

Návrh

Vision Lab – fyzikálne experimenty

Skupina SEJ2

Soňa Senkovičová, Erik Szalay, Jozef Kubík, Juraj Vetrák

16.11.2018

Obsah dokumentu

1. ÚVOD	3
1.1 ÚČEL DOKUMENTU	3
1.2 POUŽÍVANÉ DEFINÍCIE, AKRONYMY A SKRATKY	3
2. ŠPECIFIKÁCIA VONKAJŠÍCH INTERFEJSOV	3
3. FORMÁTY SÚBOROV	4
3.1 AKTUÁLNA SNÍMKA Z WEBOVEJ KAMERY V POZASTAVENOM ZÁZNAMU – JPG	4
3.2 AKTUÁLNY ZÁZNAM GRAFU - JPG	4
3.3 EXPORT DOKUMENTU Z POZASTAVENÉHO ZÁZNAMU – PDF	5
3.4 EXPORT ŠTATISTICKÝCH ÚDAJOV Z POZASTAVENÉHO ZÁZNAMU – CSV	6
3.5 IMPORT AJ EXPORT KONFIGURAČNÉHO SÚBORU - TXT	6
4. POUŽÍVATEĽSKÉ ROZHRAŇOVANIE	6
5. NÁVRH IMPLEMENTÁCIE	8
5.1 PREHĽAD POUŽÍVANÝCH TECHNOLOGIÍ	8
5.1.1 Programovací jazyk	8
5.1.2 Knižnica pre manipuláciu s počítačovou grafikou	9
5.1.3 Knižnice pre tvorbu užívateľského prostredia	10
5.1.4 Knižnica pre export PDF	10
5.2 ARCHITEKTÚRA APLIKÁCIE	11
5.3 DIAGRAMY	11
5.3.1 Triedny diagram	11
5.3.2 Komponentový diagram	11
5.3.3 Stavový diagram	11
5.3.4 Sekvenčný diagram	12
5.4 ROZDELENIE NA ČASTI (MODULY)	12
5.5 CIEĽOVÉ PROSTREDIE NASADENIA DO PREVÁDZKY	13

1. Úvod

1.1 Účel dokumentu

Účelom tohto dokumentu je špecifikovať formu a technologické detaily vývoja aplikácie Vision Lab - Fyzikálne experimenty. Dokument je určený pre vývojárov aplikácie.

1.2 Používané definície, akronymy a skratky

Na tejto sekcii sa momentálne pracuje.

2. Špecifikácia vonkajších interfejsov

Na tejto sekcii sa momentálne pracuje.

3. Formáty súborov

3.1 Aktuálna snímka z webovej kamery v pozastavenom zázname – JPG

Načítanie obrázku (screenshotu) z kamery:

```
#include <opencv2/opencv.hpp>

using namespace cv;

int main( int argc, char** argv ) {
    char* imageName = argv[1];
    Mat image;
    image = imread( imageName, 1 );

    if ( argc != 2 || !image.data ) {
        printf( " No image data \n " );
        return -1;
    }

    imshow( imageName, image ); // prípadné zobrazenie obrázku

    ...
}
```

Uloženie obrázku (screenshotu) z kamery:

```
...

imwrite( "../images/imageName.jpg", image );
```

Uloženie obrázku sa realizuje do preddefinovaného priečinka, odkiaľ ho bude načítavať libHaru [\[3.3\]](#).

3.2 Aktuálny záznam grafu - JPG

~~V pozastavenom zázname sa nachádza graf, ktorý popisuje aktuálny stav sledovaného objektu. Ten sa vyexportuje vo formáte JPG s bielym/transparentným pozadím a uloží do vývojarom nastaveného priečinka a použije sa pri tvorbe “dokumentu” [\[3.3\]](#).~~

3.3 Export dokumentu z pozastaveného záznamu – PDF

Predpokladaná je úspešná inštalácia libHaru [\[5.1.4\]](#).

Inicializovanie objektu vytváraného dokumentu (HPDF_Doc), prípadne odchytenie chybových hlások:

```
#include "hpdf.h"

HPDF_Doc pdf;
pdf = HPDF_New (error_handler, NULL);
```

Vytvorenie novej stránky dokumentu:

```
HPDF_Page page_1;

page_1 = HPDF_AddPage (pdf);
```

Písanie textu (začiatok):

```
HPDF_STATUS HPDF_Page_BeginText (HPDF_Page page_1);
```

Písanie textu:

```
HPDF_STATUS HPDF_Page_TextOut (HPDF_Page    page_1,
                                HPDF_REAL    xpos, // pozícia x
                                HPDF_REAL    ypos, // pozícia y
                                const char    *text);
```

V libHaru sú začiatkové pozície pre x, y (0, 0) a nachádzajú sa v ľavom dolnom rohu dokumentu.

Písanie textu (ukončenie):

```
HPDF_STATUS HPDF_Page_EndText (HPDF_Page page);
```

Uloženie dokumentu na disk:

```
HPDF_SaveToFile (pdf, "test.pdf");
```

Pokračovanie v tvorbe nového dokumentu:

```
HPDF_NewDoc (pdf); // odstráni pôvodný
```

Uvoľnenie všetkých zdrojov:

```
HPDF_Free (pdf);
```

3.4 Export štatistických údajov z pozastaveného záznamu – CSV

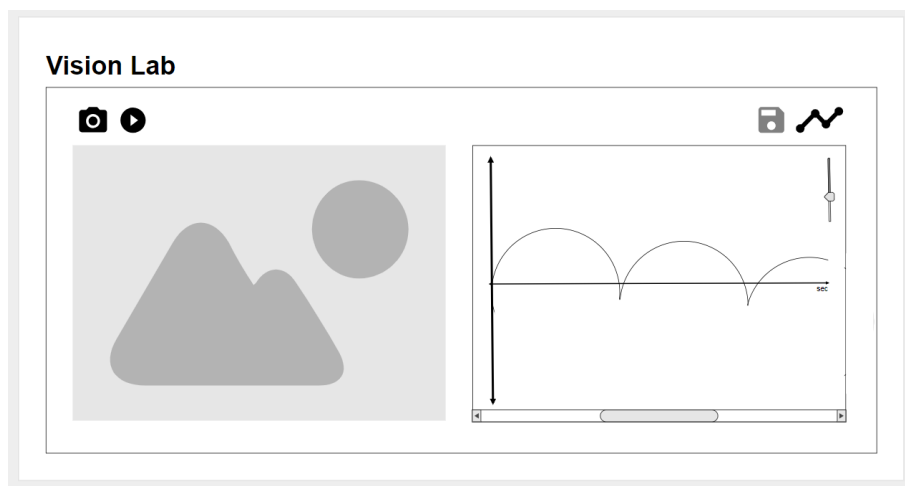
Po zvolení možnosti exportovania štatistických údajov sa vytvorí CSV súbor, ktorý bude obsahovať čiarkou oddelené údaje, ktoré si zvolil užívateľ pri grafe v checkboxoch. Otvorí sa prehliadač súborov s možnosťou uložiť tento CSV dokument na ľubovoľné, užívateľom zvolené miesto v počítači.

3.5 Import aj export konfiguračného súboru - TXT

Aplikácia bude mať (nie len) pri prvom spustení k dispozícii konfiguračný súbor so základnými nastaveniami webovej kamery v súlade s predpokladmi používania aplikácie, a to vo formáte TXT (formát TXT súboru ešte upresniť detailne). Ak užívateľ v aplikácii zmenil/prispôbil nastavenia svojej kamery pre svoje potreby, tieto nastavenia sa uložia do ďalšieho konfiguračného súboru a budú k dispozícii v prípade identifikácie danej konkrétnej kamery, čím sa automaticky nastaví už predtým zvolená konfigurácia a odpadne nutnosť znova nastavovať tú istú kameru. Exportovaná konfigurácia sa uloží do vývojarom vybraného priečinka a taktiež do formátu TXT.

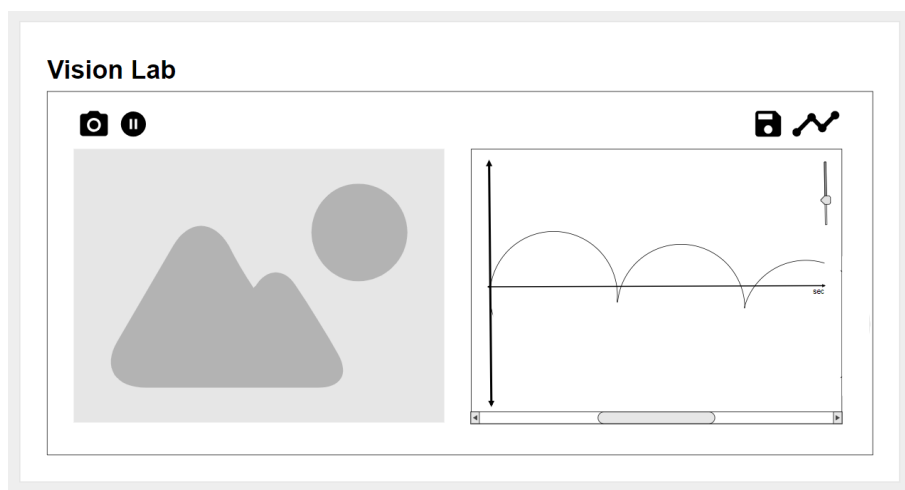
4. Používateľské rozhranie

4.1 Hlavná obrazovka



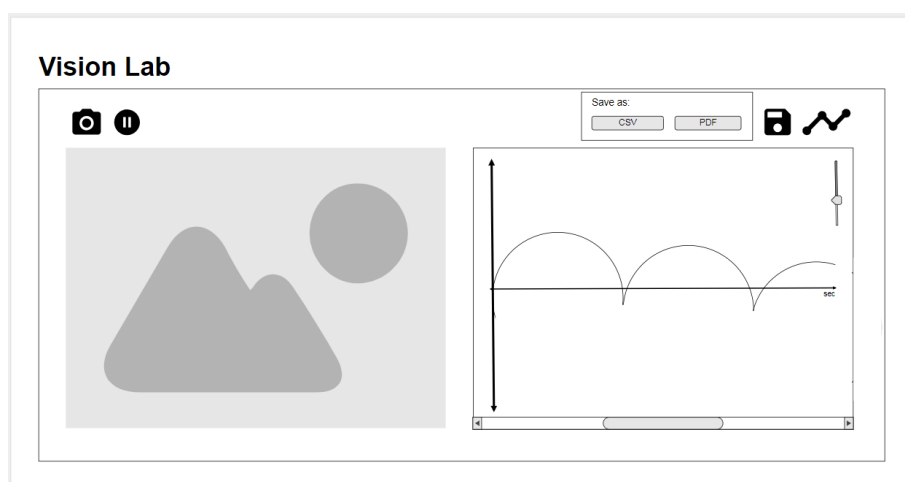
obr. 4.1 - hlavná obrazovka

4.2 Spustené meranie



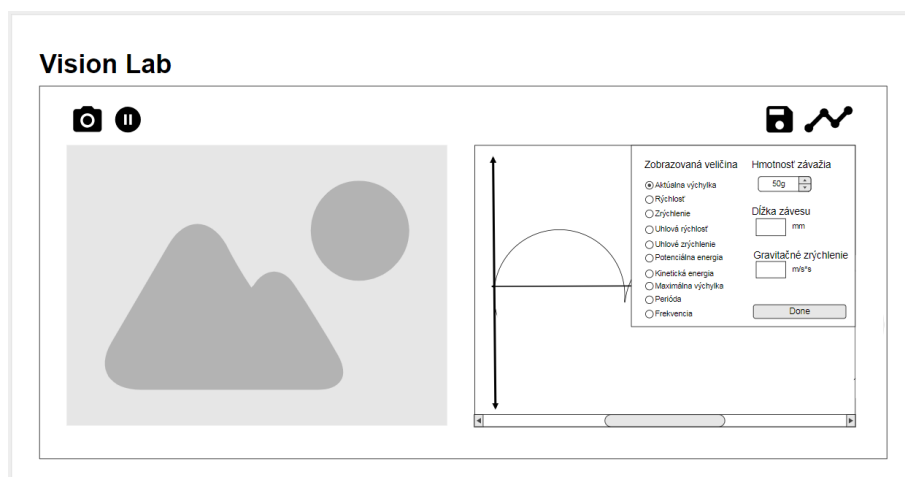
obr. 4.2 - spustené meranie

4.3 Exportovanie údajov



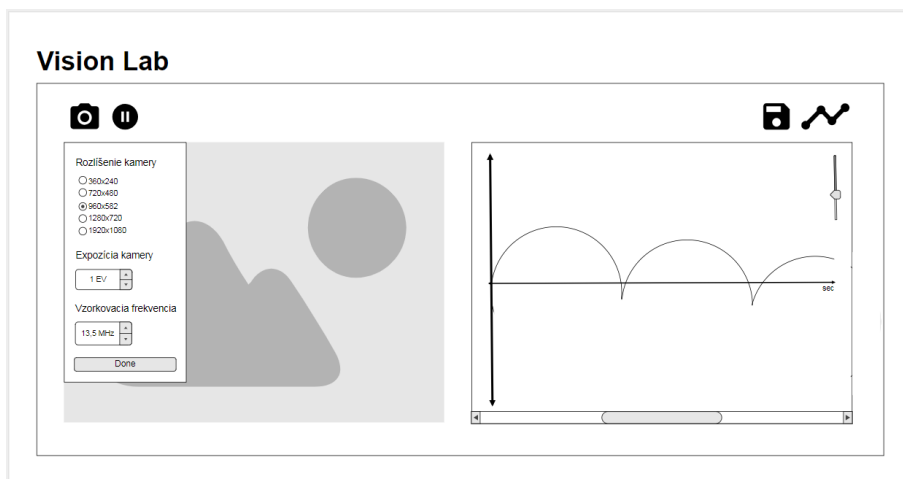
obr. 4.3 - exportovanie údajov

4.4 Nastavenie grafu a kyvadla



obr. 4.4 - nastavenie grafu a kyvadla

4.5 Nastavenie webovej kamery



obr. 4.5 - nastavenie webovej kamery

4.6 Popis

Aplikácia umožňuje sledovať záznam pohybu kyvadla a zároveň sledovať vybraný údaj (obr 4.1). Užívateľ má možnosť zastaviť a znovu spustiť záznam (obr. 4.2). Môže si vyexportovať údaje do formátov CSV (štatistické údaje) alebo PDF (rozsiahlejší dokument [\[3.3\]](#) (obr 4.3). Užívateľ si vie nastaviť, ktorý údaj chce pozorovať. Vedľa vie zadať hmotnosť závažia, dĺžku lanka a gravitačné zrýchlenie oblasti v ktorej je. Pomocou scrollbaru sa dá sledovať aj história grafu. Graf sa dá priblížiť alebo oddialiť (obr 4.4). Užívatelia majú možnosť nastaviť rozlíšenie a expozíciu kamery. Okrem toho vedia nastaviť aj frekvenciu snímania (obr 4.5).

5. Návrh implementácie

V tejto sekcii je podrobne popísaný návrh celkovej implementácie softvéru pozostávajúci z prehľadu používaných technológií, detailného popisu softvérovej architektúry, nakreslených diagramov popisujúcich funkčnosť aplikácie, rozdelenie na časti (moduly) a popis cieľového prostredia pri nasadení do prevádzky.

5.1 Prehľad používaných technológií

5.1.1 Programovací jazyk

Ako programovací jazyk bol zvolený C++, keďže predstavuje základný jazyk pre knižnicu OpenCV, ktorá bola použitá pre manipuláciu s kamerou a počítačovou grafikou.

C++ je objektovo orientovaný programovací jazyk so širokým spektrom dostupných knižníc. Keďže C++ predstavuje rozšírenú verziu C, ktorá je pomerne nízkoúrovňová, poskytuje oveľa väčšiu rýchlosť oproti vysokoúrovňovým jazykom ako Python alebo Java.

5.1.2 Knižnica pre manipuláciu s počítačovou grafikou

OpenCV (Open Source Computer Vision, <https://opencv.org/>) je knižnica pre manipuláciu s obrázkami a real-time videom. Je napísaná v C++, ktorý je aj jej primárnym interface jazykom.

OpenCV spracováva video ako sadu obrázkov (frameov). Tieto obrázky sú transformované do matice Mat, ktorá sa skladá z 2 častí:

- hlavičku (header) - obsahuje informácie o veľkosti (počet riadkov a stĺpcov), formát, v akom je obrázok uložený (napr. RGB alebo HSV), počet bitov pre každú hodnotu, či je signed, koľko hodnôt je na jeden pixel a pod.
- samotnú maticu (resp. pointer na ňu), ktorá predstavuje 2D pole s hodnotami, ktoré reprezentujú farbu každého bodu

Aby sa optimalizovala práca s veľkým objemom dát, Mat reálne obsahuje len header a pointer na samotné dáta, čím umožňuje zdieľanie rovnakých dát medzi metódami. Pointer môže ukazovať len na časť veľkého obrázka.

Manipulácia s videom v OpenCV je zabezpečená prostredníctvom triedy VideoCapture. Táto trieda umožňuje načítavanie videa zo súboru alebo z video-streamu a jeho transformáciu na jednotlivé frame-y, ako aj samotné ovládanie kamery a jej atribútov.

Modul highgui (High-level GUI and Media I/O) umožňuje základnú komunikáciu s GUI. Umožňuje zobrazovanie obrázkov aj framov z videa, buď v pôvodnej alebo upravenej verzii (napr. po zvýraznení určitého objektu alebo jeho stredu).

OpenCV obsahuje veľké množstvo metód na spracovanie obrázkov, vrátane štrukturálnej analýzy obrázkov a identifikácie objektov, ktorá dokáže identifikovať objekty rôznych tvarov (napr. funkcia moments vie identifikovať polygonálne objekty do 3. rádu). fitEllipse nájde elipsu obkolesujúcu 2D objekt.

5.1.3 Knižnice pre tvorbu užívateľského prostredia

Pre tvorbu užívateľského prostredia bola zvolená knižnica Microsoft Foundation Class (MFC) library. Ide o C++ knižnicu od Microsoftu (msdn.microsoft.com/en-us/library/d06h2x6e.aspx) pre vývoj desktopových aplikácií pre Windows.

Samotná MFC obaľuje časti tried Windows API v C++, vrátane funkcionalít, ktoré jej umožňujú využívať štandardný aplikačný framework. Obsahuje triedy pre ovládanie mnohých Windows objektov, predefinovaných okien a štandardných kontrolných prvkov.

5.1.4 Knižnica pre export PDF

libHaru (<http://libharu.org>) je open-source knižnica pre C++, určená na generovanie PDF dokumentov. Pre účely vyvíjaného softvéru sú dôležité nasledovné funkcie:

- Do generovaného PDF sa dajú umiestniť obrázky, riadky textu, poprípade odkazy.
- Podporuje vkladanie obrázkov vo formáte PNG, resp. JPG.
- Podporuje ukladanie vygenerovaného PDF dokumentu na špecifikované miesto na disku.

Pre Windows je dostupná vo forme dvoch typov vývojových prostredí:

- Static library - žiadne spúšťateľné súbory nie sú potrebné
- Shared library (libhpdf.dll)

Inštalácia pre Windows:

```
unzip libharu-X.X.X.zip
cd libharu-X.X.X
```

Microsoft VC++ Compiler:

```
nmake -f script/Makefile.msvc (or Makefile.bcc32_dll)
```

Otestovanie:

```
nmake -f script/Makefile.msvc[_dll] demo
```

5.2 Architektúra aplikácie

Na tejto sekcii sa momentálne pracuje.

5.3 Diagramy

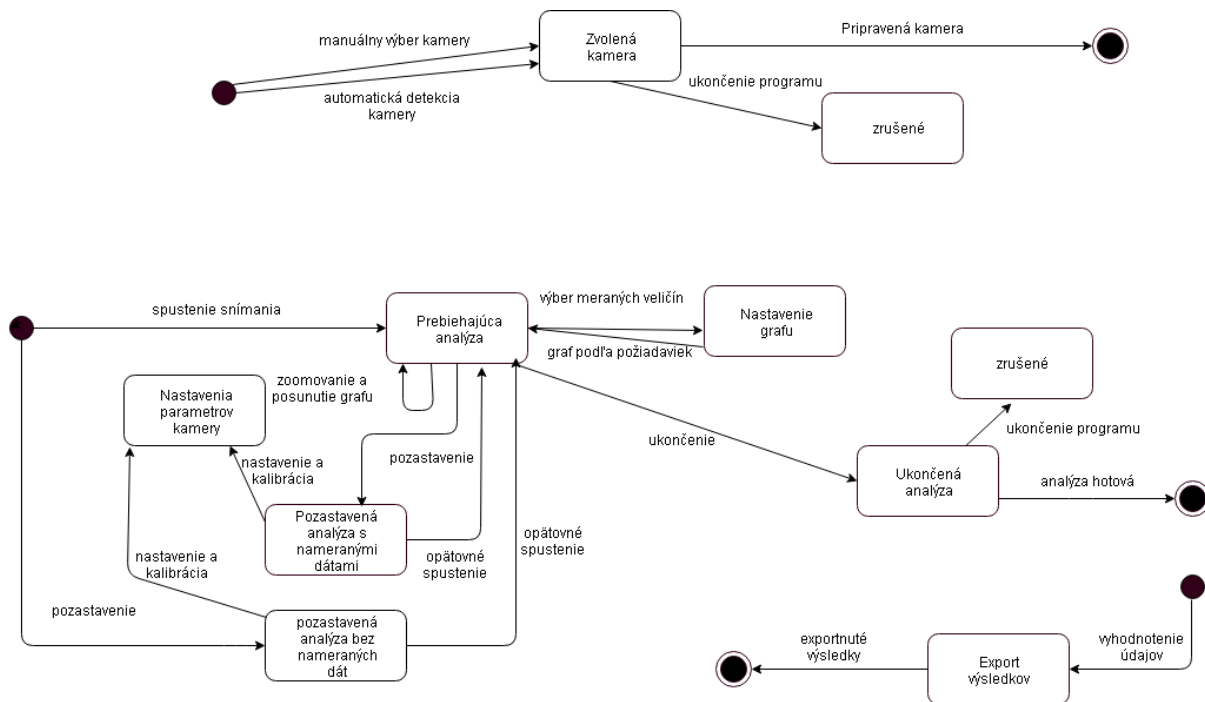
5.3.1 Triedny diagram

Na tejto sekcii sa momentálne pracuje.

5.3.2 Komponentový diagram

Na tejto sekcii sa momentálne pracuje.

5.3.3 Stavový diagram



obr 5.3.3 - Stavový diagram

5.3.4 Sekvenčný diagram

Na tejto sekcii sa momentálne pracuje.

5.4 Rozdelenie na časti (moduly)

Na tejto sekcii sa momentálne pracuje.

5.5 Cieľové prostredie nasadenia do prevádzky

Na tejto sekcii sa momentálne pracuje.