

KAPITOLA 4

Střih a korekce videa ve formátu DivX

V této kapitole najdete:

- ☞ VirtualDub – možnost úprav, střihu a výstupu videa do DivX
- ☞ VirtualDubMod – rozšířená verze programu VirtualDub a její možnosti
- ☞ Media Info – informace o druhu kontejneru a použitých kodecích
- ☞ Avi Mux Gui – software na tvorbu AVI a Matroska kontejnerů
- ☞ O postprodukčním nástroji AviSynth
- ☞ Sada filtrů pro zobrazení DivX a Xvid – FFDSHOW
- ☞ DivXRepair – utilita vyhledávající zamrzlý obraz či jeho rozpad do barevných polí
- ☞ DivXRepair – utilita vyhledávající zamrzlý obraz či jeho rozpad do barevných polí

VirtualDub – možnost úprav, střihu a výstupu videa do DivX

VirtualDub je program, či dnes už spíše rodina programů, které nám umožňují pokročilou práci s videem, střih, postprodukční úpravy a také výstup videa do DivX či Xvid formátu. Přitom je tento chytrý program šířený zdarma. Mluvili jsme o něm už v souvislosti se zachytáváním videa z externího zdroje.

V následujících řádcích budeme pracovat s verzí *VirtualDub 1.6.18*. Základní okno programu je na obrázku 4.1.



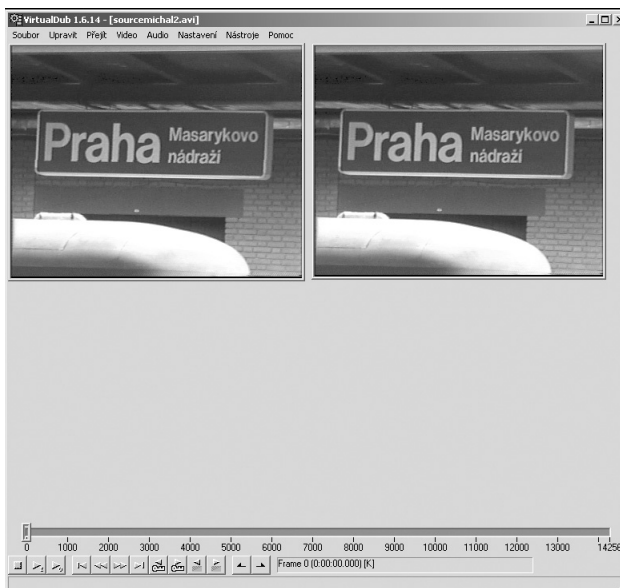
Obrázek 4.1 Základní okno programu VirtualDub

VirtualDub podporuje video ve vstupních formátech **AVI**, **ASF** anebo **MPEG-1**. Pomocí kompresních kodeků v něm pak můžeme vygenerovat video ve formátu AVI. Kromě již zmíněných funkcí obsahuje **VirtualDub** také **capture** modul pro zachytávání (grabování) videa z externího zdroje. V následujících řádcích se podíváme na jednotlivé, pro nás zajímavé funkce tohoto programu podrobněji. Nepůjde však přitom o vyčerpávající výčet, spíše jen o základní seznámení. Ostatně celá tato kniha si neklade za cíl být vyčerpávajícím popisem množství softwarových nástrojů, ale spíše jednoduchým, přehledným a praktickým uvedením do problematiky.

POZNÁMKA: Pojem capture znamená takzvané grabování (digitalizaci a ukládání) videa do digitálního formátu na harddisk. Natahujeme-li tedy video z kamery, video-přehrávače či jiného externího zdroje, provádíme grabování.

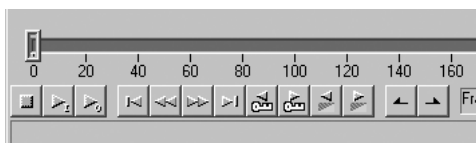
Nejprve se podíváme, jak **VirtualDub** pracuje při přehrávání a střihu videa.

1. Načteme video pomocí volby **Soubor / Otevřít video soubor**.
2. Otevřou se nám dvě instance videa, levá zobrazuje video zdrojové, pravá video výstupní, jako vidíme na obrázku 4.2.



Obrázek 4.2 Zobrazení vstupního a výstupního videa v programu VirtualDub

3. Pokud je velikost obou videí větší, než je pro náhled užitečné, kontextovým menu, které vyvoláme pravým tlačítkem nad každým z obou náhledů, zvolíme poměrné zmenšení (například 50%).
4. Chceme-li video přehrávat, najdeme vlevo dole obvyklé tlačítka **Stop** a **Play** (viz obr. 4.3) ve zdvojené podobě. První z nich se značkou 1 přehrává pouze vstupní video, druhé se značkou 0 přehraje jak vstup, tak výstup v reálném čase.



Obrázek 4.3 Tlačítka pro přehrávání videa v programu VirtualDub

5. Následují tlačítka pro posun videa (videí) na začátek či na konec a tlačítka na posun po jednotlivých snímcích.
6. Samozřejmě můžeme video také posouvat tažením jezdce v linii lemované počtem snímků, které video obsahuje.
7. Zajímavou možností, která již umožňuje zaměřit se na analýzu struktury **MPEG** videa, nabízí tlačítka pro posun dopředu a dozadu se značkou klíče. Tímto způsobem se ve videu posouváme jen po klíčových snímcích (**I-snímky**, viz tip v kapitole „Co je DivX“). Již pro představu, jak formát **MPEG**

(a tím i **DivX**, který je na něm postaven) funguje, je tato možnost velmi zajímavá, nemluvě o možnosti podrobné analýzy videa.

8. Následující dvě tlačítka slouží pro rychlý posun k další výrazné změně v obrazu, tedy projíždí zrychleně scény, které jsou mezi dvěma klíčovými snímky.

Jednou z důležitých vlastností programu **VirtualDub** je možnost střihu. Není sice na úrovni profesionálních stříhových programů, protože funguje pouze metodou **insert**, ale pro účely jednoduchých stříhových operací plně dostačuje. Nejprve si vybereme počátek sekvence, **Mark In**, a konec sekvence, **Mark Out**, tedy počátek a konec sekvence, kterou chceme přemístit, vymazat atd., a následně s ní pomocí známých funkcí vyjmout (**Cut**), kopírovat (**Copy**), vložit (**Paste**) pracujeme. Tlačítka volby počátku a konce sekvence vidíme na obrázku 4.4.



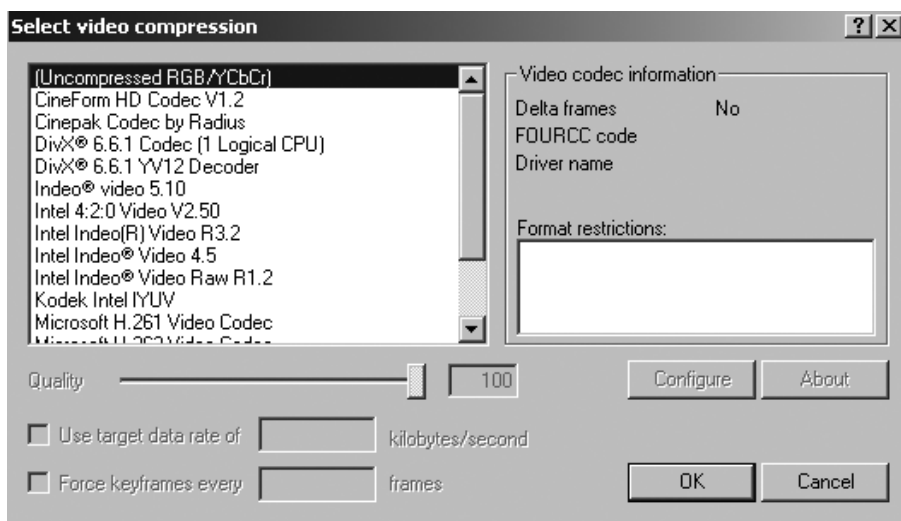
Obrázek 4.4 Tlačítka pro označení začátku a konce vyjímané sekvence v programu VirtualDub

Postup je následující:

1. Pomocí tlačítka **Mark In** označíme začátek sekvence.
2. Podobně tlačítkem **Mark Out** označíme konec požadované části videa.
3. Nyní můžeme přímo danou část exportovat pomocí volby v menu **File / Uložit segment AVI**.
4. Nebo můžeme danou část vymazat jednoduše tlačítkem **Delete**.
5. Případně můžeme danou část vyjmout pomocí klávesové zkratky **CTRL + X (Cut)**.
6. Tuto vyjmutou část pak můžeme přesunout a vložit pomocí **CTRL + V (Paste)** na jiné místo, kam jsme ukázali jezdcem pro přehrávání videa.
7. Jak je vidno, například rozdělení velkého videa na dvě části je díky tomuto postupu velmi jednoduchou operací.

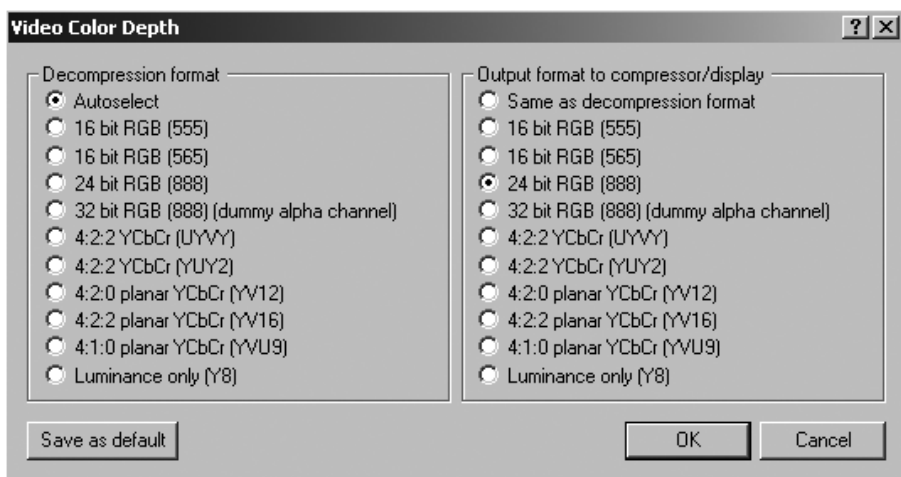
Nyní je nutno si říci, jaké možnosti exportu program nabízí. Nejprve se budeme zabývat videem, jehož všechny exportní nastavení najdeme v menu **Video**.

Nejdůležitější položkou je volba **Komprese**, pomocí níž zvolíme kodek. Ten se použije pro kompresi videa. Okno **Výběr komprese videa** vidíme na obrázku 4.5. Zde zvolíme kodek, pomocí něhož chceme výstupní video komprimovat. U každého kodeku se nám zobrazí jeho **FourCC** kód, máme možnost nechat si vypsát pomocí tlačítka Informace copyright a informace o tvůrcích kodeku. Pomocí tlačítka **Nastavení** pak nastavíme parametry komprese, jak bylo popsáno v kapitole 2.



Obrázek 4.5 Volba komprese v programu VirtualDub

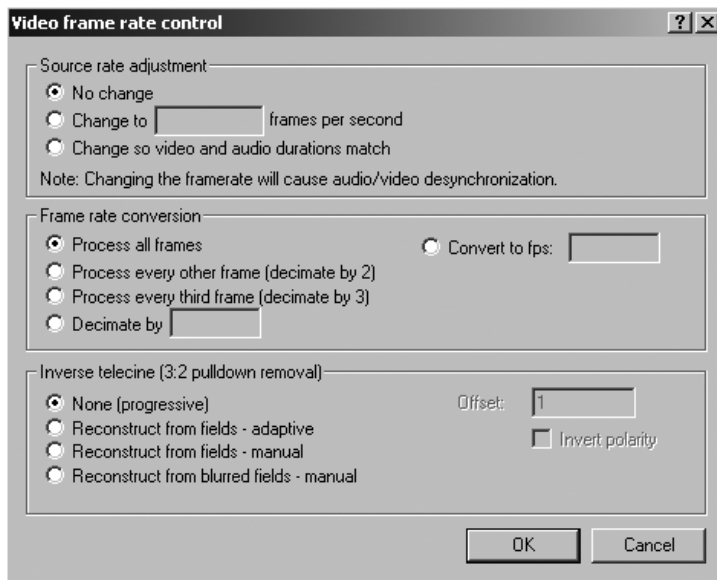
Další volbou z menu **Video** je volba **Barevná hloubka** dekomprese (zobrazení) a výstupního videa. Doporučuji nechat hodnoty nastavené na standardních, tak jak vidíme na obrázku 4.6.



Obrázek 4.6 Volba Barevné hloubky v programu VirtualDub

POZNÁMKA: Video můžeme mít v barevném módu RGB – podobně jako obrázky a jakoukoli grafiku na monitoru počítače. Tento mód je doporučený. Můžeme volit mezi 16, 24 a 32bitovou hloubkou. Ta znamená, kolika bity je reprezentován každý bod videa, v jak jemném spektru barev je tedy zobrazeno. Čím větší bitová hloubka, tím větší kvalita a také datový objem videa. Kromě tohoto módu se u videa používá často mód **YCbCr**, který je založen na vlastnostech vnímání barev pohyblivého obrazu lidským okem. Přitom **Y** znamená světelnost obrazu (**luma**) a **Cb, Cr** představují takzvané chromizační signály (**chroma-blue** a **chroma-red**). Tento mód se používá proto, že lidské oko dokáže velice dobře rozlišit změnu ve světelnosti, ale už nedokáže rozlišit malé změny v chromizačních signálech. Proto dokáže video v tomto módu přenést stejnou barevnou informaci při menším datovém objemu. Opět zde máme možnost volby barevné hloubky jednotlivých kanálů. Posledním módem je **Luminance only**, který slouží pro přenos informací o monochromatickém, tedy černobílém obrazu.

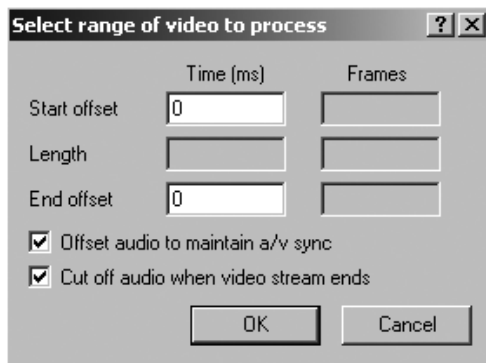
Pomocí volby **Rychlost** se nám otevře okno jako na obrázku 4.7 pod názvem **Nastavení rychlosti videa**. Zde máme možnost nastavit změnu počtu snímků za sekundu výsledného videa. Postupujeme takto:



Obrázek 4.7 Konverze rychlosti videa v programu VirtualDub

1. V oblasti nazvané **Změna rychlosti** nastavujeme proměnu počtu snímků za sekundu, buď tím, že vepíšeme do kolonky **Změnit na snímků za sekundu** požadovanou hodnotu, anebo zvolíme **Změnit podle délky videa a audia**, to v případě, že jejich délky jsou rozdílné.

2. Případně máme v oblasti nazvané **Konverze rychlosti** možnost nechat ve výstupu pouze druhý, třetí nebo x-tý snímek.



Obrázek 4.8 Výběr části videa v programu VirtualDub

3. V nabídce **Vyberte oblast videa ke zpracování**, kterou vidíme na obrázku 4.8, zvolíme, jaká oblast videa se má ve výstupu objevit. Jde o jinou, číselnou cestu ke zkrácení videa, které můžeme také provést již popsanou metodou stříhu.

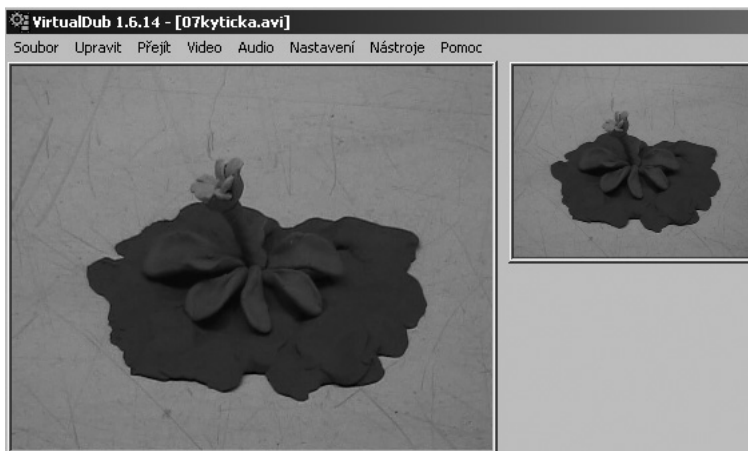


Obrázek 4.9 Volba Filtry programu VirtualDub

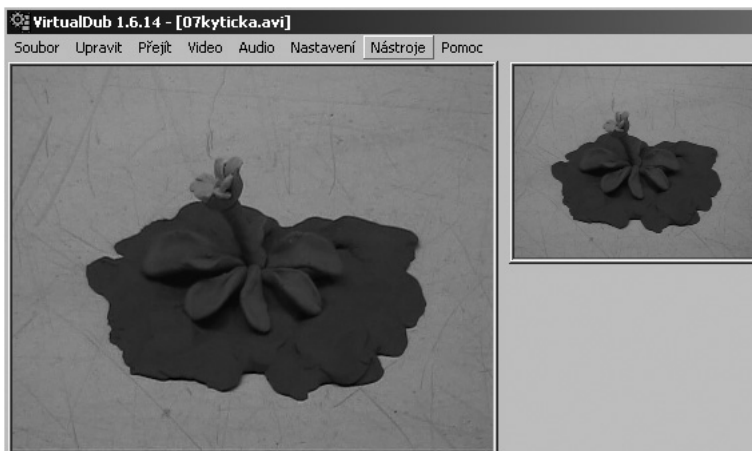
Velkou předností programu **VirtualDub** je množství vestavěných filtrů, které si vybíráme pomocí volby **Filtry**. Ta je znázorněna na obrázku 4.9. Tlačítkem **Přidat...** potom zvolíme jeden nebo více filtrů, které se budou na výstup aplikovat.

Uvedeme si jejich seznam spolu s obrazovou ukázkou výstupu:

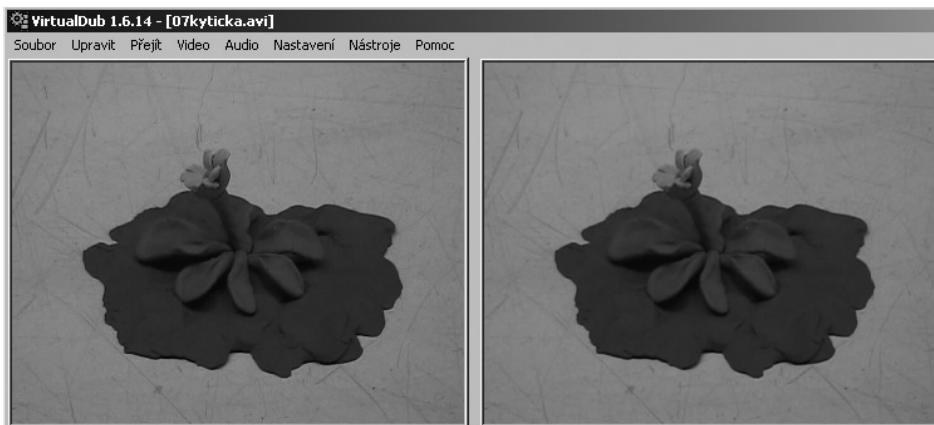
1	2 : 1 reduction	Redukce videa na poloviční rozlišení při zachování poměru stran.	obrázek 4.10
---	-----------------	--	--------------



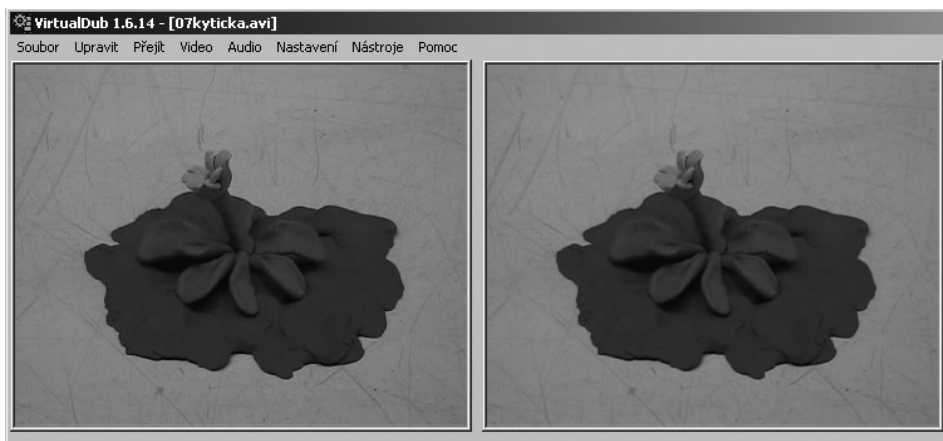
2	2 : 1 reduction (high quality)	Redukce videa na poloviční rozlišení při zachování poměru stran. Verze ve vyšší kvalitě.	obrázek 4.11
---	--------------------------------	--	--------------



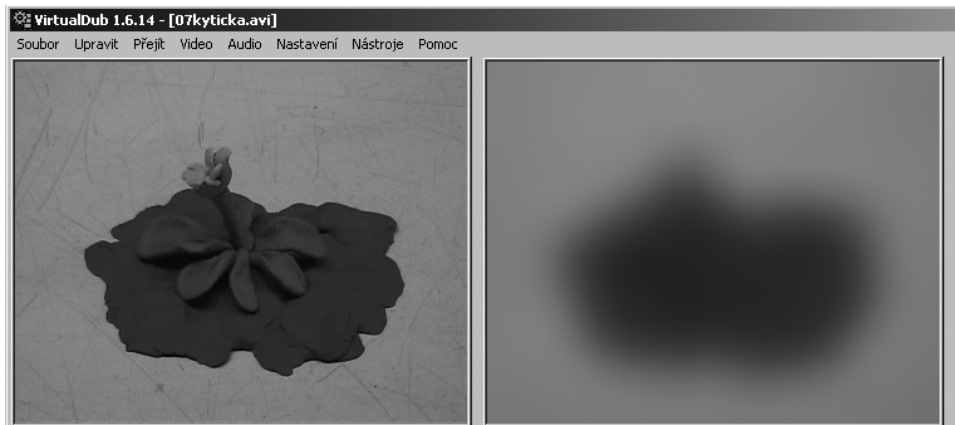
3	blur	Gaussovské rozostření videa s parametrem radius 1.	obrázek 4.12
---	------	--	--------------



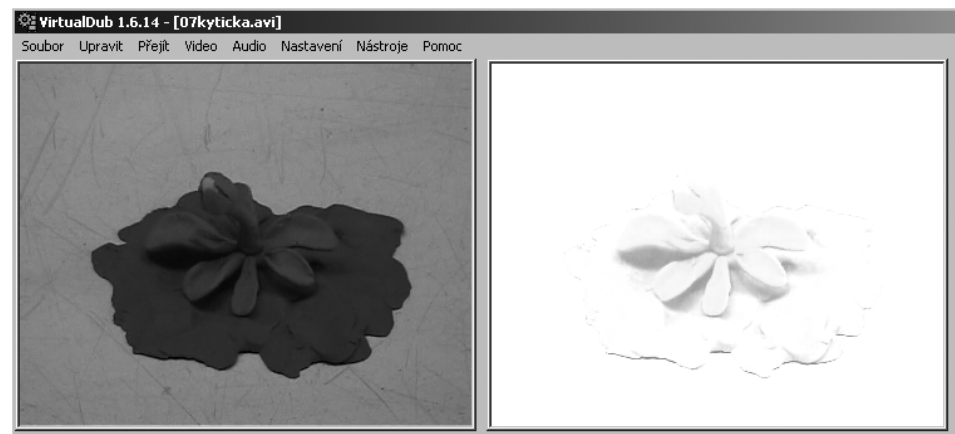
4	blur more	Gaussovské rozostření videa s parametrem radius 2.	obrázek 4.13
---	-----------	--	--------------



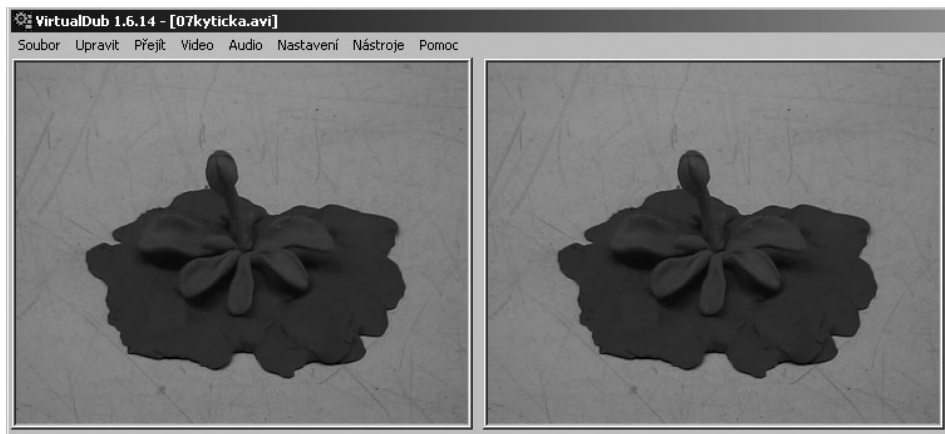
5	box blur	Rychlé aproximativní rozostření videa.	obrázek 4.14
---	----------	--	--------------



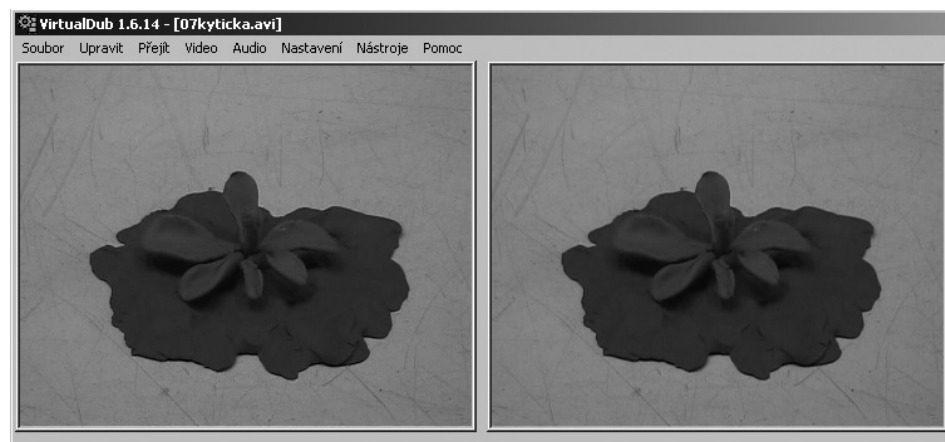
6	brightness/ contrast	Úprava kontrastu a světelnosti videa.	obrázek 4.15
---	-------------------------	---------------------------------------	--------------



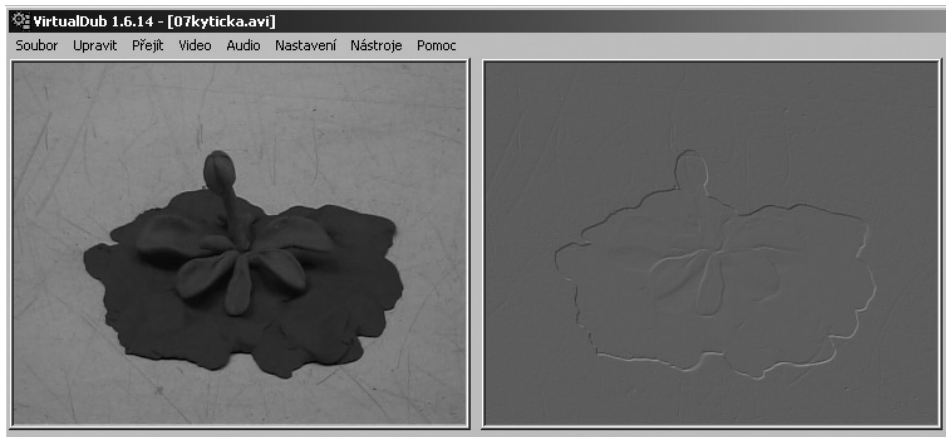
7	chroma smoother	Re-interpolace barevné informace beze změny světlosti.	obrázek 4.16
---	-----------------	--	--------------



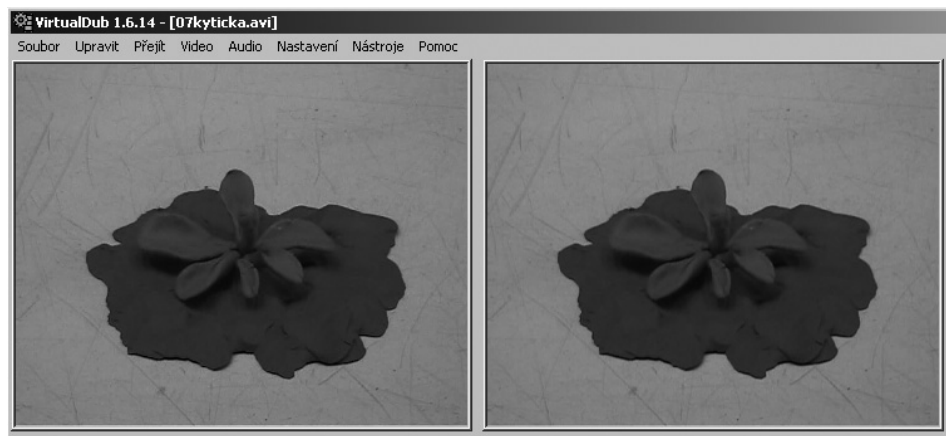
8	deinterlace	Odstranění či sloučení prokládaných snímků videa.	obrázek 4.17
---	-------------	---	--------------



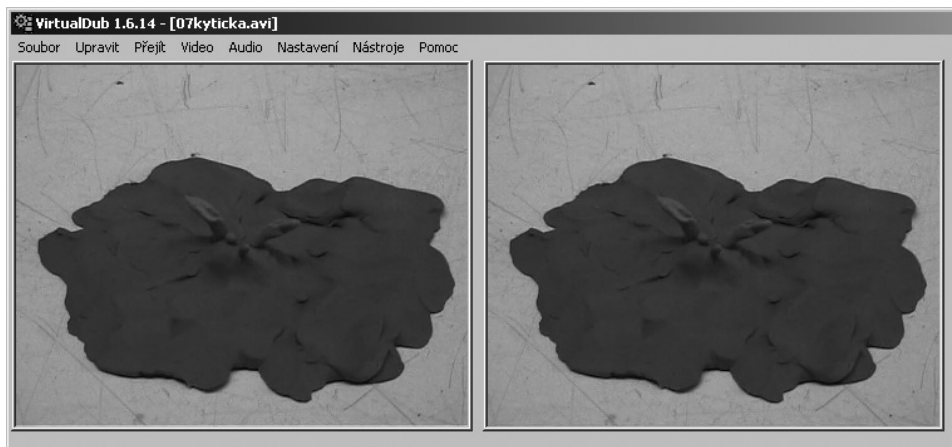
9	emboss	Pseudo 3D matrice detekovaná z okrajů předmětů obrazu.	obrázek 4.18
---	--------	--	--------------



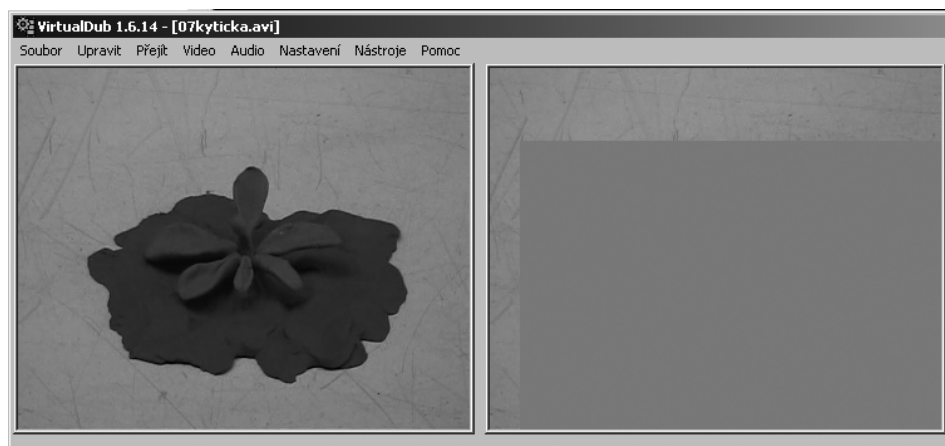
10	field bob	Odstraňuje chvění obrazu vzniklé v souvislosti s prokládáním.	obrázek 4.19
----	-----------	---	--------------



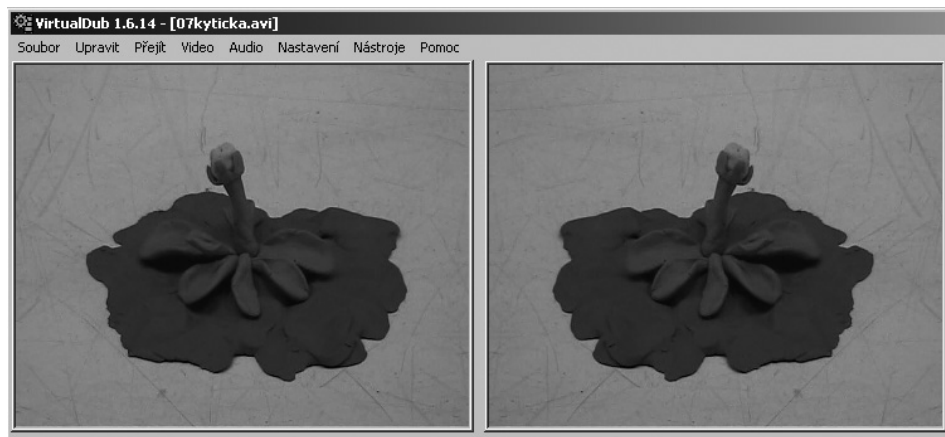
11	field swap	Opravuje zaměněné liché a sudé snímky při změně prokládání.	obrázek 4.20
----	------------	---	--------------



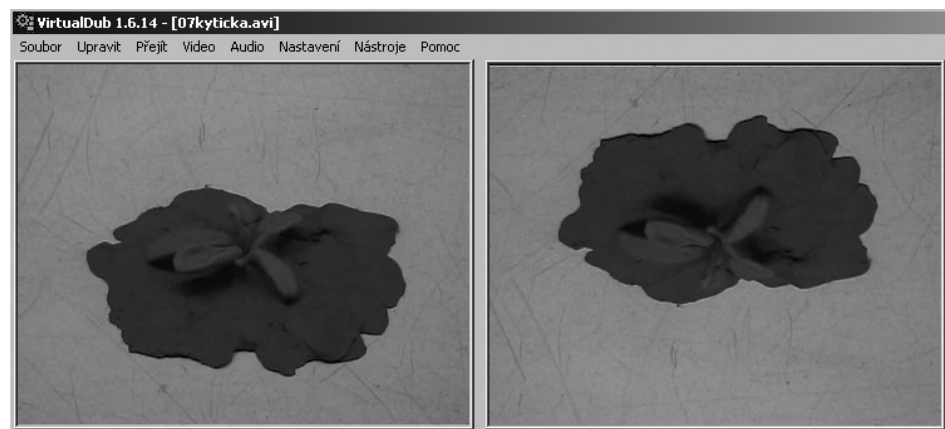
12	fill	Vyplnění určeného obdélníku uvnitř videa spojitou barvou.	obrázek 4.21
----	------	---	--------------



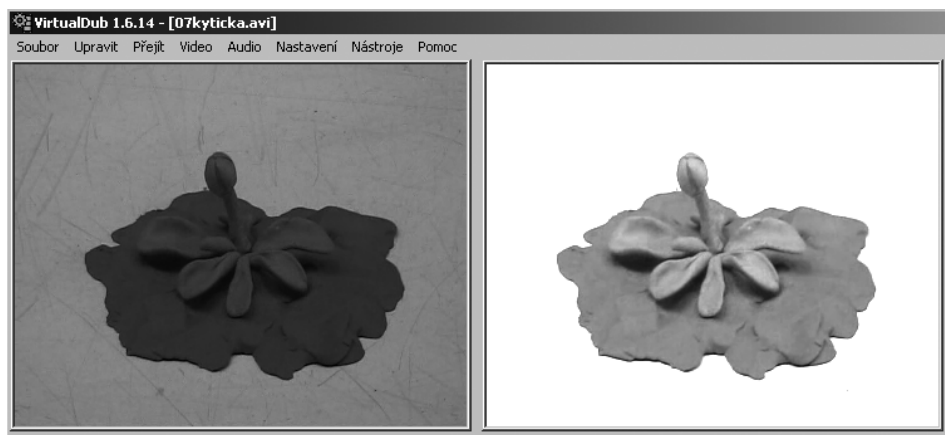
13	flip horizontally	Horizontální otočení videa.	obrázek 4.22
----	----------------------	-----------------------------	--------------



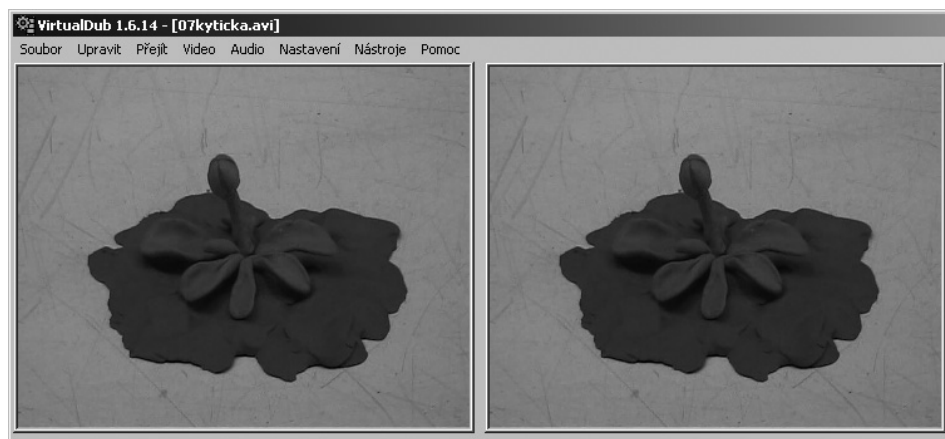
14	flip vertically	Vertikální otočení videa.	obrázek 4.23
----	--------------------	---------------------------	--------------



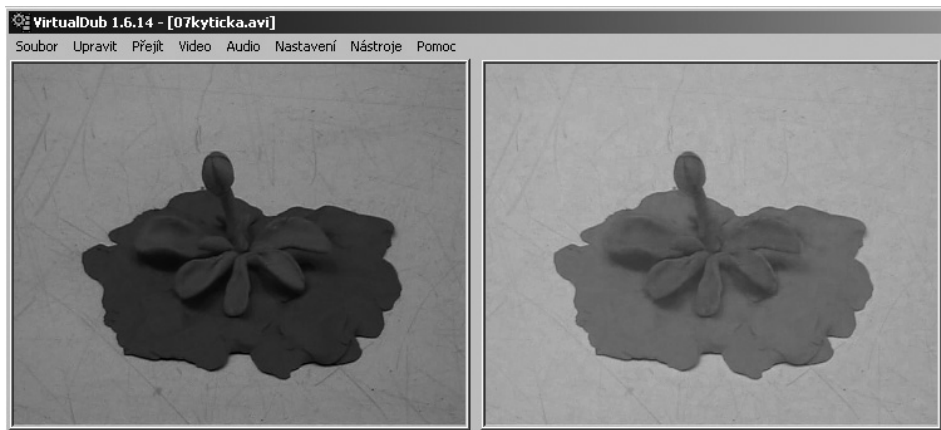
15	general convolution	Aplikace nastavitelné 3 * 3 filtrovací masky.	obrázek 4.24
----	---------------------	---	--------------



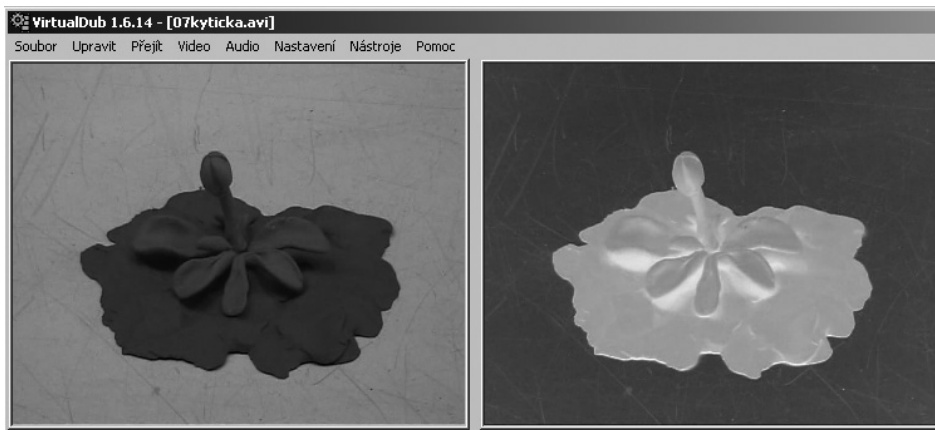
16	grayscale	Zkonvertování barevného videa do stupňů šedi.	obrázek 4.25
----	-----------	---	--------------



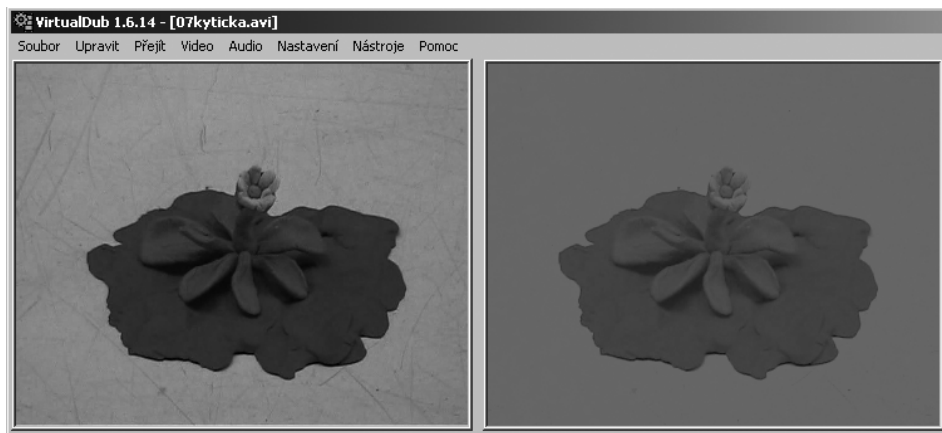
17	HSV adjust	Změna HSV módu videa.	obrázek 4.26
----	------------	-----------------------	--------------



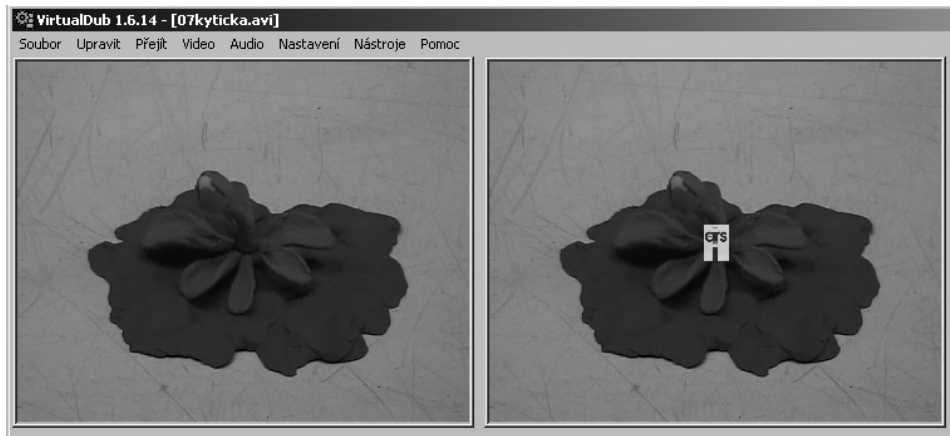
18	invert	Vytvoření barevného negativu videa.	obrázek 4.27
----	--------	-------------------------------------	--------------



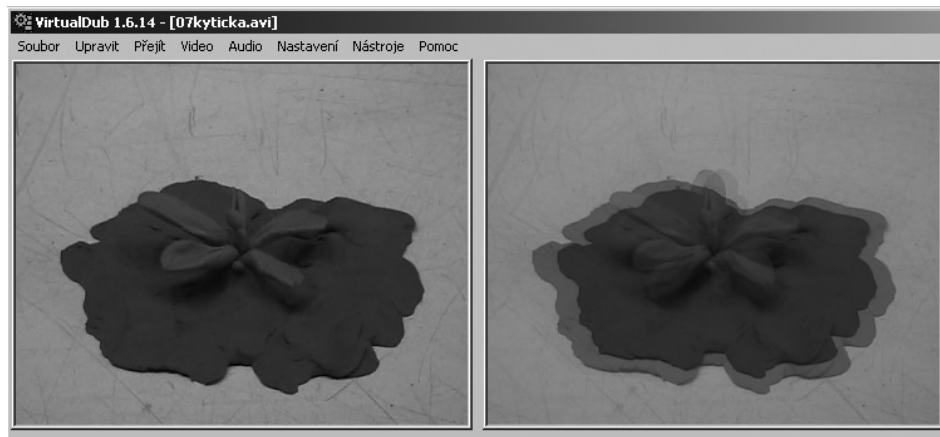
19	levels	Úprava nelineárních úrovní obrazu.	obrázek 4.28
----	--------	------------------------------------	--------------



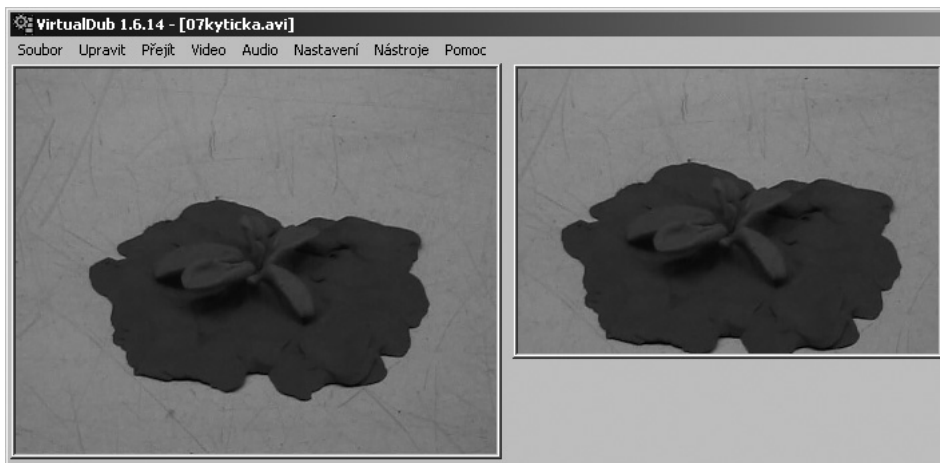
20	logo	Možnost vložit obrázek, nejčastěji logo, do výsledného videa.	obrázek 4.29
----	------	---	--------------



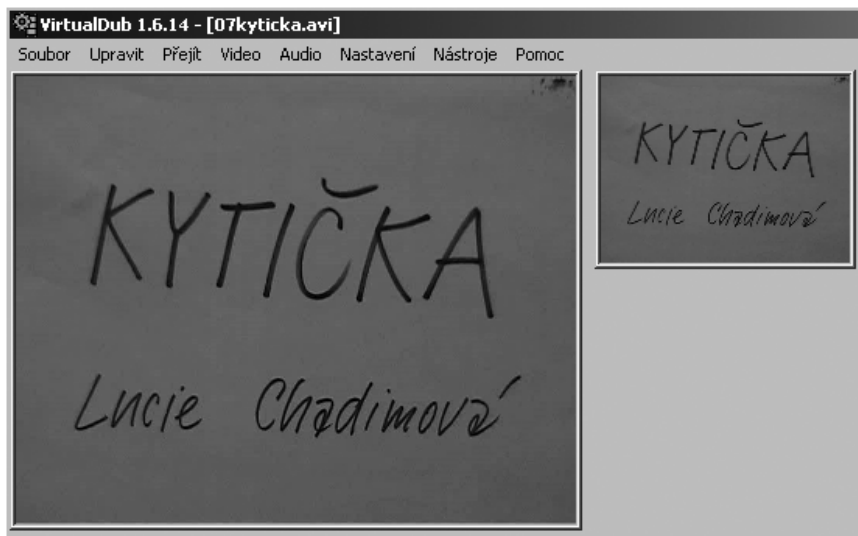
21	motion blur	Pohybové rozostření videa. Zachytává se pohyb mezi jednotlivými obrázky.	obrázek 4.30
----	-------------	--	--------------



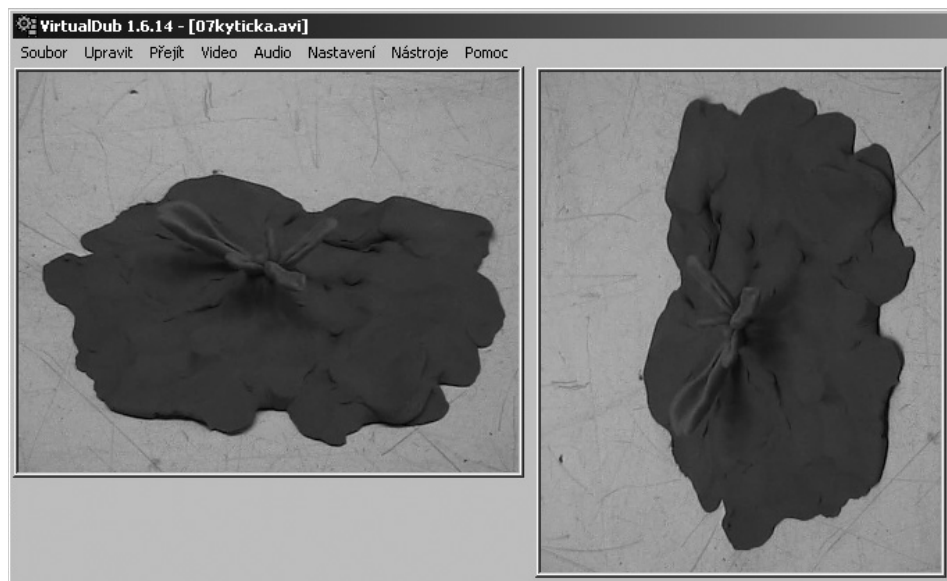
22	null transform	Nevykonává žádnou specifickou funkci, může však být využit k ořezu obrázku.	obrázek 4.31
----	----------------	---	--------------



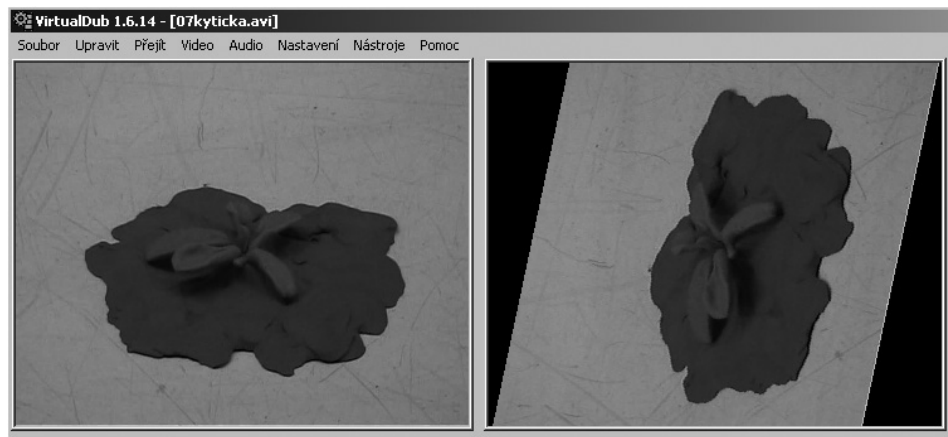
23	resize	Konverze videa do nové velikosti.	obrázek 4.32
----	--------	-----------------------------------	--------------



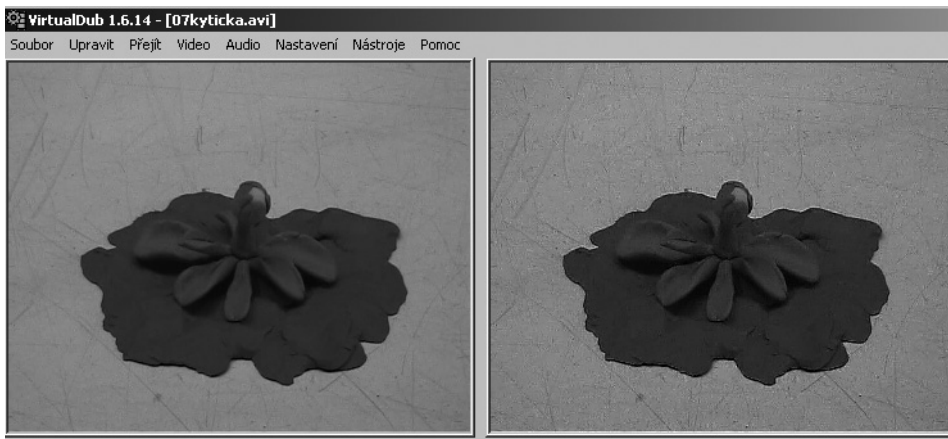
24	rotate	Rotace videa jako celku. Jde tedy o změnu velikosti videa podle nastaveného úhlu.	obrázek 4.33
----	--------	---	--------------



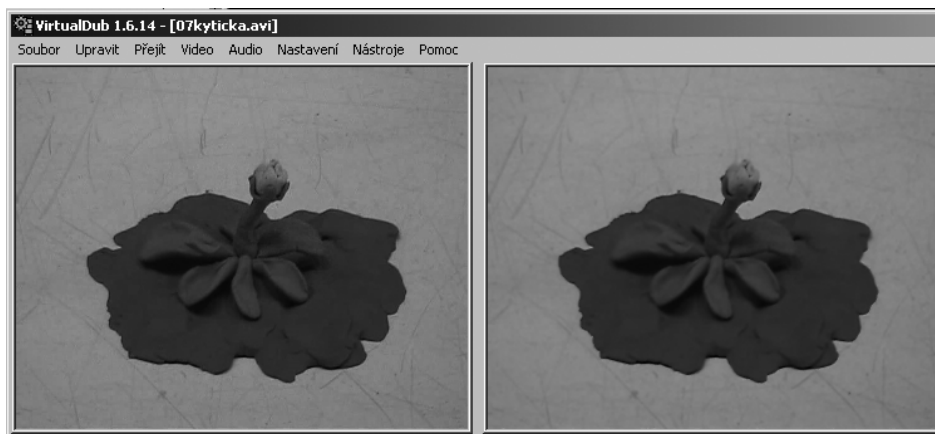
25	rotate2	Rotace obrazu videa. Velikost videa se přitom nemění. Prázdné plochy se vyplní černě.	obrázek 4.34
----	---------	---	--------------



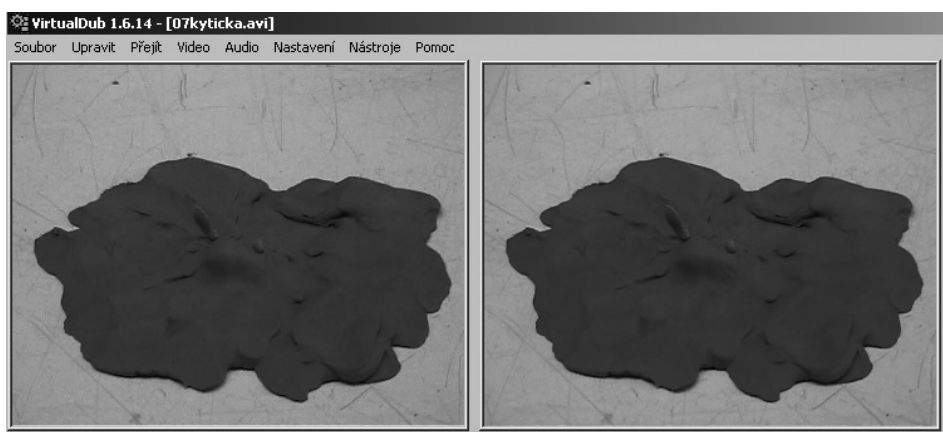
26	sharpen	Zaostření obrazu videa.	obrázek 4.35
----	---------	-------------------------	--------------



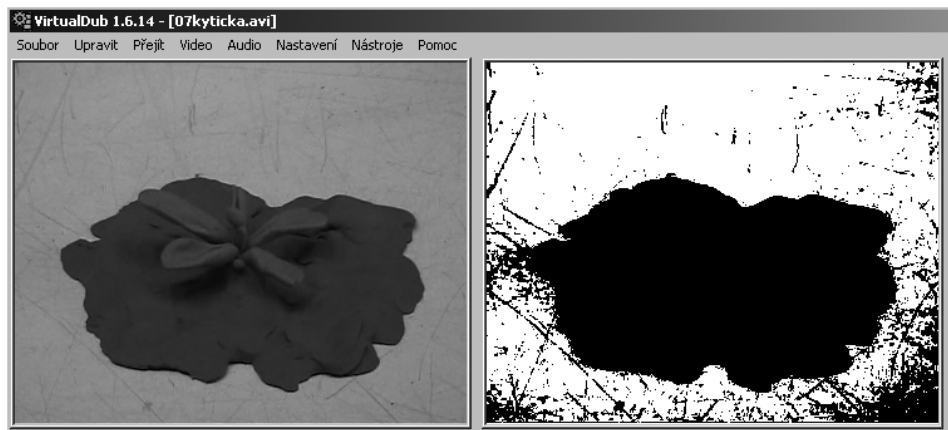
27	smoother	Rozostření, které vyhlazuje hrany předmětů v obrazu videa.	obrázek 4.36
----	----------	--	--------------



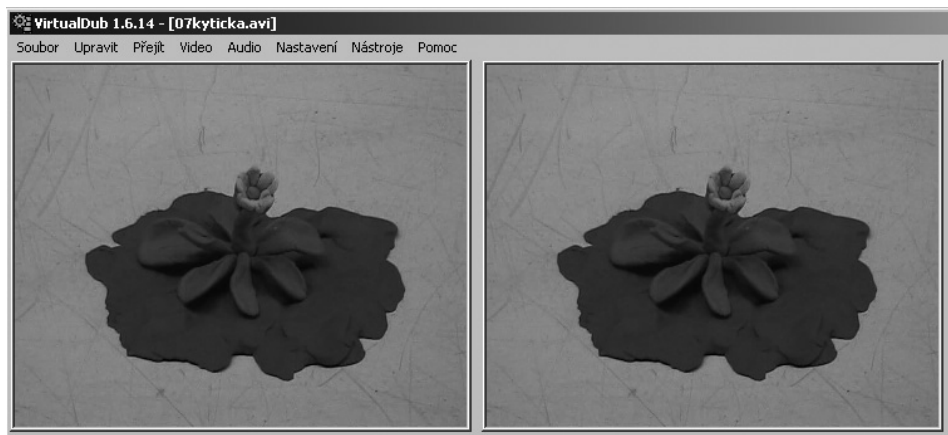
28	temporal smoother	Adaptivní úprava obrazového šumu mezi jednotlivými obrazy videa.	obrázek 4.37
----	-------------------	--	--------------



29	threshold	Konverze do černobílého zobrazení porovnáním prahu jasů.	obrázek 4.38
----	-----------	--	--------------



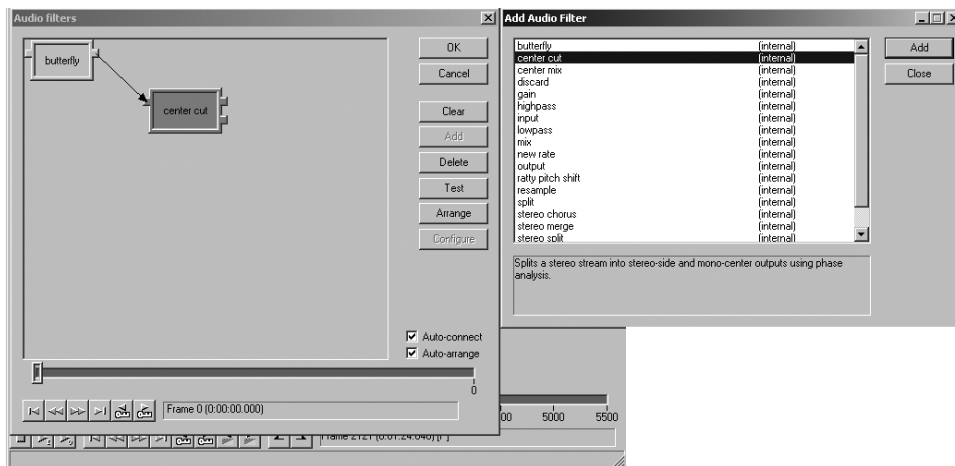
30	TV	Konverze videa do světelně chromatického barevného prostoru obvyklého u TV.	obrázek 4.39
----	----	---	--------------



Aplikaci filtru či filtrů můžeme okamžitě pozorovat na výstupním zobrazení (pravé okno s videem), jak je vidět na výše uvedených obrázcích. Nic nám také nebrání filtry v libovolném množství kombinovat. Kromě těchto vestavěných filtrů pak existuje, opět zdarma, celá řada dalších přídatných.

Co se týče zvuku, nabízí **VirtualDub** také řadu možností. Ty nalezneme v menu **Audio**. Máme možnost konvertovat video beze zvuku nebo konvertovat zvuk, který je ve vstupním videu už obsažen, případně připojit zvuk z externího souboru WAV.

Samozřejmostí je také řada filtrů pro postprodukční zpracování zvuku, jako vidíme na obrázku 4.40. Filtrace zvuku však není implicitní možností programu. Musíme ji tedy nejdříve povolit, což uděláme následovně:



Obrázek 4.40 Okno s audiofiltry v programu VirtualDub

1. V menu **Audio** zvolíme položku **Úplné zpracování**. Implicitně je zatržena možnost **Přímá kopie**, která pouze zkopíruje do výstupního souboru audio, tak jak je ve vstupním videu, případně ve zdrojovém souboru **WAV**.
2. Tím se nám zviditelní některé do té doby nedostupné volby, jak vidíme na obrázku 4.41: **Komprese...**, **Použit rozšířené filtrování**, **Převod...** a **Hlasitost...**
3. Pomocí menu **Compression...** můžeme volit kompresi výstupního videa. Touto volbou se nyní nebudeme zabývat.
4. Menu **Převod...** nám umožní změnit vzorkovací frekvenci výsledného videa.
5. Zvolíme-li **Použit rozšířené filtrování**, zviditelní se nám další volba, **Filtry...**
6. Filtry vkládáme skrze tlačítko **Přidat...**
7. Každý filtr se nám zobrazí graficky, jako obdélník se vstupem a výstupem v podobě malých vstupně-výstupních kolonek.
8. Jakmile klepneme myší na grafické zobrazení vstupu či výstupu, zobrazí se nám šipka, kterou spojíme náležitě všechny filtry za sebou.



Obrázek 4.41 Rozšířené volby audia v programu VirtualDub

9. Jde o analogii zapojování jednotlivých „zvukových skříněk“ za sebou. Proto musí každé takové zapojení začínat okénkem **Input** a končit okénkem **Output**.

Zde je seznam dostupných zvukových filtrů vestavěných přímo ve **VirtualDubu**:

butterfly	Konverze mezi levým/pravým a středním/stranovým zvukem.
center cut	Extrahuje střední kanál ze stereo audia.
center mix	Mixuje mono kanál se stereo zvukem.
discard	Ruší nechtěné zvuky.
gain	Změna hlasitosti audia.
highpass	Ořeže zvuk pod danou frekvencí.
input	Označuje zdrojové audio.
lowpass	Ořeže zvuk nad danou frekvencí.
mix	Mix různých zvuků dohromady.
output	Označuje výstupní audio.
ratty pitch shift	Změna výšky zvuku beze změny jeho rychlosti.
resample	Konverze do jiné samplovací frekvence.
split	Rozdělí audio do dvou identických kopií.
stereo chorus	Přidá efekt chorus (zmnožení zvuku).
stereo merge	Spojí dva mono zvuky do jednoho stereo zvuku.
stereo split	Rozdělí stereo zvuk do dvou mono zvuků.
stretch	Prodloužení nebo zkrácení zvuku, ovlivňuje délku zvuku a jeho výšku.

Pro nás nejdůležitější nastavení výstupu se skrývá pod volbou **Komprese...** v menu **Video**. Zde volíme výstupní formát, tedy kodek, v němž má být cílové video kódováno. Vybereme náležitý **DivX** nebo **Xvid** kodek a tlačítkem **Nastavení** otevřeme konfigurační okno kodeku, které již bylo popsáno.

Volbou **Barevná hloubka** z menu **Video** nastavíme druh barevného prostoru pro výstup. Pokud nemáme v našem videu alpha kanál, nejjistější volbou bude **24 bit RGB**.

Na závěr se ještě zastavíme u možnosti použít **VirtualDub** k načtení videa z externího zdroje. Jak jsme již řekli, nazývá se tato činnost anglicky **Capture** a najdeme ji v menu **Soubor** ve volbě **Zachytit do AVI...**; **VirtualDub** se nám přepne do módu zachytávání (**capture mode**) a objeví se zcela nové menu.

1. V menu **Soubor** vybereme volbou **Nastavit soubor...** soubor, do něhož se výsledné video zachytí.
2. V menu **Zařízení** najdeme všechna dostupná zařízení. Připojíme-li tedy kameru pomocí **FireWire** vstupu k počítači, měla by se nám zde tato možnost objevit.
3. V menu **Video** máme pomocí volby **Komprese...** možnost nahrát zachytávané video přímo komprimované.
4. Případně ve stejném menu můžeme na vstupní video aplikovat již popsané filtry, skrze volbu **Filtry...**
5. V menu **Audio** povolíme či zrušíme zachytávání zvuku spolu s videem skrze volbu **Zapnout zachytávání audia**. Tato volba je implicitně zapnutá.
6. Samotné zachytávání spustíme v menu **Zachytávání** volbou **Zachytit video**.
7. Ve stejném menu pak zachytávání ukončíme volbou **Zastavit zachytávání**.

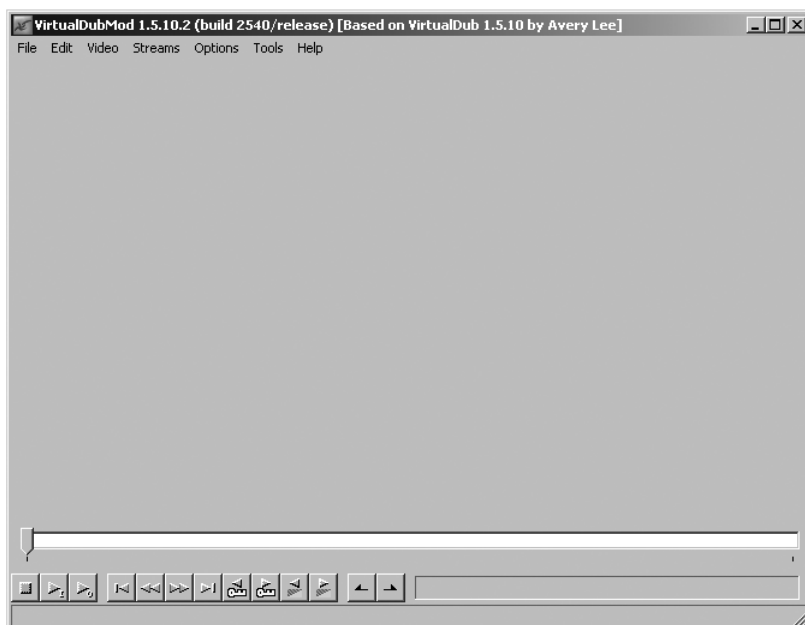
VirtualDubMod – rozšířená verze programu VirtualDub a její možnosti

Program **VirtualDub** se kromě toho, že existuje v mnoha verzích a stále se vyvíjí, také stal podkladem pro celou rodinu programů, které z něj vycházejí a doplňují jej pro jiné účely použití. O jedné takové verzi, která se jmenuje **VirtualDubMod**, si povíme v následujících řádcích.

Významnými doplňky programu **VirtualDubMod** jsou možnost používat i jiné než **AVI** kontejnery (například kontejnery **Matroska** a **OGG**) a možnost pracovat s více audiostopami.

POZNÁMKA: Kontejner **OGG** je postaven na několika kodecích pro různé účely. Především na ztrátovém zvukovém kodeku **Ogg Vorbis**, na videokodeku **Theora**, zvukovém kodeku pro mluvenou řeč **Speex**. Dále využívá bezztrátový kodek pro přenos zvuku **Flac** a kodek **Writ** pro vkládání titulků do filmu. Podobně jako u kontejneru **Matroska** je nejlepším softwarem na přehrávání tohoto kontejneru **Media Player Classic**.

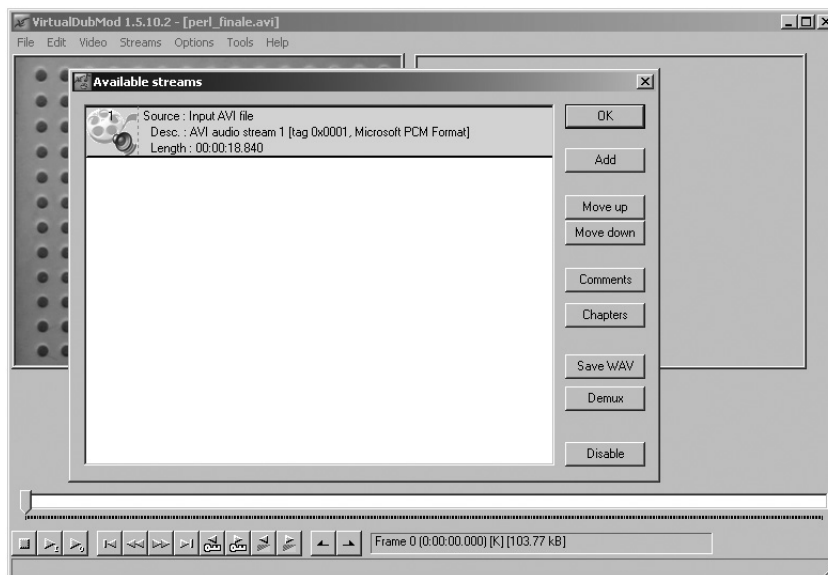
Nejdůležitější změnou oproti programu **VirtualDub** je menu **Streams** v místě, kde je v programu **VirtualDub** menu **Audio**, jak vidíte na obrázku 4.42.



Obrázek 4.42 Menu Streams v programu VirtualDubMod

V tomto menu máme možnost pracovat se zvukovými stopami, oddělit je od videa, sloučit více najednou a tak dále. Postup je následující:

1. Otevřeme zdrojové video: v menu **File** volbou **Open Video File**.
2. V menu **Streams** zadáme volbu **Streams List**.
3. Objeví se nám okno **Available Streams** jako na obrázku 4.43.
4. Zde vidíme, jaké audiostopy jsou ve videu obsaženy, v jakém formátu a kvalitě.



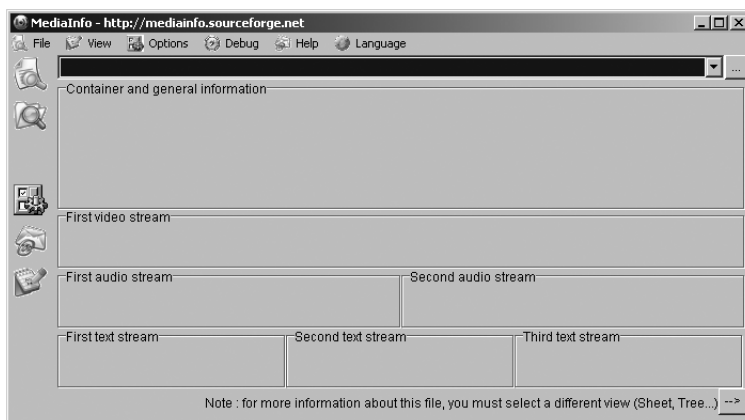
Obrázek 4.43 Okno s dostupnými zvukovými stopami v programu VirtualDubMod

5. Pomocí tlačítka **Add** přidáme další požadované audiostopy.
6. Ty potom můžeme vymazat tlačítkem **Delete**.
7. Pomocí volby **Save WAV** můžeme oddělit audiostopu pevně vloženou do videa, editovat ji v nějakém externím nástroji (například převést z formátu **WAV** do formátu **MP3**) a následně pomocí **Add** ji znovu vložit do výsledku.
8. Jinak pracujeme s videem stejným způsobem jako v programu **VirtualDub**.

Media Info – informace o druhu kontejneru a použitých kodecích

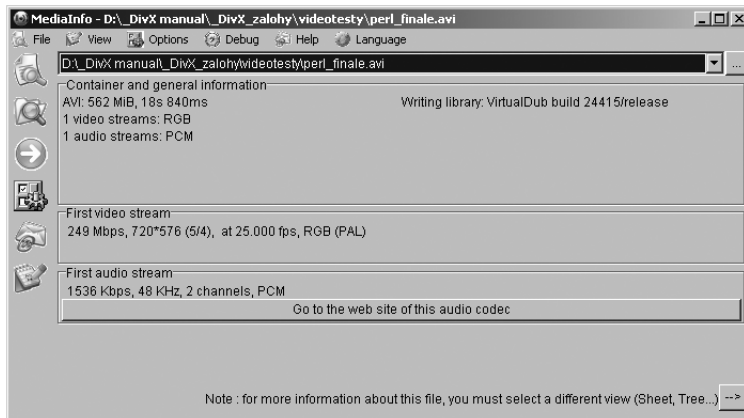
Pro zjištění, jaký druh kontejneru obsahuje daný soubor, což v první řadě poznáme podle souborové koncovky, existují také programy, které nám podají podrobné informace.

Jedním z nich je program **MediaInfo**, jehož základní okno vidíme na obrázku 4.44. Program nám po otevření souboru pomocí tlačítka **Select a multimedia file to study** otevře zkoumaný soubor a zobrazí všechny důležité informace.



Obrázek 4.44 Základní okno programu MedialInfo

Především nás informuje o použitém kontejneru (na obrázku 4.45 vidíme informace o kontejneru a jeho součástech). Snadno se takto dozvíme, obsahuje-li kontejner jednu či více audiostop, vložené titulky atd.



Obrázek 4.45 Zobrazení informací o kontejneru v programu MedialInfo

Kromě toho se dozvíme mnohé i o formátu jednotlivých součástí, tedy o kodecích užitých pro video i audio, a o způsobu kódování připojených titulků. Program nám také umožňuje podívat se na web, který se k danému kodeku váže.

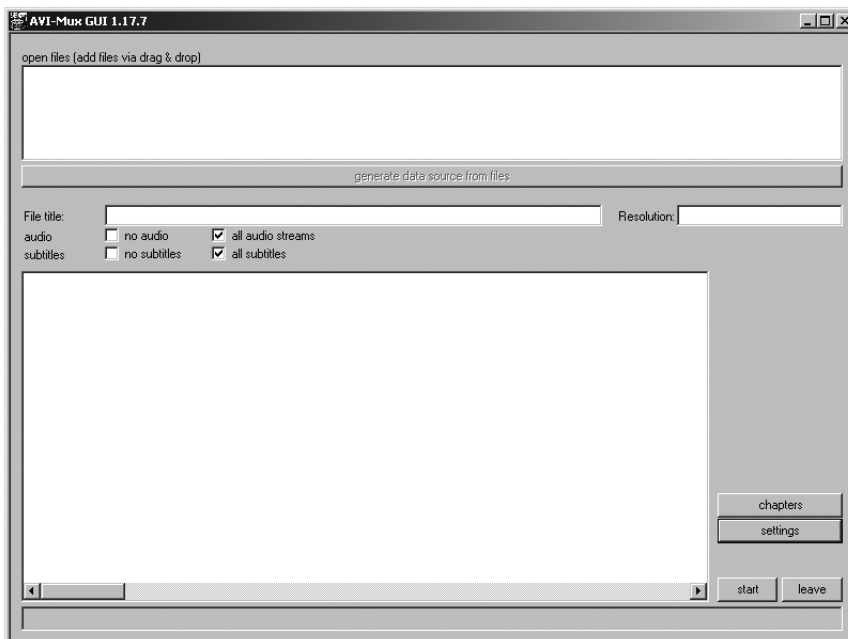
POZNÁMKA: Zajímavá vlastnost programu se otevře tlačítkem **Export in a customized format**. Máme zde možnost exportovat informace o videu do různých formátů, od tabulek CSV přes prostý text až po HTML. Díky tomu můžeme u složitějších videí, které někomu předáváme nebo publikujeme na Internetu, přibalit či nechat jednoduše zobrazit informace o použitých kodecích, kontejneru atd.

Avi Mux Gui – software na tvorbu AVI a Matroska kontejnerů

Kromě možnosti jednotlivé kontejnery přehrávat a číst o nich náležitě informace je také nutné si říci, jak kontejnery vytvářet. Prakticky celá tato kniha se většinou věnuje kontejneru AVI, v němž máme uloženo video zpracované kodekem DivX nebo Xvid.

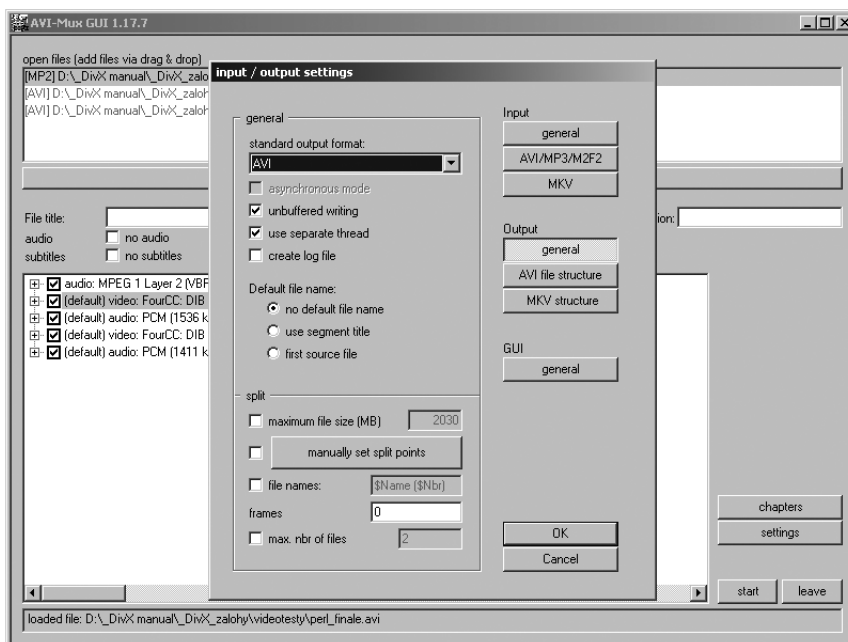
Protože jsme se zmínili také o kontejneru **Matroska**, řekneme si něco o softwaru **AVI Mux GUI**, pomocí něhož můžeme takový kontejner vytvářet. Tento software se neinstaluje, ale přesuneme celou jeho složku do složky **Program Files** nebo jinam, dle uvážení.

Základní okno programu je znázorněno na obrázku 4.46. Práce s ním je velmi jednoduchá. Metodou táhni&pusť přesuneme všechny kýžené součásti kontejneru do základního okna. Následně pomocí tlačítka **Generate Data Source from Files** zjistíme, co je obsahem vložených souborů, a můžeme vytvořit kýžený výstup.



Obrázek 4.46 Základní okno programu AviMuxGUI

Tlačítkem **Settings** otevřeme okno pro nastavení požadovaného kontejneru výstupu, jak vidíme na obrázku 4.47. Máme dvě možnosti kontejneru, buď AVI nebo MKV, což je koncovka pro **Matroska** kontejner. V poli **Split** máme možnost rozdělit výstupní soubor na více částí.



Obrázek 4.47 Nastavení výstupního kontejneru v programu AviMuxGUI

Pomocí tlačítek **AVI file structure** a **MKV file structure** v poli **Output** můžeme pak nastavit detaily struktury výstupního kontejneru. Doporučuji nechat zde nastaveny implicitní hodnoty, pokud nejsme programátoři znalí přesného zápisu hlaviček a vnitřní struktury těchto kontejnerů. Protože pro běžnou práci není taková znalost nutná a kontejner **Matroska** je prozatím stále spíše v rozběhu, budou nám stačit zde uvedené informace.

POZNÁMKA: Kontejner **Matroska** je nově se rozvíjející kontejner. Díky své struktuře, postavené na XML, je velmi univerzální. Podporuje titulky, menu, kapitoly filmu, jednotlivé titulky a lze do něj přidávat i externí soubory. Bohužel je zatím problematická podpora tohoto formátu i způsob jeho vytváření. Pro videosoubory tento kontejner používá příponu **.MKV**, pro audiosoubory **.MKA** a pro samostatné titulky příponu **.MKS**.

Tlačítkem **Start** pak spustíme konverzi. Program je možné použít také k jednoduchému převodu mezi těmito dvěma kontejnery.

Dejme tomu, že máme kontejner **MKV**, který neumíme přehrát. Potom můžeme jednoduše tento vstupní kontejner převést do používanějšího **AVI**, který přehrajeme většinou běžných a zde popisovaných přehrávačů.

O postprodukčním nástroji AviSynth

AviSynth je velmi dobrým nástrojem na postprodukční úpravy videa, převody mezi formáty, aplikaci nejrůznějších kvalitních filtrů atd. Je využíván některými již popsanými komplexními převodními programy (či systémy).

Tento program pracuje velmi netradičně jako takzvaný **Frameserver**. Výstupní video je nekomprimovaně předáváno dalším programům k zobrazení, **AviSynth** provede postupné úpravy při přehrávání a další editaci. Proces funguje tak, že je zobrazovacímu softwaru předáván každý snímek zpracovaný pomocí **AviSynth** postupně. Nevytváří se tedy žádný soubor mezi dvěma aplikacemi, **AviSynth** funguje na pozadí při samotném přehrávání videa.

Jednotlivé kroky programu **AviSynth** se provádějí pomocí skriptu, tedy textového zápisu formou příkazů. Tento fakt je na užití programu nejvíce odrazující, protože výstup nevidíme ihned, a je zde tedy výrazně omezena možnost intuitivního užívání.

Přesto je dobré o jeho možnostech alespoň vědět, zvláště proto, že jak program samotný, tak řada dostupných přídatných filtrů se distribuuje zdarma. K **AviSynth** také existuje několik skriptových editorů, které nám práci s ním poněkud ulehčí, především díky seznamu dostupných příkazů.

POZNÁMKA: Nejpoužívanější editor skriptů pro **AviSynth** je **AVSEdit**. Nabízí řadu zajímavých a doplňkových funkcí a především umožňuje přímý náhled výsledku skriptu. Budeme se jím zabývat dále v textu. Další, podle mého nepříliš povedenou aplikací je **AVSGenie**, kde zadáváme příkazy **AviSynth** pomocí formuláře a menu. Zajímavá je aplikace **AVE Visual Editor**, která řeší zadávání příkazů **AviSynth** jako vázané bloky modulů v grafickém prostředí. Bohužel není tato aplikace příliš spolehlivá a její vývoj se před několika lety zastavil.

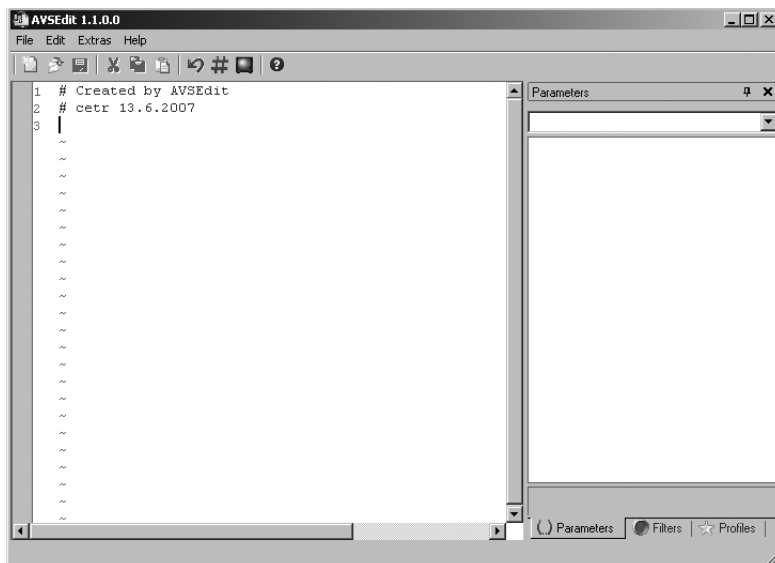
Na jednoduchém skriptu si ukážeme, jak takový proces funguje.

```
AVISource("g:\DivX_manual\videotesty\perl_finale.avi")
Trim(0, 100) ++ Trim(150, 200) ++ Trim(250, 300)
Crop(80,250,250,250)
FadeIn(10)
FadeOut(10)
```

Nejprve musíme zvolit videozdroj, což jsme učinili příkazem **AVISource**. Následně provedeme stříh tří kusů zdrojového videa pomocí příkazů **Trim**. Následující řádek video ořízne podle zadaných parametrů příkazem **Crop**. Následující dvě instrukce, **FadeIn** a **FadeOut**, přidají na začátek a konec videa roztmívačku a stmívačku (tj. postupné objevování se, respektive mizení obrazu, z nebo do černého podkladu).

Jak vidíme, touto sadou instrukcí jsme provedli klasické operace videoeditačních programů. Nevýhodou zde samozřejmě je již zmíněné neintuitivní ovládání.

Na tvorbu skriptů pro **AviSynth** doporučuji editor **AVSEdit**, který ve verzi 1.1.0.0 najdeme na příloženém CD. Základní okno tohoto programu vidíme na obrázku 4.48.



Obrázek 4.48 Základní okno skriptového editoru pro AviSynth AVSEdit

V tomto editoru, kromě toho, že obsahuje popis a možnost formulářového vkládání parametrů jednotlivých instrukcí, si můžeme v interním prohlížeči prohlédnout rychle výsledek. Také si můžeme nastavit externí prohlížeč (schopný přehrávat k videu i zvuk) v menu **Option** volbou **External Preview**. K externímu prohlížení je doporučeno zvolit program **Media Player Classic**.

Podrobný popis příkazů a jednotlivých filtrů najdeme v helpu programu **AVSEdit**, případně je podrobná dokumentace v angličtině dostupná na adrese: <http://www.avisynth.org/>.

Pro naše účely stačí tento jednoduchý úvod, který spíše ukazuje na řadu možností tohoto programu. Pokročilé skriptování je věcí samostatné publikace, která by se **AViSynth** možnostmi zabývala.

Jen ještě jedna poznámka, program využívá řady vestavěných příkazů a funkcí. V případě těch přidavných musíme na začátku skriptu zadat jejich nahrání pomocí instrukce, jako v tomto ukázkovém případě:

```
LoadPlugin("c:\Program Files\Avisynth\plugins\VagueDenoiser.dll")
```


S programem **AviSynth** je možné zpracovat řadu jinak těžko zdarma dostupných úkolů v úpravách a editaci videa. Bohužel právě jeho netradiční skriptovací rozhraní je pro řadu zájemců spíše odrazující a je nutné se programu věnovat delší čas, aby práce v něm byla smysluplná a efektivní.

Nyní již ale budeme vědět, co znamená, když nám například komplexní program **AVI2DVD** zapíše při převodu na disk soubor **AviSynth_Script_File.avs**. **AVI2DVD** je jedním z programů, které **frameserver AviSynth** využívají ke svým úkolům, a není zdaleka jediným. Pro některé fáze převodu mezi barevnými módy atd. jsou funkce **AviSynth** totiž nezbytné.

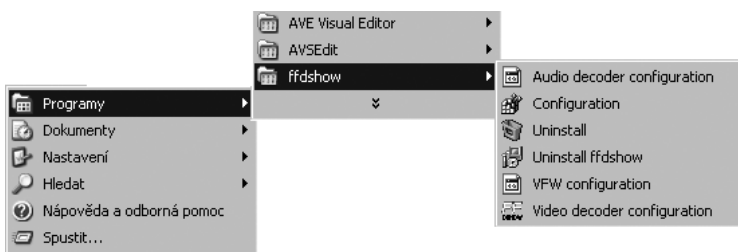
Sada filtrů pro zobrazení DivX a Xvid – FFDSHOW

Sada filtrů a dekodovacích kodeků **FFDshow** je kompletním řešením pro zobrazení DivX a Xvid videa, pro různé možnosti výstupních zobrazovacích úprav a pro zobrazení titulků. Spolupracuje přímo s přehrávačem **Media Player Classic** a dovoluje výrazně zvýšit kvalitu zobrazení u videí s menším datovým tokem nebo nějak vizuálně poškozených.

Program **FFDShow** obsahuje velké množství kodeků postavených na bázi takzvaných **libavcodec**. Zároveň umožňuje zobrazit titulky a spolupracuje s programy **AviSynth** a **VobSub**.

POZNÁMKA: Program *FFDshow* a například také *VobSub* nejsou aplikace toho typu, na které jsme většinou zvyklí. Jejich práce neprobíhá tak, že bychom přímo v nich něco otevírali a zpracovávali s jasným vstupem a výstupem. Oba programy se spouštějí na pozadí otevírání videa v jiných aplikacích, které slouží jako přehrávače. V průběhu přehrávání aplikace *FFDshow* a podobné něco přidávají, jako například titulky, nebo upravují výstup podle různých parametrů nastavení.

Co nás na tomto programu, nebo spíš sadě kodeků zajímá, jsou dvě konfigurační utility: **Audio decoder configuration** a **Video decoder configuration**, které najdeme mezi nainstalovanými programy ve Start menu, jak dokládá obrázek 4.49. Už z jejich názvu vyplývá, že sada **FFDShow** je určena pro zobrazení a přehrávání množství formátů a kodeků a není určena pro kódování; zde musíme pracovat s kodeky DivX a Xvid, jak již bylo popsáno.

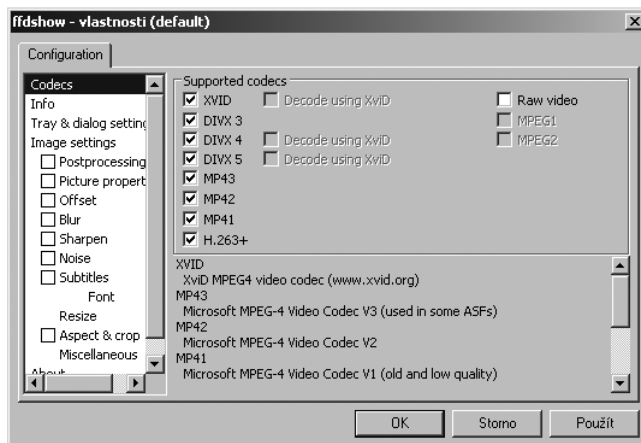


Obrázek 4.49 Nainstalované konfigurační utility sady FFDShow

Přesto se zde nabízí řada zajímavých funkcí, můžeme totiž zcela pozměnit audio či videovýstup, upravit šумы, změnit velikost videa, zapnout různé další zobrazovací filtry. A především nám s touto sadou odpadá nutnost instalace řady kodeků, které jsou v **FFDShow** obsaženy.

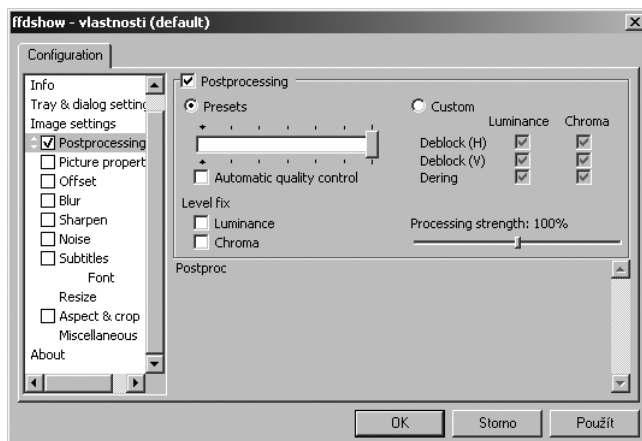
Ve stručnosti si popíšeme možnosti nastavení videodekodéru, které najdeme v nabídce nainstalovaných programů ve složce **FFDShow** pod označením **Video decoder configuration**:

1. V první záložce, **Codecs** (obrázek 4.50), najdeme seznam formátů a k nim příslušných dekodérů.



Obrázek 4.50 Záložka Codecs v nastavení sady filtrů FFDShow

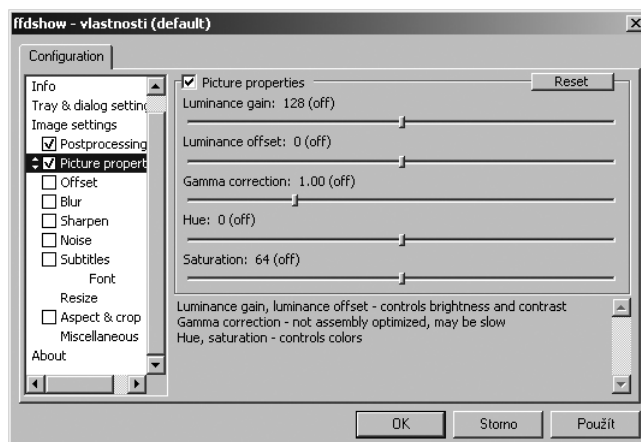
2. Všechny další nabídky volíme formou zatržítka v levém panelu. Implicitně jsou vypnuty, protože nabízejí úpravy obrazu, které budeme využívat, jen když to bude nutné.
3. Volbou **Postprocessing**, viz obrázek 4.51, spustíme automatickou opravu vadných segmentů přehrávaného videa.



Obrázek 4.51 Záložka Postprocessing v nastavení sady filtrů FFDSHOW

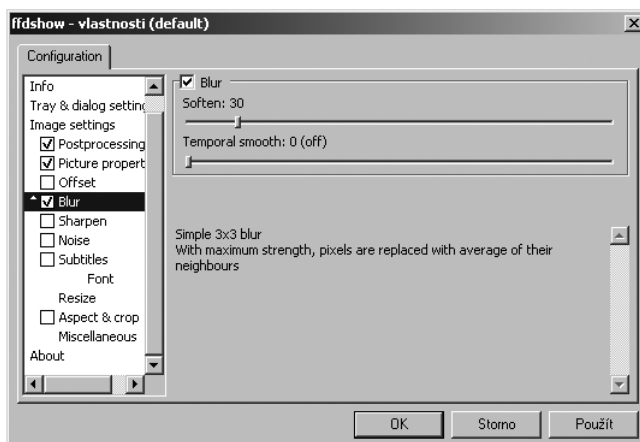
POZNÁMKA: Jak již bylo řečeno, program FFDSHOW se spouští na pozadí jiné aplikace. Ovlivňuje pouze aktuální přehrávání videa, nijak nemění zdrojový soubor!

4. Volbou **Picture properties** (obrázek 4.52) můžeme změnit jas, barevnost a další barevné parametry obrazu.



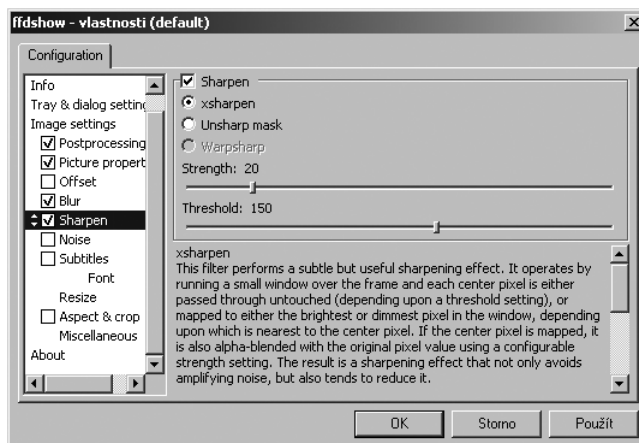
Obrázek 4.52 Záložka Picture properties v nastavení sady filtrů FFDSHOW

5. Volbou **Blur** (obrázek 4.53) můžeme odstranit šumy obrazu a obraz rozostřit.



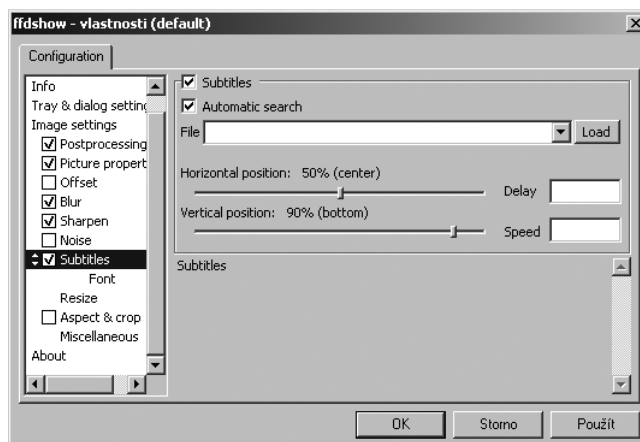
Obrázek 4.53 Záložka Blur v nastavení sady filtrů FFdShow

6. Naopak nabídkou **Sharpen** (obrázek 4.54) obraz doostříme.



Obrázek 4.54 Záložka Sharpen v nastavení sady filtrů FFdShow

7. Pomocí volby **Subtitles** (obrázek 4.55) vybereme soubor s titulky, který budeme spolu s filmem přehrávat. Zároveň zde můžeme nastavit pozici titulků vzhledem k obratu filmu.
8. Nabízených funkcí je více, pro základní orientaci a běžné možnosti úprav však tyto postačí.
9. Provedené nastavení potvrdíme tlačítkem **OK**. Projeví se až při následujícím spuštění videa.



Obrázek 4.55 Záložka Subtitles v nastavení sady filtrů FFDSHOW

DivXRepair – utilita vyhledávající zamrzlý obraz či jeho rozpad do barevných polí

Program **DivXRepair** nám velmi jednoduše umožňuje projít videosoubor a nechat si opravit vadné snímky. Tedy ty, u nichž se projevuje takzvané zamrzání, tedy setrvávání obrazu z předchozích snímků do následujících, nebo u nichž se vlivem velké komprese projevuje rozpad obrazu do 8 * 8 obrazových bodů velkých barevných polí, které nám pak z videa vytváří cosi na způsob nechtěného psychedelického rozpadu obrazu.

V zásadě jde o stejnou funkci, kterou najdeme v programu **FFDSHOW** pod názvem **Postprocessing**.

DivXRepair, jehož okno vidíte na obrázku 4.56, pracuje velmi jednoduše tak, že projde video a tyto závady odstraní, případně nás upozorní, že video je v pořádku.



Obrázek 4.56 Okno programu DivXRepair

1. Video ke zpracování přidáme tlačítkem **Add file...**
2. Zpracování spustíme tlačítkem **Repair files...** Je možné souborů ke zpracování vložit více, všechny vybrat a úpravu spustit dávkově.
3. Na závěr se nám zobrazí textový soubor s informací o výsledku úpravy.

POZNÁMKA: Je dobré si uvědomit, že na rozdíl od **FFDShow** je tento program normální aplikace. To znamená otevřeme v ní zdrojový soubor, provedeme v něm úpravy a uložíme jej. Provedené změny se tak projeví v samotném zdrojovém videu, nejen při jeho přehrávání.