může pixel nabýt. Čím větší je počet bitů, tím větší je počet možných barev a také větší prostor v paměti, v které jsou data uložena. Například jeden bit zobrazuje dvě barvy, 24 bitů však asi 16 miliónů barev. Ve skutečnosti se počet barev na pixel uložených v souboru liší od počtu, který může zobrazit dané výstupní zařízení. Nebývá problémem, pokud je počet barev nižší v souboru než počet, který může zobrazit výstupní zařízení. Problémem je opačný případ, kdy je zdrojových barev více než výstupních.



Obrázek 1-2 Zobrazení dat s méně barvami na zařízení s více barvami.

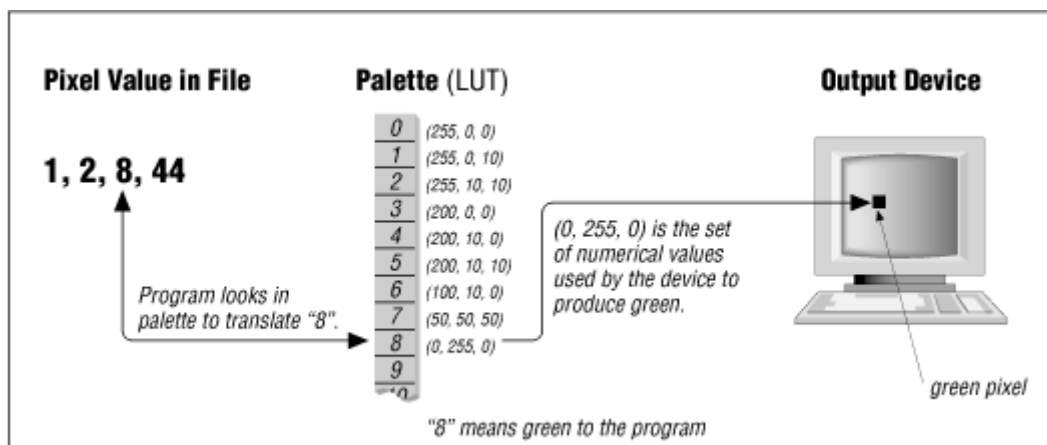
Například data o velikosti 4 bity na pixel zobrazujeme na zařízení 24 bitů na pixel. Výstupní zařízení je schopno zobrazit více barev, než je od něj žádáno od obrazové předlohy. Tento případ nám ukazuje obrázek 1-2. Opačný proces, kdy předloha má více barev, než je schopno zařízení zobrazit, má za následek ztrátu dat. Aplikace musí zredukovat počet barev tak, aby výstupní zařízení mohlo data zobrazit. Tento proces se nazývá kvantizace (obrázek 1-3).



Obrázek 1-3 Zobrazení dat s více barvami na zařízení s méně barvami.

1.3 Palety

Obrazová data, která mají hloubku více než jeden bit na pixel, se často ukládají pomocí palety. Paleta (mapa barev, indexová mapa, tabulka barev) je jednorozměrné pole hodnot barev. Grafický soubor s paletou potom obsahuje nejprve popis palety, a potom teprve vlastní data, kde jsou barvy určeny indexem do palety. V některých případech je lepší ukládat data jako indexy do palety, v jiných je lepší ukládat hodnotu barvy přímo. Pro obrazové předlohy, které obsahují nejvýše 256 barev (grafy, schémata, ale i některé obrazové předlohy), je vhodnější ukládání přes paletu, protože se tím výrazně zmenší velikost souboru. Pro pravé barvy se paleta nepoužívá. Použití palety pro určení barev je na obrázku 1-4.



Obrázek 1-4 Použití palety pro určení barev

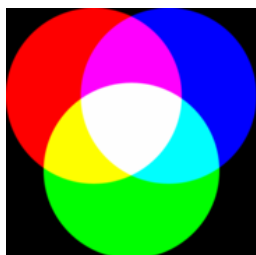
1.4 Barvy

1.4.1 Vnímání barev

Musí být splněny tři předpoklady: objekt, pozorovatel a světlo. Původním zdrojem světla je slunce, jež vnímáme jako nebarevné, denní, "bílé" světlo. Ale anglický fyzik Isaac Newton nás již dříve vyvedl z tohoto omylu svým známým pokusem s rozkladem světla pomocí skleněného hranolu na jednotlivé barevné složky (od červené po fialovou) a zpětným složením těchto složek do bílého světla. Stejným důkazem je i duha viditelná při dešti v letních měsících. Jak tedy vnímáme barvy? Tak například, když bílé světlo dopadne na červený objekt, tento objekt absorbuje (pohltní) všechny složky bílého světla kromě červené. Objekt pak odráží záření v červené části spektra. Jak již v minulém století zjistil pan Heimholz, lidské oko má buňky (receptory), které jsou citlivé na červenou, zelenou a modrou barvu, tyto barvy jsou též základními složkami bílého světla. Kombinací různých úrovní těchto tří základních barev je lidský mozek schopen vnímat tisíce jemnějších barevných odstínů. Je zde mnoho faktorů, které mohou ovlivnit způsob jakým vidíte barvu. Například světelné podmínky zásadním způsobem určují, jak barva vypadá, proto již v roce 1927 komise CIE stanovuje standardní osvětlení - illuminanty. Pro praxi DTP a přípravy tisku (norma ANSI) se používá illuminant D50, tedy světlo o barevné teplotě 5000 K. Dále u osoby, která pozoruje barvu delší dobu, dochází k únavě sítnice oka a barvy nejsou pak vnímány přesně. Také každý člověk cítí stejnou barvu poněkud jinak, a proto ji také popisuje odlišně. Toto je několik důvodů, proč jsou standardy v barevné komunikaci velmi důležité. [4]

1.4.2 Aditivní a subtraktivní modely

Aditivní míchání barev je takový způsob míchání barev, kdy se jednotlivé složky barev sčítají a vytváří světlo větší intenzity. Výsledná intenzita se rovná součtu intenzit jednotlivých složek. Pracuje se třemi základními barvami: červená, zelená a modrá. Barvy jsou vytvářeny přidáváním barev do barvy černé. Čím více se přidá dané barvy, tím více se výsledná barva blíží barvě bílé. Aditivní míchání barev odpovídá vzájemnému prolínání tří barevných kuželů světla ze tří reflektorů na bílém plátně. Každý reflektor má filtr odpovídající základní barvě. Část plátna, která je osvětlená rovnoměrně všemi třemi reflektory, je bílá. Když smícháme jen dvě barvy světla, například červenou a zelenou barvu, dostaneme barvu žlutou. Budeme-li clonou měnit poměr intenzity obou světél, dostaneme různé barevné odstíny mezi těmito barvami. Modrá a zelená barva ve stejném poměru dávají azurovou barvu, červená a modrá dávají barvu purpurovou. Smícháním dvou základních barev vznikne třetí, základní barva, která je barvou komplementární (doplňkovou), obrázek 1-5.



Obrázek 1-5 Aditivní model

Subtraktivní míchání barev je způsob míchání barev, kdy se s každou další přidanou barvou ubírá část původního světla. Pokud například skládáme na sebe barevné filtry nebo mícháme pigmentové barvy, mícháme je subtraktivní metodou. Světlo prochází jednotlivými barevnými vrstvami a je stále více pohlcováno. Výsledná barva se skládá z vlnových délek, které zbudou po odrazu nebo průchodu filtrem. Základní barvy jsou: žlutá, azurová, purpurová. Základní barvy subtraktivního míchání jsou komplementární (doplňkové) k základním barvám při jejich aditivním míchání. Smícháním modrozelené a žluté barvy vznikne barva zelená, žluté a purpurové barva červená a purpurové a modrozelené barva modrá. Smícháním všech tří základních barev dostaneme barvu černou, obrázek 1-6.



Obrázek 1-6 Subtraktivní model

1.4.3 Nejrozšířenější barevné systémy

RGB - Red, Green, Blue (červená, zelená, modrá)

Jde o aditivní systém míchání barev. V grafickém souboru používajícím RGB barevný systém je každý pixel reprezentován trojicí barev, jde o tři číselné hodnoty ve formě (R, G, B). Každá z nich přísluší dané základní barvě. Pro 24 bitové barvy trojice (0,0,0) představuje černou barvu a trojice (255,255,255) barvu bílou. Tento způsob se používá například při zobrazování na monitoru.

CMY - Cyan, Magenta, Yellow (azurová, purpurová, žlutá)

Jedná se o subtraktivní systém a používá se u tiskáren a fotografií. Barvy se zobrazují inkoustem nebo některou emulzí na bílém povrchu. Pokud je barva ozářena, každá ze tří základních barev absorbuje bílou složku. Azurová absorbuje červenou, purpurová zelenou a žlutá modrou. Pokud jsou všechny složky odečteny, vzniká barva černá. Protože v praxi je těžké získat kvalitní černou, přidává se k barvivům složka K, což je nezávislá černá barva. V systému CMY pro 24 bitovou hodnotu pixelu je (255,255,255) černá a (0,0,0) je barva bílá. Často se hodnoty udávají v procentech.

Před tiskem RGB obrázku je tedy nutné ho převést do barevného modulu CMYK. O tento proces se stará buď ovladač tiskárny, v profesionálním tisku pak tzv. RIP. Ovšem jako není dokonalý model RGB, ani CMYK nepokrývá všechny možné barvy, určitou část barevného spektra tedy zařízení pracující v systému CMYK není schopno zobrazit.

Jak je vidět, tisk doplňkových barev (syťá červená, syťá zelená a syťá modrá) je téměř nemožný. Je to způsobeno zejména právě tím, že monitor přímo vyzařuje světlo, kdežto výtisk světlo odráží. Naštěstí v případě fotografií jde o jev, který není ve výsledku patrný. Jinými slovy, vyfotíme-li předmět, který světlo odráží (což je kromě

světelných zdrojů vše), tak přestože jej na monitoru vidíme „jinak“, výsledná fotografie bude v tom nejlepším případě stejně barevná.

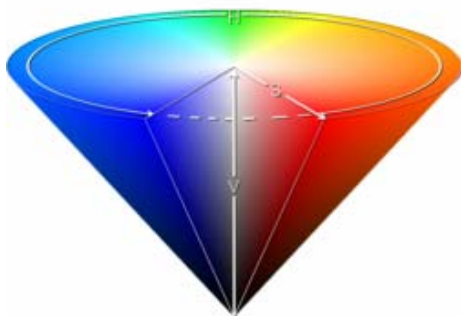
HSV - Hue, Saturation, Value (barva, sytost, jas)

Hue - barevný tón, převládající

Saturation - sytost barvy, příměs bílé barvy

Value - hodnota jasu, kolik světla emituje

HSV (obrázek 1-7) je systémem, kdy se nemíchají barvy jako takové, ale mění se v něm počet barevných úrovní. Z těchto se pak vytvářejí barvy nové. Například růžová barva je čistá červená barva s příměsí bílé barvy, čímž změním její sytost. Jas nám zase určuje, zda bude výsledná barva zářivá, či tmavá, tedy zářivost barvy.



Obrázek 1-7 Model HSV

Model YUV

Tento model se používá pro přenos televizních signálů v normě PAL. U tohoto modelu je jasová složka oddělena od barevných informací. Díky tomu lze jasový signál Y využít pro barevné i černobílé televizory. Zbývající signály U a V nesou informace o velikosti barevných složek obrazu. Obdobně funguje model YIQ nebo $Y C_B C_R$.

1.4.4 Překrývání a průhlednost

Některé formáty, například TGA, nám umožňují nastavit průhlednost pixelů. TGA formát využívá 15 bitů na pixel, ale tato hodnota pixelu je uložena v 16 bitech. Zbývající bit může být využit k zobrazení nebo nezobrazení daného pixelu. Takto se mohou překrývat dva obrazy, přičemž ve vrchním mohou být otvory, kterými je možné

vidět obraz spodní. Jiné formáty nám dokonce umožňují nastavit průhlednost. U 24 bitového TGA formátu nám zbyde 8 bitů na určení průhlednosti daného pixelu. Což umožňuje 256 úrovní nastavení. Těchto 8 bitů se také nazývá alfa kanálem.

1.5 Kompresce

Kompresce je proces, používaný pro redukci fyzické velikosti bloku informací. Kompresce nebo také komprimace dat se objevila spolu s rostoucím množstvím ukládání a přenášení informace. Snahou komprese dat je uložit informaci v kratší podobě s tím, že musíme být schopni informaci kdykoliv převést do původní podoby, nesmí tedy docházet ke ztrátě informace. Pro hodnocení kvality zhuštění informace se obvykle používá tzv. kompresní poměr, je to poměr délky zkomprimovaného souboru ku délce původního souboru. Kompresní poměr se udává buď v procentech z délky původního souboru nebo jako poměr, např. 2 : 1 znamená, že zkomprimovaný soubor má poloviční délku oproti původnímu souboru.

Téměř každý formát grafického souboru používá nějakou metodu komprese. Kompresi můžeme rozdělit na ztrátovou a neztrátovou, dále na symetrickou a asymetrickou, fyzickou a logickou kompresi.

1.5.1 Fyzická a logická komprese

Rozdíl mezi fyzickou a logickou kompresní metodou je založen na způsobu, jak jsou data uspořádána. Metoda fyzické komprese vytváří řetězce, které se alespoň trochu týkají informací v původních datech. Výsledný blok komprimovaných dat je menší než původní, protože fyzický kompresní algoritmus odstranil redundanci. Logická komprese je dovršena procesem logické substituce, tzn. nahrazením jednoho abecedního, číselného nebo binárního znaku jiným.

1.5.2 Symetrická a asymetrická komprese

Symetrická metoda komprese používá stejné algoritmy a provádí stejné množství práce při kompresi i při dekompresi. Asymetrická metoda většinou provede více práce v jednom směru. Například při vytváření záložních souborů předpokládáme, že většina z nich se číst nebude, proto se zde hodí metoda, která používá rychlou kompresi a pomalou dekompresi.

1.5.3 Ztrátová a neztrátová komprese

Při neztrátové kompresi se po zakódování a dekodování dat data vůbec nezmění, žádná data nebudou ztracena. Naopak u ztrátové komprese se některá data odstraní, aby se dosáhlo lepšího kompresního poměru. Vybírají se data, při jejichž odstranění se předloha změní co nejméně.

Několik nejrozšířenějších metod komprese:

- Kódování běžící délky (RLE) – rozšířený kompresní algoritmus používaný bitmapovými formáty jako jsou BMP, TIFF a PCX k redukci množství redundantních dat.
- Lempel-Ziv-Welch (LZW) – používáno formáty GIF a TIFF, tento algoritmus je rovněž součástí 42bitového standardu modemové komprese a také Postskriptu Level 2.
- CCITT kódování – druh datové komprese používané pro přenos faksimilií, která je standardizována CCITT (International Telegraph and Telephone Consultative Committee). Standard je založen na schématu klíčované komprese, vytvořené Davidem Hoffmanem a všeobecně známé jako Hoffmanovo kódování.
- JPEG – Joint Photographic Experts Group – Kompresní metoda používaná zejména pro data obrazových předloh a pro multimediální data. Základní JPEG implementace používá kódovací schéma založené na algoritmu diskrétní kosinové transformace (DCT).
- Fraktální – Matematická metoda kódování bitmapových obrazů reálného světa jako množiny matematických dat popisujících fraktální (podobné, opakující se vzory) vlastnosti předlohy.

2 Bitmapové soubory

Bitmapové soubory, nebo také rastrové, se používají k uchování bitmapových dat. Jsou vhodné k ukládání obrázků reálného světa, jako jsou fotografie. V podstatě obsahují přesnou mapu předlohy, pixel po pixelu.

2.1 Organizace bitmapových souborů

Skládají se ze dvou základních prvků, a to hlavičky a bitmapových dat. Informace, které nelze uložit do hlavičky se ukládají do paty.

2.1.1 Hlavička

Hlavička obsahuje informace o bitmapě umístěné dále v souboru. Hlavička je zpravidla tvořena z pevně určených polí. Nemusí se nacházet v každém souboru, ale používá se u běžně rozšířených formátů.

Seznam polí hlavičky:

Paleta

Bitmapový index

Identifikátor souboru

Verze souboru

Počet řádků v předloze

Počet pixelů na řádku

Počet bitů na pixel

Počet barevných ploch

Typ komprese

X začátek předlohy

Y začátek předlohy

Popis textu

Nevyužité místo

- Identifikátor souboru

Jde o jednoznačnou identifikační hodnotu. Úkolem je zajistit, aby aplikace mohla určit formát příslušného grafického souboru.

- Verze formátu

Po identifikaci souboru aplikace kontroluje verzi formátu a zjišťuje tak, jestli může pracovat s daty uloženými v souboru.

- Popis obrazové předlohy

Počet řádků v předloze obsahuje hodnotu, která udává, z kolika řádků se skládají skutečná bitmapová data. Počet pixelů na řádku označuje, kolik pixelů obsahuje jeden řádek. Počet bitů na pixel udává, jaké množství dat je nutné pro určení každého pixelů v barevné ploše. Nazývá se také hloubkou pixelu. Počet barevných ploch udává počet ploch, které jsou využívány. Většinou je plocha jen jedna.

- Typ komprese

Tato položka se objeví pokud formát podporuje kompresi ke zmenšení objemu dat.

- X a Y začátek předlohy

Tímto polem označujeme začátek předlohy na výstupním zařízení. Zpravidla jde o hodnotu (0,0), levý horní roh.

- Nevyužití místo

Na konci hlavičky se může vyskytovat rezervované pole, které neobsahuje žádná data. Odstraní se tím problémy při přidávání nových polí bez nutnosti měnit velikost hlavičky.

2.1.2 Bitmapová data

Bitmapová data zabírají většinu místa z celého souboru. Data jsou zpravidla umístěna hned za hlavičkou, pokud tomu tak není, musí být v hlavičce uložen offset, který určuje, kde jsou data uložena. Při zápisu dat do souboru se používají dvě metody: data zapsaná do vzorkovacích řádků a planární data.

2.1.3 Data zapsaná do vzorkovacích řádků

Při předpokladu, že obrazová předloha je poskládána z jednoho nebo více řádků, úseky bitmapových dat odpovídají jednotlivým řádkům. Tyto úseky zapisujeme za sebou od začátku až do konce souboru. Když známe velikost každého pixelu a počet pixelů na řádek, vypočítáme offset a tím určíme místo, kde začíná každý řádek v souboru. Například při pixelu o velikosti 8 bitů (1 byte) s 21 pixely na řádek (celkem 21 bytů), budou řádky začínat na pozicích 0, 21, 42, 63 v bitmapových datech. Často je však vyžadováno, aby byla délka řádku v bytech celé sudé číslo (long-word). Takže v našem případě by velikost byla 24 bytů na řádek. 3 byty na každém řádku jsou v tomto případě nevyužití, jakási vycpávka. Aplikace, která používá 21 bytů na řádek,

zabere stejné množství paměťového prostoru jako aplikace používající 24 bytů na řádek. V praxi je to však kompenzováno zvýšenou rychlostí zařízení, které je schopno zpracovávat 2 nebo 4 byty najednou.

2.1.4 Planární data

Tato metoda organizace je založena na rozdělení dat na několik ploch. Můžeme například rozdělit předlohu na tři bloky, z nichž každý reprezentuje jednu barevnou složku. Každá plocha je složena z řádků, jak již bylo popsáno v předešlé metodě. Pro znovuvytvoření obrázku je tedy nutné načíst všechny bloky.

Příklad šesti hodnot pixelů RGB formátu 24 bitová předloha, dva řádky a tři sloupce.

Data zapsaná do vzorkovacích řádků

(00,01,02) (03,04,05) (06,07,08)
(09,10,11) (12,13,14) (15,16,17)

Planární data

(00) (03) (06) (R) (09) (12) (15)
(01) (04) (07) (G) (10) (13) (16)
(02) (05) (08) (B) (11) (14) (17)

Byla zapsána stejná data, ale pouze v jiném uspořádání. V prvním případě se předloha skládá ze šesti 24 bitových pixelů, které jsou uloženy jako šest tříbytových pixelových hodnot umístěných v jedné ploše. V druhé (planární) metodě je stejná předloha uložena v 18 jednobytových pixelových hodnotách umístěných do tří ploch, z nichž každá přísluší dané barvě. Obě metody zabírají stejné množství paměťového místa. Většina bitmapových souborů je uložena neplanární metodou.

2.1.5 Organizace bitmapových dat

Máme-li předlohu, která je složena z řádků a ty z pixelů, můžeme je v souboru uložit třemi způsoby.

A. Souvislá data

Nejjednodušší způsob je uložit všechna data souvisle za sebou. Data jsou čtena stejně, jako byla uložena, tedy řádek po řádku. Data zapsaná tímto způsobem můžeme číst velmi rychle a po velkých částech.

B. Pásky

V této metodě jsou data uložena v pásech, které se také skládají z řádků souvisle uložených. Délka pásů je stejná jako délka předlohy. Obrázek je tvořen více než jedním pásem, a ty mohou být v souboru umístěny daleko od sebe. Pásky pracují jednodušeji s daty předlohy na zařízeních s omezenou velikostí paměti. Také se používají pokud jsou data v komprimovaném nebo kódovaném formátu. V tomto případě jsou data nejdříve načítána do bufferu, dekódují se a ukládají do jiného bufferu. Aplikace tak pracuje pouze s jednotlivými pásky. Formáty pracující s pásky obvykle obsahují určité informace v hlavičce souboru, jako je počet pásů, velikost každého z nich a jejich umístění.

C. Dlaždice

Třetí metoda je rozdělení předlohy na dlaždice. Šířka dlaždic může být libovolná. V praxi se však rozdělují tak, aby bylo data snadno ukládat do bufferu nebo je dekódovat. Zpravidla mají dlaždice stejnou velikost, musí být také zaručeno, že je rozdělena celá předloha a že se dlaždice nepřekrývají. Výhodou rozdělení do dlaždic je, že nám umožňuje na každou z nich použít jinou kompresi a je tak dosaženo optimálnějšího zpracování. Někdy také tato metoda umožňuje rychlejší dekódování a dekompresi. V hlavičce souboru musí být informace množství, velikosti a hodnotách offsetů. To nám také umožňuje zobrazit aplikací část předlohy, aniž by se načítala data celé předlohy.

2.1.6 Pata

Pata má stejně jako hlavička konstantní velikost. Nachází se na konci souboru a slouží jako doplněk hlavičky, pokud jsou například přidána nová data a nelze je již uložit do hlavičky.

Dále se v bitmapových souborech mohou vyskytovat ještě další datové struktury.

Paleta již popsána v odstavci 1.3

Tabulka offsetů příloh

Pokud se v souboru vyskytuje více než jedna příloha, umožňuje rozeznat začátky příloh v souboru.

Tabulka vzorkovacích řádků (pásů, dlaždic)

Používá se pro určení začátku každého vzorkovacího řádku v datech. To je výhodné tehdy, pokud jsou data komprimována a musí být zpřístupňována náhodně. Podobně je řešena tabulka vzorkovacích pásů nebo dlaždic.

Některé soubory pracují s odlišnými datovými strukturami. Příkladem může být formát TIFF. Obsahuje základní hlavičku, ale její data ukládá do sérií štítků zvaných adresáře souboru předlohy. Ty nemají stálou velikost ani stálé místo. Jsou spojovány do sérií offsetových hodnot souboru. Umožňuje konstruovat hlavičkovou strukturu, která může obsahovat jakékoliv informace, což podporuje univerzálnost souboru. Jiné formáty zase mohou informace o datech a paletě ukládat do oddělených souborů, například PAL soubory.

Shrnutí:

Bitmapové soubory mohou být snadno vytvářeny z existujících pixelových dat uložených v poli někde v paměti.

Pixelové hodnoty mohou být modifikovány individuálně nebo po větších množstvích.

Bitmapové soubory mohou být přenášeny velmi dobře na bodová výstupní zařízení, jako jsou obrazovky a tiskárny.

Mohou být velmi rozsáhlé a to zejména tehdy, pokud je v předloze velké množství barev. Datová komprese může velikost těchto dat zmenšit, data však musí být před svým použitím expandována a to může podstatně snížit rychlost čtecího a zobrazovacího procesu. Čím jsou bitmapové předlohy složitější, tím menší bude účinnost kompresního procesu.

3 Vektorové soubory

Vektorové soubory obsahují matematické popisy prvků předlohy, které jsou používány zobrazovací aplikací k vytvoření výsledného obrazu. Vektorové soubory jsou tedy tvořeny matematickým popisem objektů. Vektory jsou čárové segmenty definované počátečním bodem, směrem a délkou. Mohou být také ale mnohem složitější, například křivky. Přímky a křivky mohou být použity pro definici geometrických tvarů - kružnic, trojúhelníků a polygonů, které mohou následně definovat ještě složitější tvary - roviny, krychle a mnohostěny. Formáty vektorových souborů jsou používány od té doby, co počítače začaly zobrazovat čáry na výstupních zařízeních, vznikly tedy dříve než bitmapové formáty. Vektorové formáty, na rozdíl od bitmapových, umí ukládat také textové řetězce.

3.1 Organizace vektorového souboru

I když se vektorové soubory od sebe liší, hlavně proto, že každý byl vytvořen k jinému účelu, většina z nich obsahuje stejnou základní strukturu. Hlavičku a data obrazové předlohy.

Hlavička

Hlavička obsahuje všeobecné informace o souboru, které jsou přečteny jako první. Mohou tam být např. informace o barvách, formát souboru, číslo verze, výšku a šířku předlohy, pozici předlohy na výstupním zařízení atd. Typ informací uložených v hlavičce závisí na tom, jaká data jsou v souboru uložena. Velikost hlavičky se tudíž může měnit.

Data

Vektorové soubory jsou zpravidla tvořeny z elementárních vektorových dat, která obsahují informace o jednotlivých objektech, z kterých je předloha složena. Vektorová data v ASCII formátu uvedu na příkladu.

```
LINE,100,40,125,180, RED;CIRCLE,50,100,120,BLACK
```

Vektorová data tohoto příkladu obsahují čáru a kružnici. Prvky jsou odděleny středníky a obsahují jméno, jejich polohu a informaci o barvě. Po vlastních datech většinou následuje značka Konec souboru. Někdy bývá v souboru připojená pata souboru, kde mohou být uloženy informace obsahující datum a čas vytvoření souboru,

název programu, ve kterém byl soubor vytvořen, a počet objektů uložených v datech předlohy.

Vektorové soubory obsahují palety, před vlastním zobrazením se musí aplikace nejdříve podívat, jakým způsobem jsou definovány barvy. Někdy definujeme obrysy, které mohou být kresleny různou tloušťkou a různým stylem. Můžeme též ohraničené prvky vyplňovat barvou. Výplně mohou obsahovat šrafování nebo stínování. Vektorový prvek může být vyplněn více než jednou barvou. Nejjednodušší je vyplňování gradientní, což je hladký přechod mezi dvěma barvami.

Vektorové formáty zpravidla nedovolují kompresi dat. Jejich velikost je určena, když nepočítáme velikost palety a atributových informací, pouze počtem objektů, které obsahuje.

Vektorový formát je vhodné použít při ukládání předloh, které jsou založeny na prvcích složených čar. Jde hlavně o čáry, mnohoúhelníky a prvky, které lze rozložit na jednoduché geometrické objekty - text. Vektorová data mohou být snadno zvětšována a zmenšována. Snadno se s nimi manipuluje tak, aby co nejlépe vyhovovala požadavkům výstupního zařízení. Většinou je velmi jednoduché převést vektorová data do bitmapového formátu. Pro ukládání složitých předloh, jako jsou třeba fotografie, se vektorový formát nehodí.

3.2 Metasoubory

Metasoubory mohou obsahovat vektorová i bitmapová data. V souborech mohou být uloženy jen bitmapové nebo jen vektorové informace, ale je vhodnější, pokud obsahují oba typy dat. Metasoubory se jeden od druhého liší velkým způsobem, proto není možné je popsat všeobecným popisem. Metasoubory obsahující vektorová i bitmapová data, mohou být někdy menší než bitmapové verze stejné předlohy. Protože mohou obsahovat ASCII data dají se často velmi dobře komprimovat.

4 Video a animované formáty

Od počátku 90. let minulého století se začalo používat označení multimediální aplikace nebo multimediální software, které využívaly kombinace textových, obrazových, zvukových či animovaných nebo filmových dat. Multimediální data a informace musí být uloženy v diskových souborech, které používají formáty podobné formátům souborů s obrázky. Multimediální formáty však jsou mnohem složitější než většina jiných formátů, protože v nich musí být uloženo mnohem širší spektrum údajů. Data v těchto souborech obsahují text, obrázky, zvukový záznam, videozáznam, počítačové animace a další formy binárních dat, například číslíkové rozhraní pro hudební nástroje, řídicí informace a grafické typy písma.

4.1 Animace

Animace je způsob vytváření zdánlivě se pohybujících věcí. Slovo pochází z cizího slova znamenajícího oživení. Princip animace je zaznamenání sekvence snímků (frame), které jsou každý o sobě statický, ale liší se od sebe jen drobně. Po rychlém zobrazení těchto snímků vzniká dojem pohybu. Snímky se musí přehrávat takovou rychlostí, kterou už oko nepostřehne.

Formáty souborů animací jsou schopné uchovávat pouze nehybné obrázky, ne skutečné informace o videozáznamu. Schémata komprese obrázků, které se používá u souborů pro animace, jsou mnohem jednodušší než většina kompresí videozáznamů. Většina souborů pro animace používá kompresního schématu delta, který je formou proudového kódování, které ukládá a komprimuje pouze informaci o rozdílu mezi dvěma obrázky. To je lepší než komprese celého snímku.

4.2 Video

Počítače pracují s dvojkovou (binární, 1 a 0) informací, obraz a zvuk určený ke zpracování (stříhu) obrazu a zvuku na počítači je nutno do této formy převést. Digitální videokamery mají jedničky a nuly k dispozici přímo na tzv. DV výstupu. U analogových videokamer nebo na analogovém výstupu digitální videokamery (nebo u ostatních analogových výstupů) se musíme postarat o převod na tvar vhodný pro počítačové zpracování. To znamená na tvar čísel v dvojkové soustavě. Rozdíl mezi analogovým (např. klasická gramofonová deska) a digitálním (CD,DVD...) záznamem

informace spočívá v tom, že u digitálního záznamu se signál nezaznamenává na nosič přímo, ale nejdříve se převede na číselné hodnoty a teprve ty se uchovávají. Prakticky nic v přírodě není interpretováno digitálně a všechny tyto vstupní údaje je nutno do počítače dostat v dvojkové formě, tedy je digitalizovat. U obrazu se zaznamenávají informace o optických vlastnostech jeho jednotlivých bodů pixelů, tedy o jasu a barvě. Čím větší množství obrazových bodů jsme takto popsali a uložili na nosič, tím více se záznam blíží originálu. Objem záznamu (velikost v MB nebo jiných jednotkách) je tedy většinou významným ukazatelem jeho kvality.

Princip digitálního záznamu řeší prakticky všechny problémy svých analogových předchůdců. Především vyniká trvanlivostí a stabilitou informací. Výhoda digitálního záznamu se projeví i při kopírování. Nepřevádějí se analogové veličiny, které ze samé své podstaty nabírají další šum a zkreslení, ale čísla. A na nich není co změnit. Obraz převedený na čísla má i další možnosti. S čísly lze dále matematicky pracovat, což při dostatečně výkonném počítači znamená fantastické úpravy obrazu. Využití je obrovské a prakticky neomezené.

Předností analogového záznamu je především jeho jednoduchost a větší dynamické přenosové pásmo (analogové zvukové záznamové systémy jsou schopny nahrát skladby s velkou dynamikou věrněji a lépe než systémy digitální).

4.2.1 Digitalizace obrazu

Základem pro digitalizaci jakéhokoliv obrazu je převod světla na elektrické veličiny. K tomu slouží čip CCD. Zkratka CCD znamená Charged Coupled Device, prvek s vázaným nábojem. Zjednodušeně řečeno jde o polovodičový prvek, který při dopadu světelného záření získá elektrický náboj. Čím je větší intenzita dopadajícího světla, tím větší je náboj. Tyto miniaturní polovodičové prvky jsou ve velkém množství (až v milionech prvků) pravidelně uspořádány do plošné struktury, která zajišťuje akumulaci a přenos nábojů z čipu k dalšímu zpracování. Proti všem dříve používaným fotoelektrickým prvkům je podstatně menší a především významně citlivější. Pokud je umístěn za optickou soustavu (objektiv), vznikne zařízení pro převod obrazu na elektrické veličiny. V současnosti se za přijatelné ceny vyrábějí čipy CCD o ploše několika čtverečních milimetrů, na nichž je soustředěno až několik milionů prvků. Každý tento prvek (při troše zjednodušení) představuje obrazový bod neboli pixel. Množství pixelů zásadním způsobem určuje kvalitu získávaného obrazového záznamu. Čip CCD ovšem dává analogový záznam obrazu, tedy převod jeho jednotlivých bodů na

různé hodnoty napětí, což je pro počítač nesrozumitelné. Proto je za obrazovým čipem zařazen analogově-digitální převodník. Jeho úkolem je převod analogových elektrických veličin na numerické hodnoty. Samozřejmě, že se analogově-digitální převodník nepoužívá jen pro vstup obrazu, ale i zvuku a dalších analogových veličin. Způsob, jakým se to děje, nazýváme vzorkování, neboli sampling. Úroveň analogového signálu je měřena v pravidelných intervalech a převáděna do číselných veličin. Čím větší počet bodů bychom měřili, tím přesnější by byl záznam, a tedy i případná zpětná rekonstrukce. Počet těchto měření za sekundu je vzorkovací frekvence, čím je vyšší, tím věrnější zvuk pak z počítače zase dostaneme. Vzorkovací (sampler) frekvence se udává v počtech za sekundu, tedy v hertzech, v našem případě v kilohertzech (kHz). Výsledkem je dlouhá řada jedniček a nul vhodná pro ukládání a zpracování v počítači. U počítačového zpracování videa je to podobné, s tím rozdílem, že obraz z čipu CCD nebo jiného zdroje analogového či digitálního záznamu obrazu představuje mnohonásobně větší objem informací než u zvuku.

Obvody s obrácenou funkcí, tedy digitálně-analogové převodníky, najdeme obvykle na výstupech z počítače. Slouží pro zpětný převod již zpracovaného signálu do podoby vhodné pro analogová zařízení. Například zvuk je v počítači zaznamenán v podobě řady čísel, reproduktory či sluchátka však ke své práci potřebují spojitý napěťový signál. V digitálně-analogovém převodníku (zvukové kartě) se proto čísla zase mění zpět na jim odpovídající hodnoty napětí.

Stejně to je při počítačovém zpracování videa. Analogový záznam (v případě použití analogové videokamery) nejdříve převedeme na digitální a po úpravách (střih atd.) opět pomocí digitálně-analogového převodníku převedeme zpět na analogový videosignál, například pro sledování v TV.

V praxi jsou funkce analogově-digitálního i digitálně-analogového převodníku obsaženy na tzv. kartách, které se do počítače přidávají, když na něm chcete zpracovávat zvuk, obraz, video, televizní či rozhlasový signál atd. Všechny vstupní a výstupní konektory jsou pak obvykle soustředěny právě na příslušné kartě nebo na externích zařízeních s touto kartou spojených.

4.2.2 Datový tok

Při práci s videem nebo při jeho komprimování se jistě setkáme s pojmem datový tok - bitrate. Bitrate (datový tok) nám udává počet bitů za vteřinu, které přehrávač při přehrávání videa zpracovává. Bitrate souboru je tedy definovaný počet bitů za sekundu

obrazu (a samozřejmě i zvuku), který použije kodér pro kódování. Obecně a zjednodušeně platí, že čím více bitů za vteřinu je na kompresi použito, tím je výsledné video v lepší kvalitě.

Bitrate se rozděluje na dva typy: VBR nebo CBR.

CBR - constant bitrate, neboli stálý datový tok - datový tok je stejný (konstantní) po celou dobu přehrávání. Je jednoduchý na kompresi, avšak data proudí stále stejnou rychlostí, a to i na místech, kde to není bezpodmínečně potřeba (klidný obraz, žádný pohyb), a zabírají tak zbytečně místo na disku. Kodek tedy udržuje stále stejný bitrate bez ohledu na to, kolik je ho ve skutečnosti potřeba. Tento typ se běžně používá u starších kompresních algoritmů (např. MPEG Audio Layer 3) a má výhodu v tom, že se dá dobře sesynchronizovat díky konstantní velikosti za sekundu. Jestliže je tedy audio nahrávka vytvořena s konstantním bitrate, znamená to, že půlka skladby je přesně uprostřed souboru.

VBR - variable bitrate, neboli proměnný datový tok - zde se kompresní poměr mění podle složitosti scény, při rychlém pohybu obrazu je komprese nejmenší (a tedy datový tok největší). Naproti tomu na málo složitě pasáže se použije dat méně. Výhodou je, že takto lze dosáhnout při stejném průměrném datovém toku výrazně vyšší kvality výstupu než při konstantním datovém toku, nevýhoda naopak to, že se dá špatně odhadnout velikost výsledného souboru. Zvolený bitrate reprezentuje průměr a kodek málokdy odhadne správnou velikost. Tento typ kódování používá MPEG-4.

Obecně platí, že při VBR lze na jeden disk nahrát více dat (delší video) než u CBR, aniž by se to nějak viditelně projevilo na kvalitě obrazu.

4.2.3 Kompresie dat při práci s video a audio soubory

Jedním z největších problémů digitální práce s obrazem je nutnost zpracovávat a uchovávat neobvykle velká množství dat. Tato potíž se objevuje už u digitální fotografie, avšak mnohem markantnější je při práci s pohyblivým obrazem, tedy u digitálního videa. Není-li k dispozici dostatečná kapacita pro zpracování stávajícího objemu dat, je nutné tento objem snížit. To se děje metodou zvanou komprese. Základní princip je prostý. V každém souboru je poměrně velký podíl informace vyjádřen způsobem, který je značně neúsporný. Počítačové soubory jsou už ze své binární podstaty značně velké, například dlouhá rovná černá čára bude zaznamenána ve formě, kterou lze při troše zjednodušení přeložit asi takto: černá tečka, černá tečka, černá tečka, černá tečka a tak pořád dál, třeba i několik stránek. Přitom naprosto totéž lze vyjádřit

mnohem kratším zápisem: 200 x černá tečka. To samozřejmě platí v ještě větší míře pro velké jednobarevné plochy. Uvedený příklad s přímkou je metoda bezztrátové komprese. To znamená, že se při změně zápisu nic neztratilo a po dekomprimaci získáme původní informaci ve zcela nezměněné podobě. Někdy ovšem bezztrátová komprese nestačí a je nutné objem dat snižovat dál, i za cenu toho, že po opětovné dekompresi už nebudou data kompletní (ztrátová komprese). Poměr původního a zkomprimovaného objemu dat se nazývá kompresní poměr. Čím je větší, tím větší také bývají ztráty na kvalitě. U digitálního videa to vede ke ztrátě kvality výsledného obrazu (obraz obsahuje například méně detailů, které však člověk vnímá jen okrajově), což lze za určitých okolností akceptovat. Kompresi lze provádět jak hardwarovými prostředky (speciálními čipy na kartách pro zpracování obrazu), tak softwarovými metodami využívajícími výkon vlastního centrálního procesoru počítače (CPU). Hardwarové prostředky pochopitelně systém zatěžují méně, jsou tedy rychlejší, ale většinou dražší.

Pro centrální procesor počítače je mnohem jednodušší zkomprimovaný soubor zobrazit než naopak provádět komprimaci. Metody komprese jsou samozřejmě podstatně složitější než náš příklad s opakováním bodů přímky. U použití náročnějších metod komprese pak systémy založené na kompresi centrálním procesorem počítače nejsou schopné kompresi provádět přímo v reálném čase (tedy tak, jak data přicházejí). Tehdy se komprese provádí metodou zvanou off-line, což znamená, že provedení komprese trvá déle než samotná doba komprimovaného záznamu.

Pro kompresi počítačových dat existuje řada matematických metod, a tedy i příslušných hardwarových čipů nebo programů (těm se říká kodek). Specializovaný čip (kodek) je schopen provádět co nejrychleji požadovaný úkol (tedy kompresi a dekompresi obrazových dat), protože na rozdíl od centrálního procesoru počítače byl pro tuto činnost navržen a optimalizován.

Existuje několik obecných standardů souvisejících s metodami komprese, které se v různých oblastech práce s digitálním obrazem staly normami.

4.2.4 Parametry videosouboru

U videosouboru si můžeme přečíst ihned jeho název a jeho příponu, z které většinou odvodíme použitý kontejner. Kontejner spojuje do jednoho souboru různé streamy, což je základní součást multimediálního souboru. Stream, jinými slovy datový tok, může být video, zvuk, titulky nebo kapitoly. Kontejnery se liší v tom, jaké typy a v jakém

formátu streamy podporují. Mezi nejznámější kontejnery patří AVI, MPEG, VOB, MKV, MP4, OGM, MOV, WMV nebo RM. Pro přehrání každého video souboru potřebujeme příslušný splitter. Slouží přehrávačům pro rozdělení kontejneru na jednotlivé streamy, které poté předá patřičným kodekům. V OS systémech Windows jsou již po instalaci splittery pro formáty AVI, MPEG, ASF a WMV. Pro přehrávání dalších souborů musíme splitter doinstalovat. Současně potřebujeme také dekodér pro každý stream v souboru. Například AVI tedy není komprese videa, ale pouze kontejner, v kterém může být video a zvuk v téměř libovolné kompresi. Ale ne všechny kontejnery nám dávají takovou volnost pro použití kodeků jako kontejner AVI. Největší výběr kompresí nabízí kontejner MKV. Naopak kontejner MPEG dovoluje u videa použít pouze komprese MPEG-1 a MPEG-2. Podobně omezené jsou i kontejnery WMV nebo MP4. Pokud potřebujeme dekodér pro určitý video soubor, musíme zjistit, jakým enkodérem byl zkomprimovaný nebo alespoň v jakém standardu. Standardů pro kompresi videa je mnoho, mezi běžné patří MPEG-1, MPEG-2, MPEG-4, MJPEG nebo DV. Pro některé formáty však existuje pouze jeden kodek a pojmy kodek a formát (komprese) jsou často zaměňovány. MPEG4, tedy není žádný kodek, ale pouze standard, kterého se drží několik různých kodeků (například FFDSHOW, DivX, XviD, 3ivX nebo Nero Digital). Teoreticky je tedy možné dekodovat kterýmkoliv z těchto kodeků video zkomprimované nějakým jiným. Kompatibilita mezi různými kodeky však není úplná. DivX dekodér má například problémy s dekodováním XviDu. Nejlepším řešením je pracovat s tím kodekem, kterým bylo video zkomprimované.

Informaci, jakým kodekem bylo video zkomprimované lze získat z FourCC kódu, čtyřpísmenné značky, která je v hlavičce AVI souboru. I modernější systém identifikace kodeků nazývaný Codec ID uchovává pro zpětnou kompatibilitu i FourCC. Tento kód dokáže zjistit například programy GSpot nebo Video ToolBox. Tyto programy rovnou vypíší, ke kterému kodeku daný kód patří. Mnoho problémů při přehrávání videa je způsobeno tím, že dekodérů máme pro daný formát v systému více a použije se ten nesprávný. Například video zkomprimované kodekem DivX 5 (FourCC kód DX50) přehraje kromě originálního dekodéru i XviD, FFDshow, NeroDigital a 3ivX, ale ne vždy bez problémů. Proto je vhodné instalovat pouze ty kodeky, které opravdu potřebujeme. Důležité je mít přehled o tom, co se v systému nachází. Každý kodek si při instalaci zaregistruje jeden nebo více FourCC kódů, pro které se bude používat. V systému si kodeky přiřadí určitou prioritu, která určí, který z nich se použije

v případě, že se jich o dekodování určitého streamu hlásí více. Tyto priority dokáže měnit například program Radlight Filter Manager.

4.2.5 Kompresní systémy

DV (nebo **DVC**), první spotřební digitální formát. DV standard užívá pevný kompresní poměr 5:1 založený na diskrétní kosinové transformaci. Tento způsob kódování je jakýmsi mezistupněm mezi Motion-JPEG (MJPEG) a MPEG kompresí. Jiný než uvedený kompresní poměr nelze u tohoto systému nastavit a rovněž datový tok 25 Mbit/s je u tohoto systému konstantní. Rozhodně však kompresní formát DV není s MJPEG ani s MPEG2 totožný, i když vychází z podobných principů. Spotřebitelská verze (DV) dává jednu z největších hustot záznamu na magnetickém médiu. Systém DV je určen výhradně pro spotřební oblast, užívá vzorkovací kmitočet 13,5 MHz a rozlišení 8 bitů. Jasový a barvosný signál je zaznamenáván odděleně. Každý snímek je rozložen do 12 šikmých stop širokých 10 mikrometrů. Buben s hlavami se otáčí rychlostí 9000 otoček za minutu, jeho průměr činí 21,7 mm a obvodová rychlost pak 10,22 m/s. Zvuk je možno zaznamenávat u DV videorekordérů do dvou kanálů (se vzorkovacím kmitočtem 48 kHz) nebo do čtyř kanálů (tzn. 2 x stereo) se vzorkovacím kmitočtem 32 kHz a kvantováním 12 bitů. Systém DV používá kvantizaci (YUV) 4:2:0, datový tok je 25 Mbps. Jedná se o formát videozáznamu v digitálních kamerách, který využívá hardwarovou DCT kompresi obrazu a používá rozlišení obrazu 720x576 pixelů, snímkovou frekvenci 25 snímků/s a prokládaný obraz při datovém toku 3.6 MB/s.

DVCAM

Nadstavba systému DV pro profesionální oblast, stejný systém jako DV, datový tok 25 Mbps je totožný, má 1,5násobnou transportní rychlost pásku a větší šířku záznamových stop na pásku (15 mikrometrů), tím je zaručena vyšší spolehlivost záznamu. DVCAM používá rozměrově naprosto shodné pásky jako DV (avšak kvalitnější). Kvantizace je 4:1:1. DVCAM je kompatibilní s DV "směrem dolů", systém DVCAM může číst kazety systému DV.

DVCPRO

Propracovanější varianta systému DV od firmy Panasonic. Má dvojnásobnou transportní rychlost pásku a větší šířku stopy (18 mikrometrů oproti 15 mikrometrům). Kvantizace dosahuje profesionálních 4:2:2. Systém dosahuje standardu D-7 podle SMPTE.

MJPEG - Motion-JPEG

Tento algoritmus komprimuje každé okno zvlášť, což jej činí vhodným především pro diskové stříhové systémy, tedy pro nelineární videoeditaci. MJPEG stříhové systémy jsou vybaveny vlastním hardwarovým kodekem (tedy čipem na desce), proto při používání těchto typů stříhových karet není centrální procesor počítače příliš zatěžován. Činnost a výkonnost centrálního procesoru se projeví pouze v případech, kdy chceme do záběrů začlenit nějaký digitální efekt (například titulek, prolínání a jiné). V tomto případě vytváří tyto efekty procesor počítače. Výhodou MJPEG stříhových karet je možnost proměnlivého nastavení stupně komprese podle požadované výstupní obrazové kvality zpracovávaného videa i podle volného místa na disku počítače.

MPEG - Motion Pictures Experts Group

Vychází se zde z poznatku, že na dvou sousedních snímcích záznamu se změní jen menší část informace, zatímco většina zůstává stejná, stačí tedy zaznamenávat pouze tyto změny. Záznam se skládá z jakýchsi základních obrazů typu I (Intra Pictures), které nesou plnou obrazovou informaci a jsou komprimovány jen málo, dále z více komprimovaných obrazů P (Predicted Pictures) kódovaných podle předchozích obrazů a obrazů B (Bidirectional Pictures) s vysokým stupněm komprese a kódovaných podle předchozích i následujících snímků. Sled snímků pak vypadá přibližně takto: IBBPBBPBBPBBBI... Je zřejmé, že do takto upraveného záznamu nelze vstupovat na libovolném místě, ale pouze v místě plných, tzn. I snímků. Proto se tato metoda hodí pro záznam, na rozdíl od MJPEG však nikoliv pro počítačový stříh a editaci. MPEG1 umožňuje kompresi až 1 : 200, samozřejmě při této krajní hodnotě za cenu značných ztrát. MPEG1 je systém komprese digitálního videozáznamu či animace na média v kvalitě blízké se videosystému VHS, především pro videosignál zaznamenávaný na video CD nebo CD-I. Je součástí normy pro dnes již opouštěný záznam pohyblivého obrazu na CD-ROM, a dovoluje na disk zaznamenat asi 72 minut videa. Při této normě se pracuje s obrazem o rozlišení 352 x 228 obrazových bodů.

MPEG-1

Kodek vytvořený Motion Pictures Experts Group (MPEG) v roce 1989, která si dala za cíl standardizovat metody komprese videosignálu a vytvořit otevřenou a efektivní kompresi. Byl navržen pro práci s videem o rozlišení 352x288 bodů a 25 snímků/s při datovém toku 1500kbit/s. Ke kompresi videa využívá tři typů snímků: I, P a B. Klíčové I-snímky (Intra Pictures) jsou komprimovány obdobně jako MJPEG, ale navíc s možností komprimovat různé části obrazu různým stupněm komprese. P-snímky

(Predicted Pictures) jsou kódovány s ohledem na nejbližší předchozí I nebo P-snímek. B-snímky (Bidirectional Pictures) jsou pak dopočítávané jako rozdílové snímky mezi nejbližším předchozím I nebo P-snímekem a nejbližším následujícím I nebo P-snímekem. Jeho výhodou je vysoká kompatibilita (používá se pro VCD) a je vhodný i pro stream videa. Na druhou stranu nepodporuje prokládaný obraz, nabízí pouze konstantní datový tok a nízkou kvalitu při nízkém datovém toku. Díky své metodě komprese není vhodný pro editaci videa.

MPEG-2

Pokročilá a profesionální metoda komprese signálu umožňující oproti MPEG1 výrazně vyšší rozlišení a proměnný tok dat podle dynamiky obrazu. Je zde spojena redukce objemu dat na minimum s eliminací nadbytečných informací. Klíčové informace se komprimují v každém patnáctém pulsnímku nebo každou půlsekundu a zachycují se pouze rozdíly mezi snímky jdoucími za sebou. MPEG2 se používá v systémech digitálního záznamu videa DVD a jeho základ spočívá i v digitálním TV vysílání. Nespornou výhodou MPEG2 je jeho široká využitelnost a univerzálnost. Na rozdíl od svého předchůdce nabízí podporu prokládaných snímků a podporu variabilního datového toku. Při vysokém datovém toku (6 Mbit/s a více) dosahuje vysoké kvality obrazu. Při stejném datovém toku a rozlišení 720x576 nabízí výrazně vyšší kvalitu obrazu než kodek MPEG1. V nízkých rozlišeních se však tyto rozdíly stírají a MPEG2 je navíc mnohem náročnější na hardware. Stejně jako jeho předchůdce není vhodný pro editaci videa. Používá se pro DVD, SVCD a pro digitální vysílání.

MPEG-3 a MPEG-4

MPEG3 se používá pro kompresi zvuku. Dokáže podstatně snížit objem dat reprezentujících zvuk. Při této kompresi však dochází k mírné degradaci původního záznamu. MPEG4 je určen především pro přenos obrazových dat po telefonních linkách a sítích ISDN a zejména pro velmi malé datové toky. Na principech MPEG4 je postaven i formát ASF. Kvalita zobrazení je poplatná velké kompresi videosignálu. Z kodeku MPEG4 vznikl i nový standard DivX. Nejedná se o přesnou definici komprese a komprimačních algoritmů, ale pouze o množinu parametrů a vlastností, které musí kompresor splňovat, aby byl MPEG4 kompatibilní. Různé implementace si vždy vybírají z definice MPEG4 vždy jen to, co se jim hodí. Nejznámějšími kodeky, které využívají kompresi MPEG4, jsou Microsoft MPEG4 v1, v2 a v3, DivX 4, DivX 5, XviD. Na střih videa se nehodí pro svou velmi nízkou kvalitu a nároky na přehrávání. Aby obraz vypadal lépe, musí se používat softwarové filtry na zahlazení a jiná

vylepšení obrazu při přehrávání. Tato komprese neobsahuje synchronizační údaje, takže MPEG4 video nelze nijak upravit a také kvalita obrazu je zcela nedostatečná. Poměrně kvalitního obrazu dosáhneme pouze s použitím neprokládaného čistého obrazu jako zdroje. S tímto formátem nepracuje žádná stříhová karta ani hardwarový dekodér, hodí se pouze pokud chceme svoje výtvary někomu nahrát po internetu (zde se uplatní zejména malá velikost souborů).

DivX

DivX se nadále zdokonaluje a existuje několik verzí tohoto kodeku. Obecně se jedná o velice dobrý kompresní poměr, kdy za cenu relativně nepatrného snížení kvality je možno uložit "klasický" DVD video film na jedno CD. Kvalitou převyšuje MPEG1, ale jsou tady také nevýhody. MPEG4 není standardní formát, a tudíž jej nelze přehrávat na všech stolních přehrávačích a jediný systém, kde lze tuto metodu použít, je OS Windows 98 a výše (Windows Media Player 7). Další nevýhoda je ve zvuku, vzhledem k tomu, že vše je uloženo ve formátu AVI, nelze počítat s dokonalým prostorovým zvukem. Poslední menší nevýhodou je přehrávání, celý proces značně zatěžuje pouze CPU, který je hodně vytížen. DivX se obecně používá k převedení datového toku MPEG2 používaného v DVD video filmech na médium CD-R. Protože DivX je formát kompatibilní s MPEG4, nabízí vysokou kvalitu obrazu při nízkém datovém toku a podporuje barevné formáty YUV a RGB. Též umožňuje použít jakékoli rozlišení dělitelné čtyřmi až do 1920x1088 a nabízí pokročilé kompresní techniky.

XviD

Počátky tohoto formátu jsou spjaty s formátem DivX. V okamžiku, kdy se OpenDivX stal uzavřeným, vzalo několik programátorů pracujících na OpenDivXu zdrojové kódy ještě otevřeného OpenDivXu a osamostatnili se. Na základě těchto zdrojových kódů začali vyvíjet vlastní verzi kodeku nazvanou XviD. Jedná se stejně jako u DivXu o kodek kompatibilní s MPEG-4 a implementuje mnoho jeho vlastností. Protože se však jedná o Open-Source projekt podílí se na jeho vývoji programátoři z celého světa.

Quicktime

Kodek vyvinutý společností Apple. Quicktime po nástupu formátu MPEG ustoupil do pozadí a není již tak rozšířený. Je použitelný pro PC i Macintosh a využívá kompresi 1:5 až 1:25. Jeho nevýhodou je skutečnost, že k jeho použití je potřeba specializovaný přehrávač, který si musíte stáhnout z webových stránek společnosti Apple.

RealVideo

Formát od společnosti Real Networks. Formáty komprese od společnosti Real Networks, které nesou označení Real Video a Real System G2, nabízí obdobné vlastnosti jako Quicktime, ale jsou více zaměřeny na kompresi streamovaného videa. V současné době se využívají ke komprimaci videa pro mobilní telefony.

ASF, WMV

Kodeky od společnosti Microsoft. Těmito kodeky zareagovala společnost Microsoft na úspěchy společností Apple a RealNetworks v oblasti streamovaného videa. ASF (Advanced Streaming Format) je vývojově starší a je určen především pro stream video. Vychází z formátu AVI a částečně implementuje kompresi MPEG-4 (nepodporuje). Novější verzí ASF je formát WMV. [6]

5 Převody animací

Při převodu animací do AVI souboru lze postupovat několika způsoby. Jedním z nich je převádět konkrétní animaci, například swf, pps atd., na kterou však potřebujeme konkrétní program pro převod. Bohužel pro různé druhy animací musíme použít i různé druhy programů k převodu. Neexistuje žádný univerzální program. A někdy potřebný program pro převod dokonce neexistuje vůbec. Jinou možností je snímat aktivity na obrazovce nebo pouze zájmového okna. Tento způsob je univerzálnější, neboť můžeme převést jakoukoli animaci, která jde přehrát. Uvedeme si příklad převodu animace SWF do AVI.

Program SWF Toolbox

Jako první jsem použil program SWF Toolbox verze 3.1, ke stažení na stránkách "<http://www.eltima.com/products/swf-tools>". Pracoval jsem pouze s demo verzí, která do výsledného videa vkládá titulek o tom, že nepracujeme s registrovaným programem. Program umí, kromě převodu do AVI, například převod do EXE souboru, sérii obrázků, kompresi SWF. Pro převod lze v aplikaci vybrat kodek, jaký bude použit, nastavit kvalitu záznamu, k záznamu zachytit zvuk. Převáděl jsem animaci o velikosti 967 kB. Poprvé jsem použil kodek XviD MPEG-4, u kterého jsem ponechal výchozí nastavení a kvalita záznamu (při tomto kodeku) v programu již měnit nešla. Záznam byl proveden se zachycením stereo zvuku v nejnižší kvalitě. Výsledné AVI video mělo velikost 9,63 MB (bez zvuku asi 7 MB). Jako další jsem použil kodek Microsoft Video 1 a ponechal výchozí nastavení kvality, zvuk byl opět stereo v nejnižší kvalitě. Velikost výsledného souboru byla 30,4 MB. Velikost výsledného souboru tedy velmi záleží na zvolené kompresi a zvoleném kodeku. S programem se pracuje velmi dobře a převody proběhly bez problémů, avšak jde o placenou verzi.

Program swf>>avi converter

Tento program je zdarma. Neumožňuje však téměř žádná nastavení. Animace o velikosti 967 kB měla po převedení, při zachování výchozích nastavení, výslednou velikost 1,95 GB. Velikost lze ovlivnit velikostí výsledného okna a nastavením snímků. I při snaze snížit velikost souboru na minimum byla stále velikost okolo 100 MB a proto je tento program v praxi nepoužitelný nebo by bylo nutné s výsledným souborem dále pracovat. Vhodným programem pro úpravu a nahrávání videa je program Virtual Dub, který je šířen pod GPL licencí.

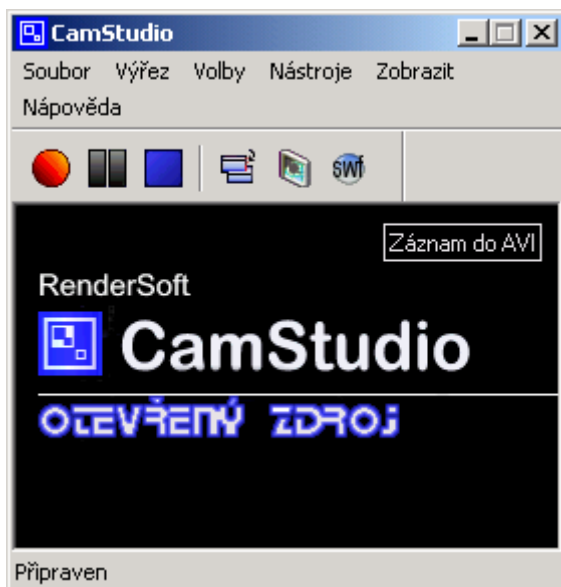
Dále je možno postupovat tak, že mnoho programů volně ke stažení zvládne animaci rozložit na jednotlivé obrázky. Z těch lze poté složit výsledné video. Tento postup je však poměrně časově náročný.

Další možností je snímání dějů na obrazovce. Těchto programů je volně ke stažení již větší množství. Já jsem se zaměřil a rozhodl podrobně popsat program CamStudio. Při hledání jsem totiž našel mnoho programů tomuto velmi podobných. Například DemoStudio, Screen Movie Studio a další.

Program CamStudio 2.0

RenderSoft CamStudio je nástroj pro nahrávání aktivit na obrazovce do standardních video AVI souborů. Může být použit pro demonstrace programů, nahrání pohybu kurzoru, výběry z menu, textové vstupy. Program umožňuje výsledek zaznamenat kromě AVI formátu, také do formátu SWF. Systémové požadavky programu jsou: Microsoft Windows 95, 98, Me, NT 4.0, 2000, XP; 400 MHz procesor; 64 MB RAM; 4 MB místa na disku.

Na obrázku 5-1 je zobrazena základní nabídka programu po jeho spuštění.



Obrázek 5-1 Základní nabídka programu

Popis ovládání programu

Pod panelem nabídek se nachází základní ovládací tlačítka.

První tlačítko slouží ke spuštění záznamu. V závislosti na tom, jakou máme vybránu oblast záznamu se záznam spustí nebo jsme vyzváni k dalšímu výběru. Druhé tlačítko je pro pozastavení záznamu. K jeho opětovnému spuštění je potřeba kliknout na první

tlačítko – záznam. Třetí tlačítko slouží k ukončení záznamu, po jeho stisknutí program vyzve k uložení výsledného souboru. Další tlačítko přepíná vzhled programu mezi normálním, kompaktním a pouze zobrazením tlačítek. Další tlačítko zobrazí dialog vysvětlivek obrazovky. Poslední páté tlačítko přepíná formát výstupního souboru (SWF nebo AVI).

Panely nabídek:

- **File - Soubor**

Record - Záznam – spustíme nahrávání. Již před spuštěním musíme mít nastaven výstupní formát, a také oblast, kterou chceme zaznamenat. V hlavním okně programu, pokud jej nemáme minimalizovaný, program zobrazuje informace, jako například uběhlý čas od počátku nahrávání, jaký kodek je použit, číslo aktuálního snímku, rychlost snímků za sekundu a informace o grafickém nastavení.

Stop - Stop – zastavíme záznam a vyskočí nám okno, kde zadáme kam soubor chceme uložit a jeho název. Po jeho zadání dochází k ukládání souboru a pokud máme nastaveno k jeho přehrání.

Pause - Pauza – pozastaví záznam, pro jeho opětovné spuštění se stiskne tlačítko Záznam nebo pro jeho zastavení tlačítko Stop.

Exit - Konec – ukončí program

- **Region - Výřez**

Výřezem určíme jakou oblast budeme zaznamenávat.

Region - Oblast – při této volbě, po spuštění záznamu, se objeví kurzor, kterým tažením po obrazovce vybereme požadovanou oblast snímání.

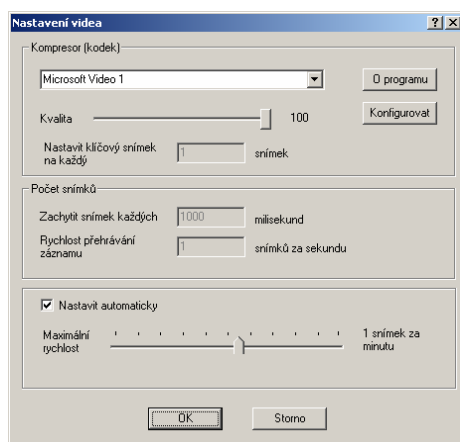
Fixed Region - Pevná oblast – tlačítkem Vybrat vybereme oblast, tzn. nemusíme zadávat souřadnice oblasti. Další možností je zadat souřadnice oblasti ručně, a to buď s pevně zadaným levým horním rohem nebo po spuštění Záznamu oblast stanovíme. Ještě zbývá volba, kdy je oblast pevně stanovena po dobu záznamu nebo můžeme s oblastí posouvat.

Full Screen - Celá obrazovka – tato volba nastavuje oblast snímání na celou obrazovku.

- **Options - Volby**

Video Options - Nastavení videa – tato volba nám umožňuje nastavit možnosti videa (obrázek 5-2). Z nabídky můžeme vybrat kodek, který bude použit a ten podle možností daného kodeku nastavit tlačítkem Konfigurovat. Posuvníkem lze nastavit

kvalitu záznamu. Dále lze nastavit buď automaticky nebo ručně počet snímků a interval klíčového snímku.



Obrázek 5-2 Nastavení videa

Cursor Options - Nastavení kurzoru – kurzor může být skrytý nebo zobrazený. Pokud je kurzor zobrazen, lze vybrat typ kurzoru, případně jej ještě zvýraznit.

Do not record audio - Nezaznamenávat zvuk

Record audio from microphone - Záznam zvuku z mikrofonu

Record audio from speakers - Záznam zvuku z reproduktorů – označením jedné z těchto možností určíme odkud se bude zaznamenávat zvuk nebo zda bude záznam neozvučen.

Audio Options - Nastavení zvuku

Audio Options for Microphone - Nastavení zvuku pro mikrofon – zde můžeme vybrat zvukové zařízení, nastavit hlasitost záznamu, nastavit formát záznamu, formát komprese a nastavit proložení zvuku a videa

Audio Options for Speakers - Nastavení zvuku pro reproduktory – lze vybrat zvukovou kartu, ke které jsou připojeny reproduktory. Nastavit hlasitost reprodukce nahraného záznamu. Také je zde volba pro konfiguraci nahrávací linky - lze nastavit automaticky nebo ručně.

Audio and Video Synchronization - Synchronizace zvuku a videa – zde jsou dvě volby. Neměnit audio nebo video proudy, nebo Nastavit předcházení zvuku před videem, který nastavujeme ručně v milisekundách.

Enable Autopan - Povolit posun výřezu – při zaškrtnutí této volby a zároveň snímání pouze určité oblasti, se tato oblast přesouvá s kurzorem myši po obrazovce.

Autopan Speed - Rychlost posunu výřezu – souvisí s nabídkou Povolit posun výřezu. Nastavuje se zde rychlost posunu okna (1-200) za kurzorem myši.

Program Options - Nastavení programu

Minimize program on start recording - Minimalizovat po startu záznamu – po startu záznamu se program minimalizuje do oznamovací oblasti na liště. Zde lze program dvojitým kliknutím obnovit nebo pravým tlačítkem myši ovládat (Záznam, Stop, Pauza, Konec).

Hide flashing rectangle during recording - Skrýt obdélník výřezu po záznamu – při záznamu určité oblasti lze touto volbou skrýt nebo zviditelnit značky, které určují snímanou oblast.

Save settings on exit - Uložit nastavení při ukončení – při zaškrtnutí této volby se uloží aktuální nastavení programu po jeho ukončení.

Capture translucent/layered Windows - Zachycovat průhlednou vrstvu – ve Windows 2000 nebo XP se mohou vyskytovat průhledná okna, která nebudou zachycena, když tato volba nebude zaškrtnuta.

Play AVI file when recording stops - Přehrát AVI soubor po ukončení záznamu – při zaškrtnutí volby přehraje záznam po ukončení nahrávání.

Temporary directory for recording - Dočasný adresář pro záznam – pro dočasný adresář lze vybrat ze tří voleb. Použít adresář systému – obvykle adresář temp daného profilu. Použít adresář programu – program si vytvoří soubor „~temp.avi“ tam, kde je nainstalován. Poslední možností je použít libovolný adresář.

Recording Thread Priority - Priorita záznamu – některé programy vyžadují velké systémové prostředky, v takovém případě se program CamStudio stává nepoužitelným. Lze proto nastavit prioritu programu, ale může se tím ovlivnit výkon celého systému.

Name of AVI file – Jméno AVI souboru – vyzve k zadání cesty a jména souboru nebo lze nastavit automatické pojmenování souboru. Soubor je uložen do adresáře, kde je program nainstalován.

Troubleshoot - Nejčastější problémy – pod touto volbou jsou popsány nejčastější problémy a jejich zaškrtnutím daný problém řešit.

Record to Flash Options – Nastavení SWF záznamu – nastavení voleb pokud je záznam nahráván do SWF.

Keyboard Shortcuts - Klávesové zkratky – seznam a nastavení klávesových zkratk programu.

- **Tools - Nástroje**

Screen Annotations - Vysvětlivky obrazovky – umožňuje opatřit obrazovku nákresey, poznámkami a doplňkovým textem.

Video Annotations - Vysvětlivky videa – vyžaduje nainstalovanou webovou kameru a umožňuje do záznamu přidat záběry z webové kamery.

SWF Producer (Converts AVI to flash) - SWF tvůrce (konvert AVI na Flash) – pod touto volbou se skrývá program, který provádí konverzi AVI souboru na SWF soubor (obrázek 5-3).



Obrázek 5-3 SWF tvůrce – převod SWF na AVI

- **View - Zobrazit** – přepíná vzhled programu mezi následujícími možnostmi.

Normal View - Normálně

Compact View - Kompaktní vzhled

Buttons View - Pouze tlačítka

- **Help - Nápověda**

Help - Témata – obsahuje nápovědu k programu

CamStudio Website - Domovská stránka CamStudio – otevře domovskou webovou stránku programu – www.rendersoftware.com/products/camstudio.

Frequently Asked Questions - Nejčastější dotazy – odkaz na webovou stránku s nejčastějšími dotazy – www.rendersoftware.com/products/camstudio/faq.htm.

About - O programu – informace o programu.

Licence programu

Program je šířen pod GNU GPL licencí.

GNU General Public License. Software šířený pod licencí GPL je možno volně používat, modifikovat i šířit, ale za předpokladu, že tento software bude šířen bezplatně (případně za distribuční náklady) s možností získat bezplatně zdrojové kódy. Toto opatření se týká nejen samotného softwaru, ale i softwaru, který je od něj odvozen. Na produkty šířené pod GPL se nevztahuje žádná záruka. Licence je schválená sdružením OSI a plně odpovídá Debian Free Software Guidelines.

Práce s programem

Práce s programem CamStudio je poměrně jednoduchá. Program není příliš složitý a ovládání se rychle osvojí. Program jsem testoval i na notebooku Acer TravelMate s procesorem Intel Celeron 1.33GHz, 120 MB RAM. Toto je poměrně slabá konfigurace na práci s grafikou, přesto jsem se nesešel s vážnějšími problémy při záznamu. Programy při převodu využívají CPU na plný výkon a také zabírají velkou část paměti. Výsledné video velmi ovlivňuje nastavení programu, zejména zvolení správného kodeku a nastavení kvality záznamu, případně použití záznamu se zvukem. Záznam animace s kodekem Microsoft Video 1 (použito výchozí nastavení a byl zaznamenáván zvuk) o délce 2 minuty 20 sekund měl velikost asi 74 MB. Oproti tomu při stejných parametrech, ale při použití kodeku XviD mělo video délku 30 MB. Při použití kodeku XviD, ale bez záznamu zvuku mělo výsledné video délku asi 3 MB. Program neumožňuje dostatečné nastavení zvukových voleb (kompresi), proto při použití se záznamem zvuku je výsledný soubor příliš veliký.

6 Závěr

Cílem této práce bylo seznámení se základy počítačové grafiky, základní strukturou grafických formátů, digitálního videa a převod animací do video formátu. Obecně práce s grafikou, převody mezi formáty nebo zpracování videa je poměrně hardwarově a časově náročná záležitost. Ne vždy je výsledek uspokojivý, někdy dokonce nemožný. Do budoucna se dají očekávat zvyšující se nároky na kvalitu záznamů. Co nejdříve obraz a zvuk. S tím ovšem souvisí velké objemy dat a způsob jakým data přenášet. Budou vznikat nové formáty, které budou odpovídat rozvoji hardwarových technologií. Kapacitní požadavky na nosiče dat se budou zvyšovat. S tím souvisí snaha o co největší míru komprese při zachování stávající kvality. Zejména pak při použití internetu, kde se dá očekávat jeho využití při distribuci a přenášení dat.

Seznam literatury

- [1] Počítačová grafika, Ing. Stanislav Horný, CSc., Vysoká škola ekonomická v Praze
- [2] James D. Murray, William vanRyper Encyklopedie grafických formátů druhé vydání, Computer Press
- [3] Jiří Žára, Bedřich Beneš, Petr Felkel, Moderní počítačová grafika, Computer Press
- [4] <http://cmykmarker.dtpstudio.cz> (6.7.2006)
- [5] <http://www.wikipedia.cz> (20.10.2006)
- [6] <http://www.video.az4u.info> (7.10.2006)
- [7] <http://www.rendersoftware.com/products/camstudio> (22.10.2006)