

## ANALÝZA VIDEOZÁZNAMŮ ZA ÚČELY MĚŘENÍ PARAMETRŮ SLEDOVANÝCH OBJEKTŮ

Lačezar LIČEV<sup>1</sup>, Petr ZAJÍC<sup>2</sup>

<sup>1</sup>Katedra informatiky, FEI VŠB TU Ostrava, tř. 17. listopadu 15, 708 33, Ostrava, Česká republika

*lacezar.licev@vsb.cz*

<sup>2</sup>Katedra informatiky, FEI VŠB TU Ostrava, tř. 17. listopadu 15, 708 33, Ostrava, Česká republika

### Abstrakt

Cílem této práce je vývoj modulu, který bude uživateli sloužit jednak jako jednoduchý přehrávač AVI souboru a navíc bude umět na základě událostí zachycených na EKG grafu uložit snímky, kterých se událost týká. Jelikož je EKG graf k dispozici pouze na videu, bude nutné provést analýzu tohoto videa a graf na něm správně rozpoznat. Vytvořený modul je začleněn do systému FOTOM 2009 a spolu s dalšími současně vyvíjenými moduly tak vytvořit novou verzi systému FOTOM<sup>NG</sup>.

**Klíčová slova:** analýza obrazu; Java, fotogrammetrie; FOTOM 2008; FOTOM<sup>NG</sup>; NetBeans Platform; prahování; video; EKG.

### Abstract

The goal of this thesis is development of a Fotom module which is able to play video files and capture frames based on events. Since those events are related to the ECG shown on the video, some analysis will be needed to recognize the ECG. This module plugged together with two other simultaneously developed modules in the FOTOM 2009 will make a brand new version of this application – FOTOM<sup>NG</sup>.

**Keywords:** image analysis; Java; photogrammetry; FOTOM 2008; FOTOM<sup>NG</sup>; NetBeans Platform; thresholding; video; ECG.

## 1 ÚVOD

Cílem této práce je vývoj modulu pro zpracování videozáznamů těchto ultrazvukových vyšetření. Modul bude poté integrován do systému FOTOM 2009, což je aktuální verze programů rodiny FOTOM, a spolu s dalšími současně vyvíjenými moduly vytvoří novou verzi aplikace – FOTOM<sup>NG</sup>.

Příspěvek pojednává o analýze medicínských snímků a čtenář se dozví, jaké důvody vedly k vývoji tohoto modulu. V této části je také uvedeno, i jak by se dal tento modul využít i v jiných odvětvích kromě lékařství.

V další části jsou popsány použité softwarové prostředky, např. o platformě NetBeans Platform a o vývojovém prostředí NetBeans IDE a také o přednostech a možnostech rozšíření Java Advanced Imaging. Dále jsou rovněž nastíněny důvody, které vedly k výběru knihoven Xuggler. Ty slouží ke kódování a dekodování videa, s nímž se zabývám ve svém modulu.

Na závěr jsou zhodnoceny dosažené výsledky a ověřena funkčnost vytvořeného modulu spolu s ostatním novým modulem na 2D modelování objektů v rámci systému FOTOM<sup>NG</sup>.

## 2 ANALÝZA MEDICÍNSKÝCH SNÍMKŮ ZA POUŽITÍ FOTOM<sup>NG</sup>

### 2.1 FOTOM<sup>NG</sup>

Již od roku 2001 je na katedře informatiky FEI VŠB-TUO vyvíjen systém FOTOM, který slouží k digitálnímu zpracování snímků [4]. Původně se tento systém používal pro zpracování důlních snímků a měření důlních jam, ale časem se z něj stal výkonný komplexní systém, který umožnil detekovat různé zájmové objekty a vizualizovat měření. Jelikož byl původní program vytvořen jako jednoúčelový systém, nepočítalo se s žádnou

možností jeho rozšíření a tak byl vývoj každého dalšího modulu stále složitější a díky absenci dokumentace se z programu stávala velká „koule“ nesrozumitelného kódu.

Proto vznikl v roce 2009 úplně nový systém FOTOM 2009, jehož jádro bylo speciálně navrženo tak, aby podporovalo jednoduchý vývoj a integraci nových modulů. Společné API aplikace umožňuje komunikaci a spolupráci jednotlivých modulů, takže odpadá nutnost psát stovky řádků redundantního (nadbytečného) kódu. FOTOM 2009 byl napsán v programovacím jazyce Java a postaven na NetBeans platformě.

V současnosti probíhá vývoj nových modulů, které spolu s jádrem vytvoří zcela novou verzi aplikace FOTOM<sup>NG</sup> rozšířenou o možnosti 2D modelování, definici zájmových objektů pomocí Bézierovy křivky, přinese nové filtry pro zpracování obrazu a také analýzu různých videozáznamů.

## 2.2 Vyšetření krční tepny

Vyšetření krční tepny bylo řešeno v rámci grantových úkolů GAČR na 3D analýzu videa zachyceného ultrazvukovou sondou, které úspěšně vyřešil doc. Ing. Lačezar Ličev, CSc.

Vzhledem k tomu, že se sonda snímající tepnu při každém úderu srdce pohne o 1 mm, je zapotřebí z ultrazvukového záznamu vybrat snímky, které zachycují stav tepny v určitém opakujícím se okamžiku - intervalu (tj. doby mezi každými dvěma údery srdce). V každém intervalu dochází totiž k roztažení a opětovnému smrštění cévy. Cílem je zachytit cévu ve stejné míře roztaženosti jako v předchozím intervalu, i když intervaly v praxi nejsou stejně dlouhé. Je proto potřeba navrhnout a vytvořit modul pro automatické rozpoznávání těchto událostí, aby se co nejvíce minimalizoval lidský faktor.

## 2.3 Další možnosti použití

Modul pro analýzu videozáznamů ultrazvukových vyšetření nachází své využití i mimo medicínský sektor. Jeho nasazení je vhodné všude tam, kde vzniká video záznam kamerou pohybující se konstantní rychlostí. Jako příklad můžeme uvést měření výtahové šachty, štol, důlní jámy a další. Nemusíme se ale omezovat jen na podzemní použití. Jako sondu si můžeme představit i letadlo, které letí konstantní rychlostí a snímá krajinu, která pod ním probíhá.

# 3 SOFTWAREVÉ PROSTŘEDKY

## 3.1 Netbeans Platform

Pro vývoj aplikací v Javě se nabízí několik různých vývojových prostředí. Výběr začíná u jednoduších textových editorů, které umožňují pouze zvýrazňovat syntaxi a končí u komplexních vývojových prostředí, které kromě zvýrazňování syntaxe nabízejí celou škálu dalších pomocných nástrojů (debugger, syntaktická analýza, zvýrazňování chyb v kódu, inteligentní doplňování kódu, profilování, GUI (Graphical user interface) designér, a další), které mají programátorovi co nejvíce ulehčit práci. Mezi dvě nejznámější a nejrozšířenější patří NetBeans IDE (Integrated development environment) a Eclipse. Obě tato vývojová prostředí nabízejí i své vlastní verze platformem.

Jelikož je systém Fotom 2009 postaven na architektuře NetBeans Platform, bylo celkem jasné, že modul byl vyvíjen právě v prostředí NetBeans IDE.

## 3.2 Java Advanced Imaging

Kvůli omezeným možnostem standardního balíku Java2D bylo použito při vývoji jádra aplikace FOTOM 2009 rozšíření Java Advanced Imaging (JAI), které je vhodnější pro složitější transformace a zpracování složitějšími filtry [6]. Velké množství nástrojů, které JAI nabízí, bylo implementováno pro snadnější použití přímo do jádra aplikace FOTOM 2009.

## 3.3 Zpracování videa

Java nabízí pro práci s videem svůj vlastní framework JMF. Toto byla hned první možnost, kterou bylo možné použít. Avšak po zjištění, že poslední verze tohoto frameworku (JMF 2.1.1e) byla vydána v roce

2003, bylo rozhodnuto, že je tento framework již velice zastaralý a měl by problém zpracovat novější formáty videa.

Další alternativy JMF jsou třeba FMJ, který využijí hlavně vývojáři, kteří postavili svoji aplikaci na JMF, ale chtěli by přejít na něco novějšího. Jeho hlavní výhodou je v tom, že má stejné API, a proto je snadné vyměnit zastaralý JMF za FMJ. Dále stojí za zmínku Jffmpeg, Direct-Show Java Wrapper, Fobs a nebo Xuggler, které uživatelé doporučují

Xuggler

Xuggler je zapouzdření projektu FFmpeg, který umožňuje nahrávání, konverzi a streamování digitálního zvuku (audia) a obrazu (videa). Jelikož je použití projektu FFmpeg v jazyce Java nepohodlné a složité, vznikl kvůli tomuto účelu projekt Xuggler, který svým uživatelům nabízí multiplexování a demultiplexování všech nejpoužívanějších formátů (avi, flv, mov, mp4, 3gp, h264 a mnoho dalších).

Xuggler se skládá ze dvou komponent. Jednou z nich je sada JAR souborů a druhou je sada sdílených knihoven (.dll pro systém Windows, .so pro Linux a .dvlb pro Mac). Xuggler dále nabízí dvě základní API.

- MediaTool API, které poskytuje jednoduché rozhraní pro rychlé psaní kódu
- Advanced API, které nabízí programátorovi maximum z možností Xuggleru

## 4 NÁVRH A IMPLEMENTACE MODULU

Hlavním úkolem bylo vytvořit modul, který bude možné lehce začlenit do stávající aplikace FOTOM 2009 a který bude maximálně využívat možnosti jeho jádra.

### 4.1 Video přehrávač

#### 4.1.1 Specifikace požadavků

Dále bylo nutné navrhnout video přehrávač, který bude umět přehrát co největší množství formátů videa, a bude podporovat základní funkce, jako jsou Přehrát, Pauza, Stop, Následující snímek, Předchozí snímek a Přejít na určitou pozici. Přehrávač by také měl být schopen plynule přehrávat video, i když se vyskytne nečekaná zátěž systému a snímky se nestíhají dekodovat tak rychle, jak se očekává.

#### 4.1.2 Návrh

Pro minimalizaci „sekání“ videa byl navržen systém producenta a konzumenta, kde producent „vyrábí“ dekodované snímky a ukládá je do zásobníku, zatímco konzument tyto snímky ve správný čas ze zásobníku vytáhne a zobrazí na obrazovku. Toto se děje ve dvou souběžně běžících vláknech (vlákno se využívá při programování paralelních systémů). Proto bylo nutné se vypořádat se správnou synchronizací těchto vláken tak, aby se vlákno producenta potlačilo vždy, když je zásobník plný a aktivovalo se pokaždé, když je možné do zásobníku opět vkládat. A naopak je stejně tak nutné, aby se konzument potlačil vždy, když je zásobník prázdný a aktivoval, když se do něj něco vloží.

#### 4.1.3 Implementace

Pro programátory, kteří pracují s více vlákny, nabízí Java celou kolekci tříd (balíček `java.util.concurrent`), které jsou přímo napsány tak, aby měl programátor co nejméně práce se synchronizací a aby nemusel řešit nutné zamykání objektů. Z této kolekce bylo použito rozhraní `BlockingQueue` použitím třídy `ArrayBlockingQueue`, která ukládá objekty v požadovaném pořadí FIFO.

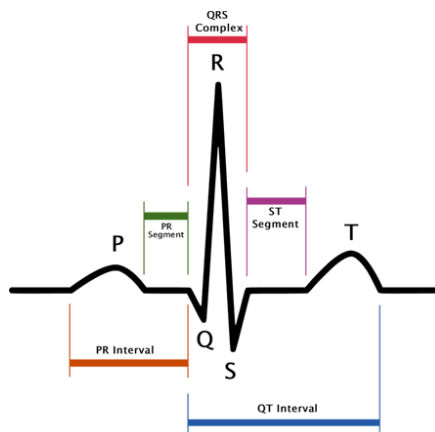
Při přehrávání je také nutné hlásit každou změnu pozice ve videu grafickému rozhraní, aby se mohlo adekvátně obnovit textové políčko s informacemi o aktuální časové známce a číslem aktuálního snímku a také aby se aktualizovala pozice posuvníku. Tento systém je založen na principu událostí a posluchačů.

## 4.2 Analýza videa

Základní pojmy:

Elektrokardiografie (EKG) Technika záznamu činnosti srdce. Elektrody spojené se zaznamenávajícím přístrojem (elektrokardiograf) se umístí na kůži čtyř končetin a na hrudní stěnu. Záznam se nazývá elektrokardiogram.

Elektrokardiogram Záznam elektrické aktivity srdce na pohybujícím se papírovém proužku. Tento záznam je pořízen přístrojem nazývaným elektrokardiograf. Užívá se při diagnostice srdečních onemocnění, které mohou způsobit typické změny EKG.



Obr. 1. Běžný průběh signálu EKG

Impuls pro kontrakci myokardu vzniká v tzv. sinoatriálním (SA) uzlu v oblasti pravé předsíně, odkud se šíří dál. Pro účel našeho stručného výkladu je důležité si uvědomit, že tento primární signál je natolik slabý, že jej při běžném záznamu EKG prakticky nezaznamenáme. První vlna signálu, kterou můžeme na EKG záznamu vidět, je vlna P, která svědčí o depolarizaci předsíní, tedy o jejich počínající kontrakci. Samotnou repolarizaci předsíní na EKG nejsme schopni rozpoznat, neboť příslušný biosignál je zastíněn daleko vyšším signálem, pocházejícím od depolarizace komor [3]; tento signál je charakterizován komplexem vln QRS. Následující vlna T svědčí o další repolarizaci komor [3].

#### 4.2.1 Specifikace požadavků

Dalším krokem je vymyslet způsob, jakým se bude analyzovat video, aby bylo dosaženo požadovaných výsledků. Obrázek 2 nám ukazuje jeden zachycený snímek z analyzovaného videa. Dole na snímku si prosím povšimněte EKG křivky s mezerou uprostřed. Tato mezera znamená místo, které se momentálně překresluje na novou hodnotu. Nejaktuálnější hodnota je v místě, kde končí levá část grafu. Hned za mezerou v oblasti, kde začíná pravá strana grafu, je zaznamenána nejstarší hodnota, která se na dalším snímku překreslí. V tomto grafu je třeba rozpoznat jednotlivá místa tak, že když si uživatel vybere, že ho zajímá, jak vypadá ultrazvukový snímek ve chvíli, kdy nastane např. vlna T, měly by se mu ukázat všechny snímky, které právě probíhající vlnu T zobrazují.

#### 4.2.2 Návrh

Strojové rozpoznávání oblasti EKG grafu je velice náročné a nespolehlivé a proto uživatel má k dispozici nástroj SelectionTool, kterým označí oblast, ve které se graf nachází a jen s touto oblastí se pak dále bude pracovat. Bude se jednat o klasický nástroj obdélníkového výběru oblasti, se kterým se již uživatel určitě setkal v mnoha jiných programech. Výhodou práce s menší oblastí je hlavně rychlosti zpracování.

Skutečnost, že se graf neustále dokola překresluje, velmi komplikuje jeho analýzu. Proto by bylo vhodné sestavit kompletní kontinuální pás grafu, se kterým se bude dále lépe pracovat. Nyní si prosím všimneme bílé šipky v pravém dolním rohu. Zde se naskytuje další úskalí, se kterým bylo nutné se vypořádat. Část pravé strany grafu je totiž zakrytý písmenkem B a rozpoznávání tvaru grafu je v tomto místě prakticky nemožné. Lidské oko si dokáže chybějící část domyslet, ale pro počítač je to velký problém.

Při sestavování pásu grafu bude procházen každý snímek videa, tedy jen jeho výřez s grafem, který určí uživatel označením oblasti. Na každém tomto snímku je nutné najít, kde se nachází právě vykreslovaná hodnota. Proto je hledána mezera, která graf překresluje. V prvním snímku je nutno tedy projít celý výřez

snímku a v každém sloupci hledat již vykreslený graf. Pokud je toto místo nalezené, posuneme se na další sloupec a hledání se opakují. Pokud ovšem ve sloupci není čára grafu nalezena, znamená to, že se může jednat o překreslovací mezeru, ale také se může jednat jen o chybu. Před hledáním je vhodné totiž obrázek předzpracovat prahováním (thresholding). Prahování je funkce, která obecně upravuje hod-noty vstupu podle předpisu [4]:



Obr. 2. Ultrazvukový snímek z videa

$$f(c) = \begin{cases} A & \text{pokud } c < \text{práh} \\ B & \text{pokud } c \geq \text{práh} \end{cases} \quad (1)$$

kde:

$c$  - je vstupní hodnota jasu nebo barvy

$f(c)$  - výsledná hodnota

práh - je prahovací hodnota

A,B - jsou nové hodnoty pro vstupní hodnotu pod a nad prahem

V tomto případě bude hodnota nastavena na 0 a hodnota na 255, tedy černá a bílá. Hodnota prahu je na základě dlouhém zkoumání nastavena na hodnotu 72. Toto předzpracování má za úkol eliminovat šum na pozadí a zvýraznit samotný graf (Obrázek 3). Jako vedlejší efekt ovšem mohou vzniknout i umělé díry a nekonzistence grafu. Proto je třeba nastavit jistou toleranci při hledání vykreslovací mezery, aby se co nejvíce eliminovalo toto riziko.



Obr. 3. EKG graf předzpracovaný prahováním

Jakmile se nalezne vykreslovací mezeru, algoritmus si zapamatuje několik sloupců vlevo od této mezery a pokračuje v hledání na dalším snímku. Nakonec vše spojí dohromady a vznikne nám tak dlouhý pás s odpovídajícím grafem. Jak již bylo řečeno, nám velice komplikuje ono písmeno B, které zakrývá část grafu, a proto je nemožné nalézt v tomto místě vykreslovací mezeru, pokud by se tam nacházela. Jestliže algoritmus nenalezne v celém výřezu žádnou vykreslovací mezeru, má se za to, že je mezeru schována za tímto písmenem a tudíž se do paměti uloží jen prázdné místo. Ve výsledku pak má pás s grafem pár mezer, které můžou vytvářet problémy, pokud v tomto místě měly být vykresleny kmity QRS komplexu (viz Obrázek 1). V následující části jsou na pásu grafu vyhledávány jednotlivé QRS komplexy, které odpovídají periodám a na jejichž základě se měří srdeční tep.

Na detekci QRS komplexů existuje mnoho algoritmů. Všechny ale počítají s tím, že je vstupem přesný signál a ne nekvalitní obrázek grafu ve špatném rozlišení. Proto je navržen vlastní algoritmus, jak detekovat QRS komplexy z obrázku. Ten je založen na výpočtu výběrového rozptylu. Je zřejmé, že pro rovné části grafu bude rozptyl velmi malý, zatímco u rozkmitů bude rozptyl mnohokrát větší.

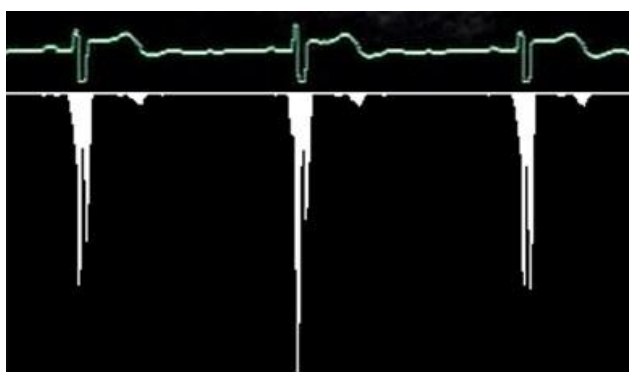
Ve statistice se výběrový rozptyl definuje jako podíl součtu kvadrátů odchylek jednotlivých hodnot od průměru a rozsahu souboru sníženého o jedničku [2], tedy:

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} \quad (2)$$

kde:

- $x_i$  - je v našem případě výška grafu ve sloupci, neboli x-ová souřadnice pro
- $\bar{x}$  - je aritmetický průměr
- $n$  - je počet sloupců

Výpočet rozptylu je prováděn vždy jen pro malou část grafu, např. okénko o délce 5px, které při dalším průchodu posunu o 1px doprava. Pro každé takovéto okénko je získáno číslo, které vyjadřuje rozptyl grafu v tomto okénku. Obrázek 4 ukazuje, jak vypadá sloupcový graf vypočtených hodnot rozptylu pro malou část vstupního EKG grafu. V tomto okamžiku mám tedy množinu rozptylů a bude nás zajímat vždy pouze první největší rozptyl. Pořadové číslo tohoto rozptylu nám odhalí místo QRS komplexu.



Obr. 4. Výpočet rozptylu

Jakmile jsou nalezeny všechny QRS komplexy, mohou si lehce rozdělit graf na jednotlivé periody. Když si uživatel určí, že ho zajímá nějaké konkrétní místo na grafu, přepočítá se toto místo relativně na polohu v té konkrétní periodě a tato relativní poloha se zaznačí i do ostatních period. Tímto způsobem ošetříme to, že jednotlivé periody nejsou stejně dlouhé. Např. při zvýšené tělesné námaze se nám hned zrychlí srdeční tep, což se projeví zkrácením period.

Nyní jsou v grafu označená místa, které uživatele zajímají, a už zbývá jen jejich přemapování na čísla snímků, které se mohou uložit na disk jako obrázky.

#### 4.2.3 Implementace

Aby si mohl uživatel označit oblast, ve které se nachází EKG graf, byl vytvořen nástroj zvaný SelectionTool. Nástroj využívá API (Application programming interface) Fotomu a lze s ním tedy pracovat jako s kterýmkoliv jiným již implementovaným nástrojem. Při psaní nástroje byl kladen důraz na to, aby bylo efektivně vyřešeno jeho vykreslování – překresluje se jen ta část plátna (okna), která se změnila. Pro malá plátna je celkem jedno kolik se toho překresluje, ale na velkých plátnech je to již znatelné, např. u akce přesouvání výběru, kdy je třeba překreslovat plátno s nástrojem real-time. Výběr je možné tažením myši přesouvat a měnit velikost. Při implementaci nástroje je počítáno i s tím, aby uživatel nemohl přetáhnout nebo zvětšit výběr mimo plátno. Nástroj má také množství vlastností, které může uživatel jednoduše měnit v nastavení. Platforma NetBeans nabízí k tomuto účelu okénko, do kterého se vloží nastavitelné vlastnosti.

Integrace vytvořeného nástroje do aplikace spočívá ve vložení tlačítka do palety nástrojů, kterou vytváří VideoAnalyzerTopComponent. Nejprve je třeba vytvořit popisný XML soubor, který obsahuje informace o tom, která třída reprezentuje nástroj, dále jaké ikony bude nástroj používat a také jeho název a tooltip.

Definovaný nástroj je ještě třeba vložit do samotné palety, která ho zobrazí jako tlačítko s definovanou ikonou a názvem. Vkládání se provádí v souboru layer.xml, kde VideoAnalyzerToolPalette je název palety,

do které se nástroj vloží, což umožňuje, aby měl každý modul svou vlastní paletu, Tools je název kategorie a SelectionTool.xml je definice nástroje.

Pro sestavování grafu je připravena hlavní třída EkgBuilder. Tato třída dědí vlastnosti z třídy SwingWorker a implementuje rozhraní Cancellable. Tzn., že se může spouštět na pozadí a že se také může její běh zastavit.

V průběhu sestavování EKG grafu i během jeho analýzy pracuji s obrazem, který prošel před-zpracováním prahování. K tomuto účelu mi posloužily třídy z jádra aplikace RgbToGrayscale a BinarizeTreshold. První slouží k převodu obrázku do stupňů šedi a druhá provede samotné prahování.

Detekce QRS komplexu k výpočtu rozptylu, který se v detekci QRS komplexu používá, slouží třída Variance. Při vytváření instance se předá konstruktoru malé okénko grafu, ze kterého se rozptyl vypočte a metodou getVariance() se vrátí jeho hodnota. Grafickou reprezentaci vypočtených hodnot pro malý výřez grafu ukazuje Obrázek 4. Na tomto obrázku je možné si všimnout, že výpočtem vzniknou pro každou periodu dvě lokální maxima. První označuje hledaný QRS komplex a druhé maximum odpovídá umístění vlny S (viz Obrázek 1). Jakmile jsou vypočteny všechny rozptyly, začnou se postupně procházet a porovnávat. Pokud je rozptyl větší než 27 pixelů, má se za to, že se jedná o QRS komplex a jeho poloha se zaznamená. Po nalezení jednoho QRS komplexu se přesuneme o pár pixelů doprava, abychom se vyhnuli falešné detekci, protože vlna S má hodnoty rozptylu také větší než 27 pixelů. Tak najdeme všechny QRS komplexy a podle jejich umístění rozdělím graf na jednotlivé intervaly. Po kliknutí myši do oblasti sestaveného grafu se zobrazí náhledy snímků, které odpovídají stejné události v každém intervalu. Tyto snímky je možné kdykoliv uložit na disk.

Nalezené hranice interval ovšem nejsou konečné. Pokud není uživatel spokojený s výsledkem automatické detekce, má ještě možnost ručního nastavení hranic intervalů a to tak, že je jednoduše přetáhne na nové místo. Při tažení první hranice intervalu dochází k posunu všech hranic najednou a při tažení jakékoliv jiné hranice dochází ke změně měřítka. I po nastavení nových hranic je možnost se vrátit zpět na původní hranice, protože se staré hranice pořád udržují v paměti.

### 4.3 Ukládání snímků

#### 4.3.1 Specifikace požadavků

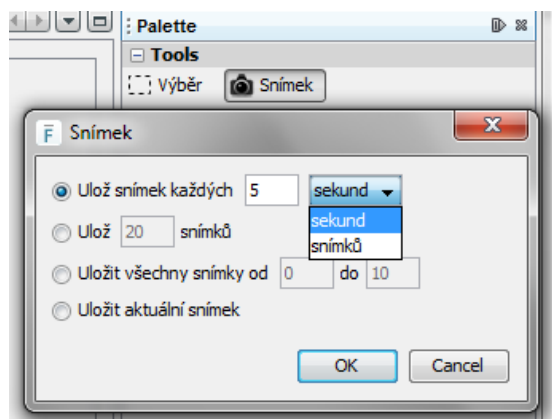
Posledním požadavkem je možnost ukládat vybrané snímky na disk nepravidelně na základě událostí v EKG grafu a také podle pravidel, které si bude moci uživatel vybrat a sám nastavit. Prvním pravidlem bude možnost vytáhnout z celého videa snímek každých sekund nebo každých snímků. Dále by měl mít uživatel možnost uložit z videa např. 20 snímků, které jsou z videa vybrány v pravidelných intervalech. Třetím pravidlem je možnost uložit všechny snímky, které jsou v určitém intervalu a poslední možnost je uložit aktuálně zobrazený snímek.

#### 4.3.2 Návrh

Pro univerzální využití je navržena metoda, která vrátí obrázek na pozici, která jí byla předána parametrem. Pro výběr možností, které snímky se uloží, bude mít uživatel k dispozici panel s tlačítky a textovými poli, kde si vše potřebné sám nastaví. O samotné ukládání snímků se pak bude starat třída k tomu určená a bude umět ukládat snímky v základních formátech BMP, JPEG, GIF a PNG.

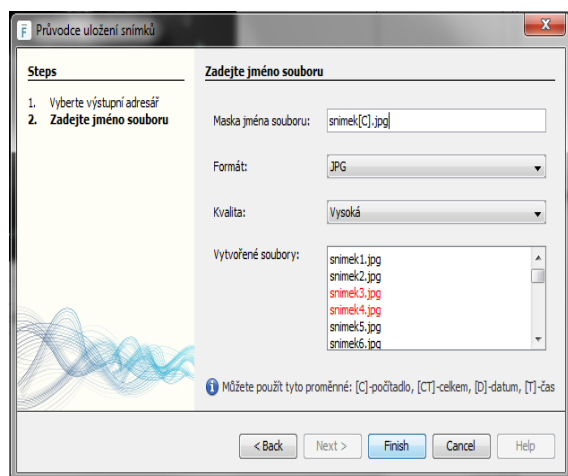
#### 4.3.3 Implementace

Pro jednoduché nastavení jsem vytvořil v paletě nástroj, který zobrazí panel se zaškrťovacími tlačítky a textovými poli, kde si uživatel nastaví, jaké snímky chce uložit, viz Obrázek 5. Při ukládání snímků se zobrazí průvodce uložení snímků Obrázek 6, kde je možnost zvolit výstupní adresář, kam se snímky uloží, masku jejich názvu, formát a pokud se jedná o jpeg, tak i kvalitu. Před samotným uložením je k dispozici náhled souborů, které budou vytvořeny a pokud již některé soubory existují, zvýrazní se v tomto seznamu červeně. Při ukládání se pak ještě program pro jistotu ptá, jestli se má existující soubor opravdu přepsat.



Obr. 5. Možnosti nástroje Snímek

Obrázek 7 demonstuje vzhled výsledného modulu. V levé horní části je okno prohlížeče, ve kterém se zobrazují uložené objekty. Levá spodní část obsahuje seznam objektů, definovaných na plátně. V tomto případě lze na plátno přehrávače přidávat pouze objekty nástroje výběru, takže toto je jediný typ, který se v seznamu objeví. Definované objekty lze upravovat a dle libosti mazat. V pravé části se nachází paleta s nástroji a pod ní okno, kde se dá přesněji nastavit velikost výběru. Zbývá hlavní okno aplikace, kde se přehrává video a vykresluje graf spolu s vybranými snímky. Přehrávač lze ovládat pomocí tlačítek umístěných pod jeho plátnem a video je možné také točením kolečka myši přibližovat a oddalovat. Vedle ovládacích prvků jsem umístil informační proužek, ve kterém se vypisuje číslo aktuálního snímku a jeho čas. Pod tlačítky je vymezen prostor pro vykreslení sestaveného pásu grafu. Vykreslený graf umožňuje a očekává interakci uživatele. Ten si tak může vybrat místo, které ho na grafu zajímá a po výpočtu všech ostatních odpovídajících míst se napravo od plátna přehrávače zobrazí náhledy vybraných snímků.



Obr. 6. Průvodce uložení snímků



Obr. 7. Celkový pohled na výsledný modul

## 5 ZÁVĚR

Cílem této práce bylo popsat vývoj a vyvinout modul pro provedení „rozsekání“ videozáznamů na jednotlivé snímky různých formátů dle událostí zachycených na tomto záznamu. Dle našeho názoru byl cíl práce splněn a výsledný modul byl poté integrován do systému FOTOM 2009.

Vyvinutí modulu poskytuje možnost načíst a přehrát videa různých formátů. Uživateli se tak dostává k dispozici mocný nástroj určený hlavně k ulehčení analýzy ultrazvukových záznamů. Modul poskytuje základní funkce video přehrávače, umí z videa vybrat a uložit jednotlivé snímky na základě uživatelského nastavení a také analyzovat video a rozpoznávat tak jednotlivé periody EKG grafu, který je na video záznamu zachycen. Uživatel má také možnost nastavit si sám intervaly, pokud nebyly rozpoznány nebo



pokud byly rozpoznány špatně. Při tvorbě video přehrávače byl kladen důraz na plynulost přehrávání i při neočekávané zátěži systému.

Pomocí tohoto modulu lze analyzovat videozáznam ultrazvukového vyšetření a tím umožnit lepší prevenci vzniku cévní mozkové příhody (CMP). CMP je v současnosti třetí nejčastější příčinou úmrtí v lidské populaci. Jedním z řešení, jak snížit výskyt CMP je právě prevence a její včasné odhalení. Vyšetření ultrazvukovou sondou je momentálně nejvhodnější metodou pro toto sledování a pomocí vytvořeného modulu je možno snímky lépe analyzovat.

V současné době jsou do systému FOTOM 2009 vyvíjeny další dva moduly. První modul umožní 2D modelování. Druhý modul poskytuje rozšířené možnosti prahování, definici objektů pomocí Bézierovy křivky a filtry pro úpravu jasu a kontrastu. Doplnění stávající verze systému FOTOMU o tyto tři nové moduly vytvoří novou verzi aplikace – FOTOMNG.

## LITERATURA

- [1] BÖCK, H. (2010) *Platforma NetBeans: Podrobný průvodce programátora*. 1. vyd. Brno: Computer Press, 2010. 320 s. ISBN 978-80-251-3116-9
- [2] BRIŠ, R., LITSCHMANNOVÁ, M. (2004) *Statistika I.: pro kombinované a distanční studium* [online]. 2004. [cit. 2011-04-20] URL: <[http://is457.vsb.cz/bris/Teaching/Statistika\\_1/Kapitola\\_1.PDF](http://is457.vsb.cz/bris/Teaching/Statistika_1/Kapitola_1.PDF)>
- [3] *Elektrokardiogram – Wikipedie* [online]. 2011, poslední revize 5. března 2011 [cit. 2011-04-23] URL: <<http://cs.wikipedia.org/wiki/Elektrokardiogram>>
- [4] LIČEV, L. (2010) *Analýza, modelování, rozpoznávání a vizualizace procesu měření objektů na snímcích*. Ed. Computer Press. 1. vyd. Brno: Computer Press, 2010. 125 s. ISBN 978-80-251-3296-8
- [5] *Prahování – Wikipedie* [online]. 2011, poslední revize 12. ledna 2011 [cit. 2011-04-16] URL: <<http://cs.wikipedia.org/wiki/Prahování>>
- [6] SUN Microsystems. *Programming in Java Advanced Imaging* [online]. 1999, poslední revize 18. prosince 2007 [cit. 2011-04-13] URL: <[http://java.sun.com/products/java-media/jai/forDevelopers/jai1\\_0\\_1guide-unc/](http://java.sun.com/products/java-media/jai/forDevelopers/jai1_0_1guide-unc/)>