

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV FYZIKÁLNÍHO INŽENÝRSTVÍ  
FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF PHYSICAL ENGINEERING

# UŽIVATELSKÝ PROGRAM PRO ZOBRAZENÍ A ZPRACOVÁNÍ KVANTITATIVNÍ FÁZE Z DIGITÁLNÍHO HOLOGRAFICKÉHO MIKROSKOPU

USER PROGRAMME FOR IMAGING AND VISUALIZATION  
OF QUANTITATIVE PHASE FROM THE DIGITAL HOLOGRAPHIC MICROSCOPE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

LUKÁŠ HLADÍK

VEDOUCÍ PRÁCE  
SUPERVISOR  
BRNO 2010

Ing. HANA UHLÍŘOVÁ, Ph.D.

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav fyzikálního inženýrství

Akademický rok: 2009/2010

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

student(ka): Lukáš Hladík

který/která studuje v **bakalářském studijním programu**

obor: **Fyzikální inženýrství a nanotechnologie (3901R043)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Uživatelský program pro zobrazení a zpracování kvantitativní fáze z digitálního holografického mikroskopu**

v anglickém jazyce:

### **User programme for imaging and visualization of quantitative phase from the digital holographic microscope**

Stručná charakteristika problematiky úkolu:

Transmisní digitální holografický mikroskop (DHM) vyvinutý a sestrojený v Laboratoři optické mikroskopie na ÚFI VUT v Brně slouží především k výzkumu rakovinných buněk a jejich reakcí na vnější podněty. V tomto aplikačním projektu probíhá blízká spolupráce s biologickým pracovištěm, kde je často potřeba prezentovat a analyzovat naměřená data. Pro tento účel je potřeba vyvinout program pro prohlížení a zpracování obrázků z DHM metodou dynamických fázových diferencí s volitelným krokem difference a možností ukládání časosběrných pozorování.

Cíle bakalářské práce:

Má být vytvořen uživatelský program pro zpracování kvantitativní fáze, která je rekonstruována z hologramu a to ve smyslu dynamických fázových diferencí s barevným kódováním [1], [2]. Krok diferencí bude volitelný pro uživatele. Výstupem má být soubor typu .avi nebo jiného video-formátu. Program bude sloužit také pro prohlížení obrázků zpracované kvantitativní fázové informace z DHM. Program by měl být kompatibilní se softwarem pro ovládání kamery a DHM rekonstrukci obrazu pro případné zabudování.

Seznam odborné literatury:

- [1] Janečková H., Veselý P. and Chmelík R.: Proving Tumour Cells by Acute Nutritional/Energy Deprivation as a Survival Threat: A Task for Microscopy. Anticancer Research 29, 2339-2346, 2009.
- [2] Uhlířová H.: Mikroskopie časově proměnných biologických objektů. Brno, 2009. 72 s. Dizertační práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Vedoucí dizertační práce doc. RNDr. Radim Chmelík, Ph.D.
- [3] Janečková H.: Interferenční mikroskopie biologických vzorků. Brno, 2006. 51 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Vedoucí diplomové práce doc. RNDr. Radim Chmelík, Ph.D.

Vedoucí bakalářské práce: Ing. Hana Uhlířová, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 26.11.2009

L.S.

---

prof. RNDr. Tomáš Šíkola, CSc.  
Ředitel ústavu

---

prof. RNDr. Miroslav Doupovec, CSc.  
Děkan fakulty

## **Abstrakt**

Tato bakalářská práce se zabývá problematikou tvorby uživatelského počítačového programu určeného ke zpracování obrazů kvantitativního fázového kontrastu z digitálního holografického mikroskopu. Teoretická část práce obsahuje základní poznatky o získávání a vyhodnocování zpracovávaných obrazů včetně popisu jejich digitální reprezentace v počítači. V experimentální části jsou rozebrány použité techniky a algoritmy na kterých je program vystavěn. Tato část práce dále obsahuje popis grafického uživatelského rozhraní výsledného programu a zhodnocení vlivu různých uživatelských nastavení na vlastnosti výstupních souborů.

## **Summary**

This bachelor thesis deals with creation of an user computer program for processing of phase contrast images from a digital holographic microscope. The theoretical section of the thesis contains basic information about obtaining and evaluation of processed images, including a description of their digital representation in the computer. The experimental part deals with techniques and algorithms on which the program is built. This part also contains a description of the graphical user interface of the program and evaluate the impact of different user settings for output files properties.

## **klíčová slova**

zpracování obrazu, digitální holografický mikroskop, zobrazení buňky, dynamické fázové difference, suchá hmota buňky, optická mikroskopie

## **key words**

image processing, digital holographic microscope, cell imaging, dynamic phase differences, dry mass of cell, optical microscopy

HLADÍK, L.: *Uživatelský program pro zobrazení a zpracování kvantitativní fáze z digitálního holografického mikroskopu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 29 s. Vedoucí bakalářské práce Ing. Hana Uhlířová, Ph.D.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a použil pouze podklady uvedené v příloženém seznamu použité literatury.

26. května, 2010

Lukáš Hladík

Děkuji své vedoucí Ing. Haně Uhlířové Ph.D. za časté konzultace a cenné připomínky během tvorby této bakalářské práce.

Lukáš Hladík

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Formulace problému</b>	<b>8</b>
2.1	Digitální holografický mikroskop (DHM)	8
2.2	Konstrukce DHM	8
2.3	Rekonstrukce obrazu	9
2.4	Úpravy fázového zobrazení	9
2.5	Význam fáze při pozorování buněk	10
2.6	Dynamické fázové difference (DPD)	12
<b>3</b>	<b>Digitalizace obrazu</b>	<b>14</b>
3.1	Reprezentace obrazu	14
3.2	Barevný model RGB	14
3.3	Převod barevného obrazu na odstíny šedé	15
<b>4</b>	<b>Nástroje programového řešení</b>	<b>16</b>
4.1	Programovací jazyk C#	16
4.2	.NET	16
4.3	Prostředí .NET Framework	16
4.4	DirectShow	17
<b>5</b>	<b>Implementace programu</b>	<b>18</b>
5.1	Čtení bitmapových souborů	18
5.2	Výpočet dynamických fázových diferencí	18
5.3	Vytváření bitmapových souborů	20
5.4	Vytváření videosouborů	20
<b>6</b>	<b>Orientace v prostředí DPDView</b>	<b>21</b>
6.1	Instalace	21
6.2	Systémové požadavky	21
6.3	Základní obrazovka programu	22
6.4	Vstupní soubory	23
6.5	Výpočet DPD	24
6.6	Náhled videa	25
6.7	Uložení videa do souboru	26
<b>7</b>	<b>Závěr</b>	<b>27</b>
	<b>Seznam použitých symbolů</b>	<b>28</b>
	<b>Literatura</b>	<b>29</b>



# 1 Úvod

V Laboratoři optické mikroskopie na Ústavu fyzikálního inženýrství Fakulty strojního inženýrství VUT v Brně byl vyvinut a sestrojen transmisní digitální holografický mikroskop (DHM) viz obr. 1a, který slouží především k výzkumu rakovinných buněk a jejich reakcí na vnější podněty. V tomto aplikačním projektu probíhá blízká spolupráce s biologickým pracovištěm, kde je často potřeba prezentovat a analyzovat naměřená data. Pro tento účel je potřeba vyvinout program pro prohlížení a zpracování obrázků z DHM metodou dynamických fázových diferencí s volitelným krokem difference a možností ukládání časosběrných pozorování [1].

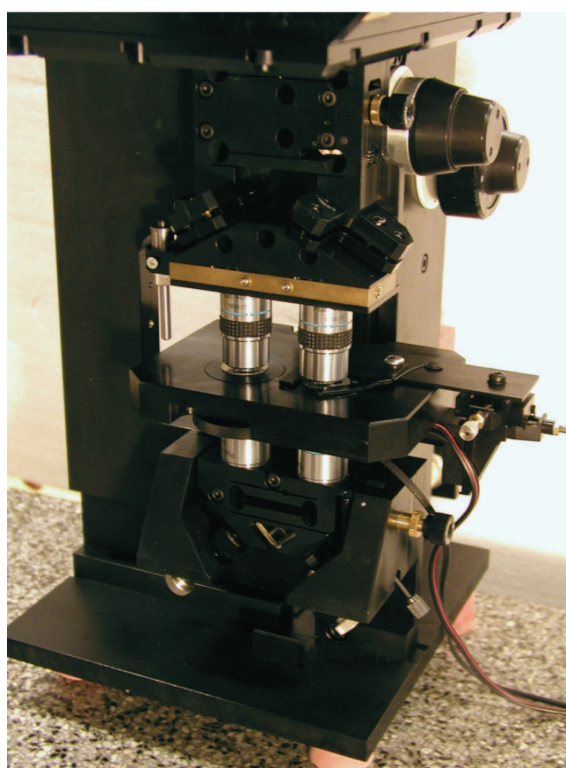
Cílem této bakalářské práce je tedy vytvoření uživatelského programu pro zpracování kvantitativního fázového kontrastu, který je rekonstruován z hologramu a to ve smyslu dynamických fázových diferencí s barevným kódováním [2]. Krok diferencí bude volitelný pro uživatele. Výstupem má být soubor typu AVI nebo jiného videoformátu. Program bude sloužit také pro prohlížení obrázků zpracované kvantitativní fázové informace z DHM. Program by měl být kompatibilní se software pro ovládání kamery a rekonstrukci obrazu z DHM pro případné zabudování.

První část práce se zabývá vysvětlením základních pojmů a principů spojených s holografickou mikroskopií. Další část představuje teoretický úvod do problematiky digitalizace obrazu, po které následuje část věnující se nástrojům programového řešení použitých při tvorbě programu. Realizační část práce pojednává o vlastním návrhu programu a algoritmech na kterých je vystavěn. V poslední části je popsáno grafické uživatelské rozhraní konečné implementace programu a způsob práce s ním včetně zhodnocení vlivů uživatelských nastavení.

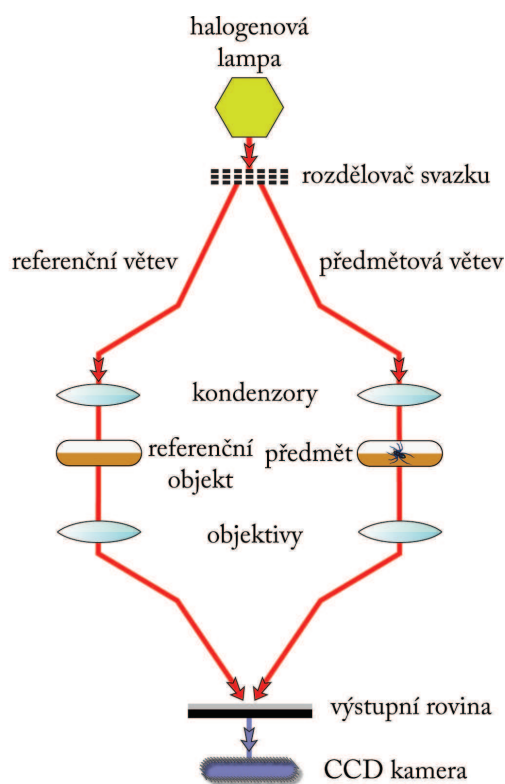
## 2 Formulace problému

### 2.1 Digitální holografický mikroskop (DHM)

Holografická mikroskopie umožňuje pozorovat průhledné nebo poloprůhledné objekty, které mohou být obtížně pozorovatelné pomocí klasických mikroskopů, kvůli jejich velmi nízké absorpci světla. Holografický mikroskop vytváří hologram, který je úplným záznamem vlny. Kromě intenzity lze tedy zobrazit i změny fázového kontrastu indukovaného vzorkem. Tato metoda vizualizace v případě buňky vypovídá o skutečném rozmístění její suché hmoty [2]. Změny fázového kontrastu lze v tomto DHM zaznamenat s vysokou vzorkovací frekvencí (12 Hz), a monitorovat tak dynamické děje v buňce. Proto se tento mikroskop hodí k časosběrným záznamům reakcí živých buněk na vnější podněty. Schéma tohoto mikroskopu zkonstruovaného a používaného v Laboratoři optické mikroskopie na Ústavu fyzikálního inženýrství Fakulty strojního inženýrství VUT je na obr. 1b.



a) DHM mikroskop, ÚFI, FSI, VUT.



b) Konstrukce DHM.

Obrázek 1: Digitální holografický mikroskop.

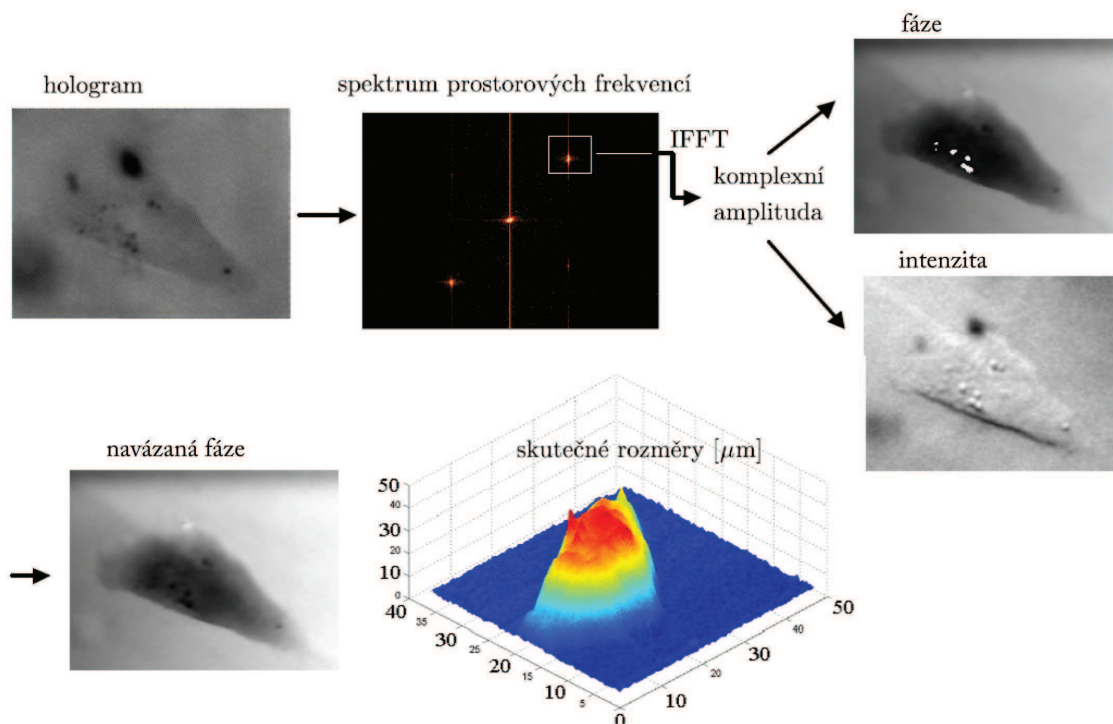
### 2.2 Konstrukce DHM

Zdrojem světla je halogenová lampa, která vyzařuje s velmi malou koherenční délkou a šířkou ([2], str. 13). Důsledkem použití světla s malou koherenční délkou a šířkou je eliminace koherenčního šumu v obraze a potlačení světla rozptýleného mimo rovinu zaostření. Jde o neinvazivní metodu pozorování, to znamená, že významným způsobem neovlivňuje biologické vzorky, které jsou předmětem pozorování a tak můžeme sledovat jejich reakce na vnější podněty. Používá se červené světlo protože je biologickými objekty

málo rozptylováno. Svazek světla ze zdroje je rozdělen na dva svazky, které procházejí skrz opticky ekvivalentní kondenzery. Objektová vlna prochází přes pozorovaný předmět, zatímco referenční vlna prochází skrz referenční objekt. V našem konkrétním případě je předmětem komůrka s médiem v němž se nacházejí pozorované buňky a referenčním objektem je pouze komůrka s médiem bez buněk. Rozdíl optických drah svazků je tedy v ideálním případě způsoben pouze průchodem světla buňkami. Obě větve poté procházejí přes opticky ekvivalentní objektivy, které zobrazí předmět do výstupní roviny mikroskopu, kde se setkávají svazky z obou větví a interferují. Vzniklý hologram ve výstupní rovině je následně snímán pomocí CCD kamery. Podrobnější popis DHM viz [3].

## 2.3 Rekonstrukce obrazu

Pořízené snímky (hologramy) jsou zpracovány fourierovskými metodami, čímž je zrekonstruována intenzita a fázový rozdíl světelné vlny procházející pozorovaným objektem vůči referenční vlně (dále už jen fáze). Hodnota fáze závisí na indexu lomu a tloušťce pozorovaného objektu. Pak tedy v místech, kde se mění index lomu nebo tloušťka, dochází i ke změně fáze.



Obrázek 2: Schéma rekonstrukce obrazu ([3], str. 27).

## 2.4 Úpravy fázového zobrazení

Z předcházejícího odstavce vyplývá, že z holografického zobrazení lze numericky pomocí fourierovských metod zrekonstruovat celou předmětovou vlnu — intenzitu a fázi. Zobrazení fáze je nutné dále upravovat ([2], str. 43).

Mezi prováděné úpravy patří v první řadě její navázání. To je nezbytné z toho důvodu, že objekty v předmětové části mohou způsobit změnu fázového rozdílu větší než  $2\pi$ . To

by vedlo k problémům při reprezentaci digitálního obrazu fáze, jehož oborem hodnot je pouze interval  $\langle 0, 2\pi \rangle$ . Bližší popis postupu navázání fáze je možno nalézt v [3], str. 25.

Výše zmíněnou úpravou získáme obraz fáze předmětové vlny, která vznikla součtem fázového posuvu při průchodu pozorovaným objektem a fázových vad vzniklých průchodem vlny optickou soustavou tvořící mikroskop nebo odchylky při numerickém výpočtu rekonstrukce zobrazení. Tyto chyby se mohou projevit modulací nebo nakloněním (prohnutím) fázového zobrazení pozorovaných objektů. Odstranění těchto chyb se provádí odečtením zrekonstruovaného zobrazení fáze bez preparátu (pozadí) od zobrazení fáze s preparátem.

Zde vyvstává další problém spojený s tím, že se během pozorování mění i toto pozadí. Důvodem těchto změn může být nestabilita justážních prvků mikroskopu nebo případné tepelné změny preparátu vlivem okolí. Řešení spočívá v proložení oblastí bez preparátů polynomiální plochou, která aproximuje pozadí. Ta je následně od daného fázového snímku odečtena. V současné situaci to znamená, že tyto korekce nelze provádět pro preparáty, které vyplňují celé zorné pole, jinak by nebylo možné v každém snímku fáze v dané sérii pozorování nalézt oblasti bez preparátu.

Rozsah hodnot fáze je v důsledku navazování a dalších odchylek v každém snímku jiný. Z toho plyne, že průměrná hodnota fáze pozadí má v každém snímku jinou hodnotu. Pokud bychom chtěli prohlížet časosběrné série po sobě následujících snímků byl by každý obrázek jiný. Z tohoto důvodu je nutné přiřadit stejnou hodnotu fáze ve všech snímcích, aby se hladina fáze pozadí vyrovnala. Řešení této problematiky se stalo předmětem několika bakalářských a diplomových prací.

## 2.5 Význam fáze při pozorování buněk

Při vyhodnocování výsledků pozorování nás zajímá především obraz fáze. Je to dáno tím, že při holografickém zobrazení je úroveň šedé v obrazu lineárně úměrná fázi, proto toto zobrazení můžeme nazvat jako kvantitativní fázový kontrast. Vztahy a závěry uvedené v tomto odstavci pocházejí ze ([2], str. 43).

Rozdíl optické dráhy  $\delta$  vlny způsobený průchodem látkou vůči vlně procházející vzduchem závisí na rozdílu indexů lomu látky. Index lomu látky a vzduchu je funkcí jejich hustoty  $\rho$ , což vyjadřuje Lorentzův–Lorenzův vztah ([2], str. 38). Buňku lze považovat za roztok proteinů, jehož hustota lineárně závisí na koncentraci proteinů. Proteiny jsou poměrně dost velké molekuly, ale i v případě značných variací v jejich složení nedojde k výrazné změně jejich indexu lomu. Za těchto podmínek platí vztah mezi indexem lomu buňky  $n_B$ , indexem lomu tekutiny  $n_R$ , ve které jsou proteiny rozpuštěny a koncentrací proteinů v buňce  $C$ , v této podobě

$$n_B - n_R = \alpha_s C. \quad (1)$$

Koncentrace  $C$  roztoku je obvykle měřena v jednotkách  $\text{g}/100 \text{ ml} = \text{g}/100 \text{ cm}^3$  a konstanta  $\alpha_s$ , uváděná většinou v jednotkách  $100 \text{ ml} = 100 \text{ cm}^3/\text{g}$ , je známa jako specifický refraktivní přírůstek. V rámci dizertační práce Ing. Hany Uhlířové Ph.D. byl shrnut vliv několika vnějších faktorů na hodnotu  $\alpha_s$ . Z tohoto souhrnu vyplynul poznatek, že hodnota  $\alpha_s$  je pro vodné roztoky proteinů přibližně konstantní.

Rozdíl optických drah  $\delta$  indukovaný buňkou, který je možné naměřit interferenčním mikroskopem je v případě ideálního zobrazení definován vztahem

$$\delta(x, y) = \int [n_B(x, y, z) - n_M] dz, \quad (2)$$

kde  $x, y$  označují souřadnice kolmé na optickou osu objektivu předmětové větve,  $z$  je souřadnice ve směru optické osy a  $n_B, n_M$  jsou příslušné indexy lomu buňky a média. Médium je označení pro okolní prostředí buňky. Dále předpokládáme, že v předmětové větvi DHM je buňka v médiu a v referenční větvi je opticky totožný předmět, ale bez buňky. Protože médium je svým složením velmi příbuzné rozpouštědлу v buňce, můžeme mu přisoudit stejný index lomu, tedy  $n_M = n_R$ . Veličina  $C$  ve vztahu (1) je v jednotkách g/100 ml, zavedeme  $C'$  tak, že  $[C'] = \text{g/ml}$ . Potom (1) lze přepsat do tvaru

$$n_B(x, y, z) = n_R + 100\alpha_s C'(x, y, z). \quad (3)$$

Jak bylo uvedeno výše  $C'$  udává koncentraci, tedy množství suché hmoty buňky<sup>1</sup> v gramech na mililitr, pak celkovou suchou hmotu buňky  $W$  v gramech udává integrál přes objem buňky  $\mathcal{V}$ .

$$W = \iiint_{\mathcal{V}} C'(x, y, z) dx dy dz. \quad (4)$$

Fázový rozdíl předmětové a referenční vlny  $\phi(x, y)$  získaný rekonstrukcí hologramu můžeme definovat jako

$$\phi(x, y) = k\delta(x, y) = \frac{2\pi}{\lambda}\delta(x, y), \quad (5)$$

kde  $k$  je vlnové číslo,  $\lambda$  je vlnová délka použitého světla.

Vztah mezi pozorovanou fází a suchou hmotou buňky můžeme pomocí vztahů (3), (4) a (5) vyjádřit jako

$$W = \frac{\lambda}{200\pi\alpha_s} \iint_S \phi(x, y) dx dy. \quad (6)$$

Ze vztahu (6) je zřejmé, že pozorovaná fáze buňky v DHM je úměrná plošné hustotě její suché hmoty.

Při přechodu na diskrétní rozložení fáze způsobené digitalizací obrazu můžeme psát

$$W(i, j) = \frac{\lambda}{200\pi\alpha_s} \iint_{S_{i,j}} \phi(i, j) dx dy, \quad (7)$$

kde  $i$  a  $j$  určují polohu pixelu ve fázovém zobrazení,  $W(i, j)$  označuje suchou hmotu odpovídající ploše buňky zobrazené na obrazový pixel  $i, j$  a  $S_{i,j}$  označuje plochu daného pixelu, přes kterou integrujeme.

Fáze  $\phi(i, j)$  je považována za konstantní na ploše pixelu fázového zobrazení. Protože každému pixelu zobrazení odpovídá stejná plocha průmětu preparátu do roviny  $x, y$ , kterou označíme  $\Delta S$ , fáze  $\phi(i, j)$  buňky na ploše odpovídající ploše průmětu do jednoho pixelu je přímo úměrná suché hmotě buňky  $W(i, j)$ , tedy

$$W(i, j) = \frac{\lambda\Delta S}{200\pi\alpha_s} \phi(i, j). \quad (8)$$

---

<sup>1</sup>Suchou hmotu tvoří látky jako proteiny, cukry, tuky a jejich sloučeniny, nukleové kyseliny a aminokyseliny, které jsou nositelem životních procesů v buňce ([2], str. 42).



## 2.6 Dynamické fázové difference (DPD)

Jedná se o metodu vyvinutou ke zpracování kvantitativního fázového kontrastu z DHM v rámci dizertační práce Ing. Hany Uhlířové Ph.D. Pro využití kvantitativní informace, kterou poskytuje fázová rekonstrukce z DHM, má význam jak její hodnota vůči nulové fázi definované v místech, kde fáze předmětové a referenční vlny mají stejnou hodnotu, tak také relativní hodnota vůči fázi v různých časových okamžicích během pozorování. Protože DHM umožňuje pozorování živých buněk, je možno zkoumat vliv prostředí na jejich motilitu (pohyblivost). Např. určitý typ buněk reaguje na nepříznivé prostředí tím, že se stáhne do sebe. Navíc pohyb buňky nepředstavuje pouze změnu tvaru jejího okraje, ale i vnitrobuněčné přemísťování hmoty uvnitř. Tato metoda tedy slouží pro zviditelnění právě těchto změn rozložení suché hmoty buňky během pozorování.

Metoda dynamických fázových diferencí je založena na odlišném zobrazení kladných a záporných rozdílů kvantitativního fázového kontrastu dvou snímků stejného preparátu ([2], str. 44). V případě, že  $\Phi_D > 0$ , jedná se o přírůstek suché hmoty buňky a je znázorněn v obrázku DPD červenou barvou tak, že jas barvy odpovídá hodnotě  $\Phi_D$ . Hodnoty  $\Phi_D < 0$  vyjadřující úbytek suché hmoty, jsou označeny ve stejném obrázku DPD zelenou barvou a jas barvy odpovídá hodnotě  $\Phi_D$ . Pokud je  $\Phi_D = 0$ , tak je jas barvy nulový (černá barva).

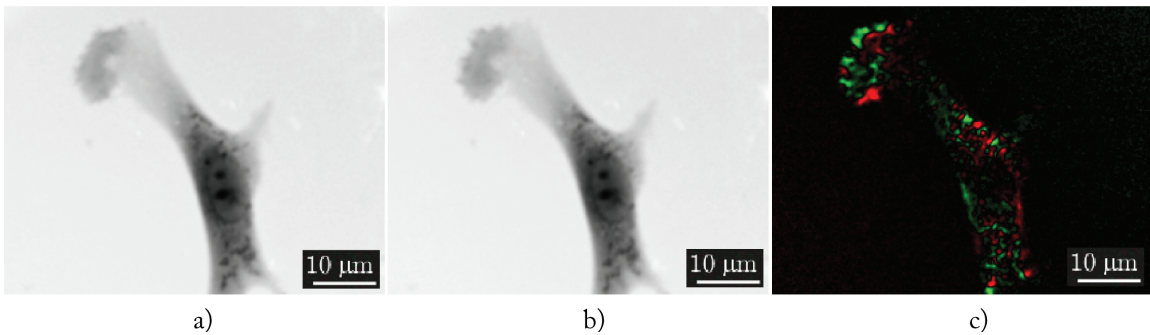
DPD můžeme vypočítat dvěma způsoby. První z nich se nazývá *postupné DPD* a počítá se jako rozdíl po sobě následujících snímků fáze, tedy

$$\Phi_{D1,q}(i, j) = \Phi_{q+1}(i, j) - \Phi_q(i, j), \quad q = 1, 2, \dots, w - 1, \quad (9)$$

kde  $\Phi$  je obraz fáze po numerických úpravách,  $w$  je počet snímků.

Druhý se nazývá *DPD vzhledem k počátku* a počítá se jako rozdíl příslušného a počátečního snímku podle

$$\Phi_{D2,q}(i, j) = \Phi_{q+1}(i, j) - \Phi_1(i, j), \quad q = 1, 2, \dots, w - 1. \quad (10)$$



Obrázek 3: Kvantitativní fázový kontrast buňky v čase  $t$  na obr. a,  $t + 30$  s na obr. b a DPD mezi těmito snímky na obr. c, z něhož je patrné zvýraznění změny rozložení suché hmoty buňky.

Tímto způsobem lze vizuálně vyhodnotit okamžité změny rozmístění suché hmoty buňky během reakce (postupné DPD) nebo vývoj reakce (DPD vzhledem počátku). Vizualizace dynamických fázových diferencí a jejich barevné označení umožňuje zvýraznit

pohyby drobných částí buňky, které jsou pouhým pozorováním kvantitativního fázového kontrastu velmi těžko postřehnutelné. Tato skutečnost je demonstrována na obr. 3. Zatímco rozdíl rozložení hmoty buňky mezi obr. 3a a 3b v časech  $t$  a  $t+30$  s je obtížně postřehnutelný, DPD na obr. 3c dává jasnou představu o tom, k jakým změnám v časovém intervalu 30 sekund došlo.

## 3 Digitalizace obrazu

Základním kamenem pro zpracování obrazu je vlastní získání obrazu reálného světa a jeho převod do digitální formy vhodné pro uložení a další zpracování v počítači ([4], str. 12). To, co vidíme v reálném světě, lze chápat jako nějakou spojitou veličinu — obrazovou funkci (představme si hodnoty jasu). Počítače nám neumožňují zaznamenat a uložit veličinu jako spojitou, nýbrž diskretní. Proces přechodu od spojitě veličiny k diskretní se nazývá digitalizace. V této souvislosti se objevuje pojem Pixel - jenž vznikl ze zkrácení anglického sousloví *picture element*. Pixel představuje jeden obrazový bod nesoucí informaci o hodnotě jasu nebo v případě barevného obrazu o barvě. Jako snímače obrazu jsou dnes nejvíce rozšířeny obrazové senzory nazývané dle technologie a způsobu transportu informace jako CCD (*charged coupled device*) nebo CMOS (*complementary metal-oxide semiconductor*).

### 3.1 Reprezentace obrazu

Při dalším zpracování obrazu potřebujeme obraz a jasové hodnoty v něm vhodným způsobem reprezentovat ([4], str. 25). Definice se v různých disciplínách od sebe liší. Formálně lze obraz definovat jako spojitou funkci dvou proměnných  $f(x, y)$ , kde  $f$  označuje hodnotu jasu a  $(x, y)$  jsou souřadnice určující pozici v obraze. Předpokládáme, že obraz, se kterým reálně budeme pracovat, má omezené rozměry

$$\langle x_{\min}, x_{\max} \rangle \times \langle y_{\min}, y_{\max} \rangle . \quad (11)$$

Hodnotou obrazové funkce může být jedno číslo, které reprezentuje hodnotu jasu v obraze na pozici  $(x, y)$ . V případě barevného obrazu to ale může být trojice čísel  $f_R(x, y), f_G(x, y), f_B(x, y)$ , které reprezentují hodnotu jasu jednotlivých barevných složek v modelu RGB.

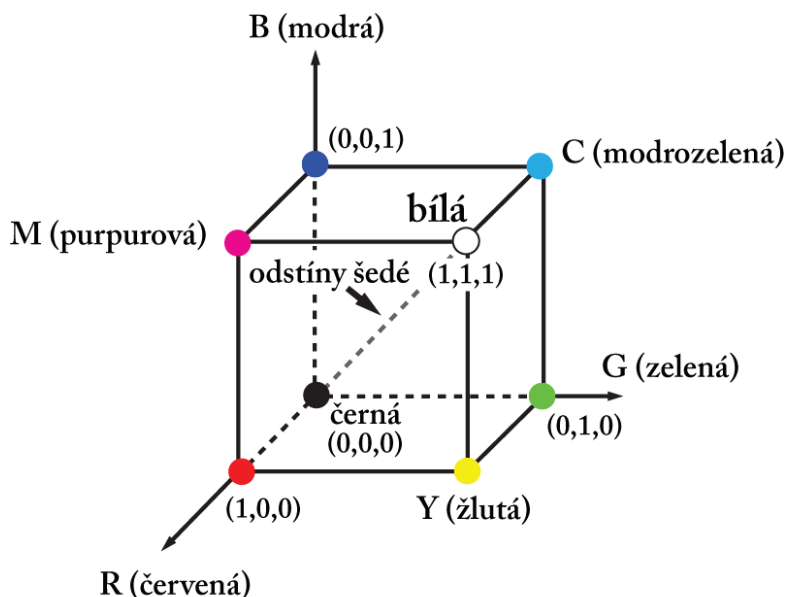
V praxi je obor hodnot omezen technickými možnostmi. Funkční hodnoty  $f$  jsou po digitalizaci diskretní, souřadnice  $(x, y)$  jsou celočíselné. Pro reprezentaci šedotónového obrazu používáme dvojrozměrnou matici  $G$  s hodnotami jasu  $f(x, y)$ . Pro reprezentaci barevného obrazu používáme trojrozměrné pole  $f(x, y, barva)$ , kde *barva* je konstanta označující o kterou barvu se jedná (0 = červená, 1 = zelená, 2 = modrá). Pro určení přesného odstínu barvy byly vytvořeny různé barevné modely.

### 3.2 Barevný model RGB

Tento popis barev je využíván v zobrazovacích zařízeních, kdy jeden viditelný bod (pixel) tvoří tři velmi blízko umístěné body barevných složek Red (červené, R), Green (zelené, G), Blue (modré, B) ([4], str. 19). Jednotlivým složkám byly přiřazeny vlnové délky: Red = 700 nm, Green = 546,1 nm, Blue = 435,8 nm. Složením jednotlivých barevných složek RGB v patřičném poměru získáme výslednou barvu. Proto říkáme, že tento model barev je aditivní - tedy pro hodnoty (R = 0, G = 0, B = 0) je výsledná barva černá. Se zvyšující se hodnotou se barvy „přidávají“, až po maximální hodnotu (R = 1, G = 1, B = 1) kdy je výsledná barva bílá (viz obr. 4). Platí, že pokud jsou hodnoty jednotlivých složek stejné, jedná se o odstín šedi. V počítačové grafice se většinou uvažuje rozsah hodnot v celých číslech v rozmezí 0–255. V této souvislosti se hovoří i o hloubce barev, která udává číselné rozmezí pro jednotlivé složky. Barevná hloubka 8 bitů je tvořena výběrem (tabulkou) 256 definovaných barev. Výběr barev je buď definován použitým



standardem nebo musí být definován vzhledem k danému obrazu. Při  $3 \times 8$  bitech (na každou barvu 8), tak lze vyjádřit asi 16,7 milionu barev. U použití 24bitové hloubky hovoříme již o „TrueColor“ obrazech, což odpovídá tomu, co je možno vidět v reálném světě.



Obrázek 4: Reprezentace barevného prostoru RGB.

### 3.3 Převod barevného obrazu na odstíny šedé

Barevný obraz můžeme převést na obraz v odstínech šedi, kde odstíny šedi jsou vyjádřeny hodnotami jasu. Jedním z důvodů je následné zpracování, kde není třeba zkoumat barevné složky separátně. Protože je lidské oko různě citlivé na jednotlivé složky barev RGB, nelze použít prosté sečtení jednotlivých složek, ale pro převod se používá váhových koeficientů

$$f = 0,299R + 0,587G + 0,114B, \quad (12)$$

kde  $f$  je výsledná úroveň jasu v šedotónovém obraze a  $R$ ,  $G$  a  $B$  mají význam úrovní jasu jednotlivých složek daného bodu v barevném obraze.

## 4 Nástroje programového řešení

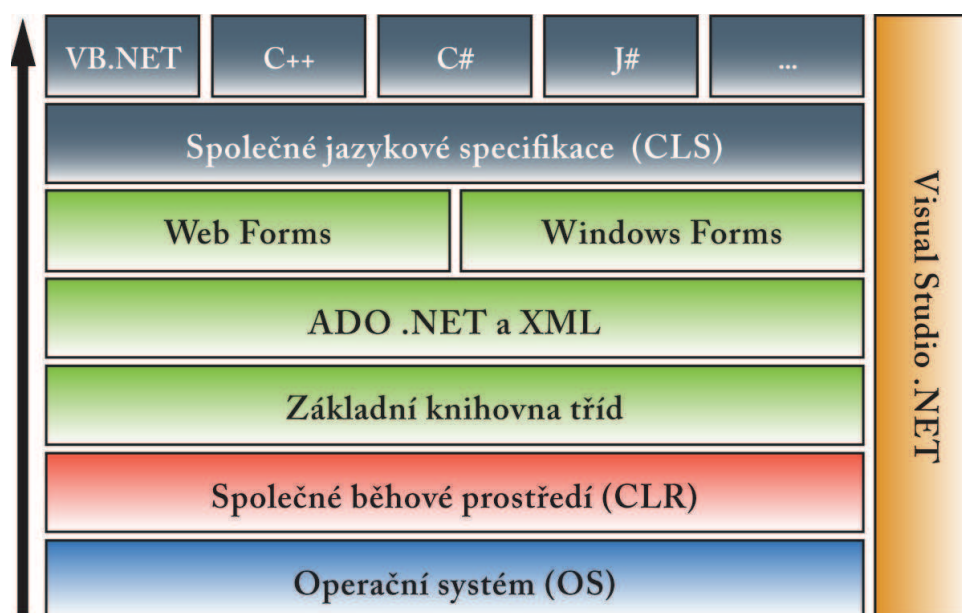
K vývoji programu byl zvolen jazyk C#, kvůli jeho orientaci na platformu .NET Framework, která umožňuje snadné a rychlé vytváření grafických uživatelských prostředí dle současných standardů.

### 4.1 Programovací jazyk C#

C# je výkonný, ale přitom jednoduchý, jazyk zaměřený především na vývojáře aplikací na platformě .NET Framework ([5], str. 21). Zdědil velmi mnoho dobrých vlastností jazyků C++ a Microsoft Visual Basic, ale jen málo z jejich nesrovnalostí a anachronismů, takže výsledkem je čistší a logičtější jazyk. Můžeme se na něj dívat jako na zjednodušenou a čistě objektovou verzi programovacího jazyka C++. Lze ho využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení (PDA a mobilní telefony) atd.

### 4.2 .NET

Architektura .NET („dotnet“ podle anglického *dot NET* = *tečka NET*, *NET* pochází z *network*, síť) byla navržena firmou Microsoft k propojení všech prvků, které mohou být propojené pomocí software - lidí, počítačů, staveb a dalších zařízení [4]. K tomu využívá tzv. NET kompatibilní programovací jazyky, mezi které patří Visual Basic.NET, C++, C#, JScript.NET, ASP.NET a další.



Obrázek 5: Celkový pohled na architekturu .NET [7].

### 4.3 Prostředí .NET Framework

K tomu, aby programy využívající architektury .NET mohly být spuštěny na uživatelském počítači musí být nainstalován balík .NET Framework, který je jedním z mnoha vývojových platforem určených pro tvorbu aplikací v této architektuře. Tento balík se

skládá z několika součástí, mezi nimiž velkou roli hraje tzv. Základní knihovna tříd (*Base Class Library*) ([6], str. 28). Ta obsahuje již předprogramovaná řešení nejčastějších nízkoúrovňových programátorských problémů např. funkce pro ukládání a otevírání různých druhů dat ve výše uvedených .NET jazycích. Jinak řečeno, .NET aplikace může být napsána v libovolném podporovaném programovacím jazyce a bez dalšího vynaloženého úsilí vývojáře na převod do jiných programovacích jazyků. Druhým nejdůležitějším modulem je tzv. Společné běhové prostředí (*Common Language Runtime, CLR*). Toto prostředí zajišťuje běh programů přeložených z různých programovacích jazyků do vlastního mezijazyka. CLR umožňuje jejich vzájemnou spolupráci, takže různé součásti programu mohou být napsány v různých programovacích jazycích. Stará se o automatickou správu paměti, o řízení doby života objektů a o další funkce nezbytné pro běh programů v tomto prostředí.

## 4.4 DirectShow

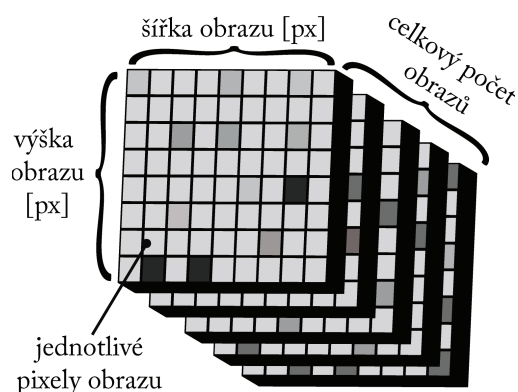
DirectShow je multimediální vývojová platforma, vytvořená firmou Microsoft pro vývojáře software [9]. Je určena k vykonávání různorodých operací s multimediálními soubory nebo datovými toky. V programu je k DirectShow přistupováno ve formě dynamické sdílené knihovny (Interop.QuartzTypeLib.dll), která je součástí jeho instalačního archívu.

## 5 Implementace programu

Tato kapitola popisuje některé detaily implementace a ukazuje dosažené výsledky. Dále popisuje řešení načítání a tvorby bitmapových obrazů, tvorbu videosouborů a především DPD, jejichž výpočetní algoritmus je rozebrán podrobněji.

### 5.1 Čtení bitmapových souborů

Rekonstruované obrazy fáze jsou barevné bitmapové<sup>2</sup> soubory. K takovému typu souboru je možno v jazyce C# přistupovat pomocí instance třídy `System.Drawing.Bitmap`, jejímž argumentem je řetězec obsahující cestu k souboru obrazu. K procházení přes jednotlivé pixely bitmapy je nutné její data nejdříve „uzamknout“ v paměti pomocí metody `Bitmap.LockBits`. V dalším kroku pomocí cyklu procházíme jednotlivými pixely bitmapy a v každé iteraci zjišťujeme hodnoty složek RGB metodou `Color.FromArgb`. Protože je pro naše další zpracování důležitá jen hodnota jasu pixelu v obrazu fáze, vypočteme si ji ze vztahu (12) a výslednou hodnotu uložíme na odpovídající pozici  $x, y$  do třírozměrného pole ve tvaru (výška obrazu  $\times$  šířka obrazu  $\times$  celkový počet obrazů ke zpracování), jak je znázorněno na obr. 6.



Obrázek 6: Zvolený způsob ukládání obrazových dat v paměti počítače.

### 5.2 Výpočet dynamických fázových diferencí

Algoritmus výpočtu DPD je principiálně popsán v odstavci 2.6. Všechna obrazová data jsou uložena v třírozměrném poli *Pha* ve tvaru uvedeném v odstavci 5.1. Tento tvar zajišťuje snadný přístup k požadovaným obrazovým souborům. Řídícím cyklem výpočtu DPD je cyklus procházející přes pořadí jednotlivých obrazů tak, jak jsou za sebou uloženy v proměnné *Pha*. K pohybu po jednotlivých pixelech daného obrazu slouží dva další do sebe zanořené cykly, jejichž řídicí proměnné nabývají hodnot od nuly do maximálního rozměru příslušného obrazu. Na této nejnižší úrovni zanoření je realizována podmínka, ověřující uživatelem vybraný typ DPD. Podle této podmínky se řízení programu rozhodne, který z bloků kódu v těle podmínky bude dál zpracován. Zápis hodnot rozdílů jasu probíhá do proměnné typu pole *PoleDPD* o velikosti (výška obrazu  $\times$  šířka obrazu  $\times$  barevná složka). Programový zápis výpočtu rozdílů jasu příslušných obrazových bodů v případě určování postupných DPD vypadá zjednodušeně takto:

<sup>2</sup>Bitmapa = mapa bitů, z anglického *bitmap*, jeden ze způsobů reprezentace obrazu v paměti počítače [4].

```

PoleDPD[, ,červevá] = Pha[, ,číslo obrazu+krok]-Pha[, ,číslo obrazu];
PoleDPD[, ,zelená] = Pha[, ,číslo obrazu]-Pha[, ,číslo obrazu+krok];
PoleDPD[, ,modrá] = 0;

```

A programový zápis výpočtu DPD vůči počátku následovně:

```

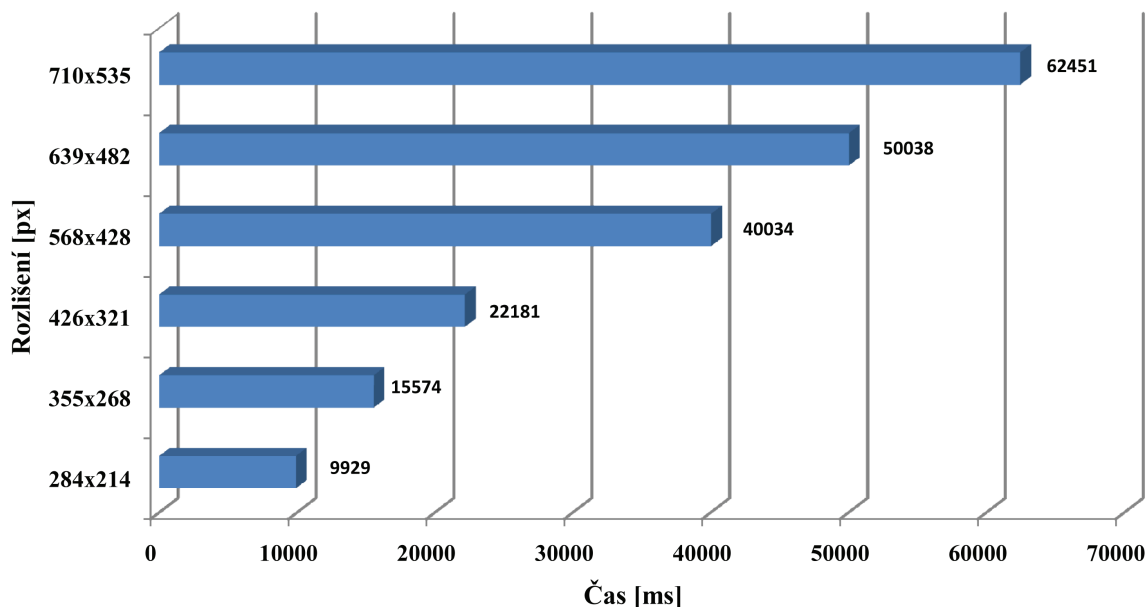
PoleDPD[, ,červevá] = Pha[, ,číslo obrazu+krok]-Pha[, ,první obraz];
PoleDPD[, ,zelená] = Pha[, ,první obraz]-Pha[, ,číslo obrazu+krok];
PoleDPD[, ,modrá] = 0;

```

Jak je patrné ze zápisů uvedených výše, modrá složka je pouze inicializována nulovou hodnotou, což je nutné pro pozdější vytvoření barevné bitmapy. Z důvodu nízké citlivosti lidského oka na tuto barvu, ji ke zvýraznění DPD nepoužíváme.

V této chvíli převzme iniciativu rozhodovací část cyklu, která přiřadí vypočteným rozdílům jasu barevný význam. Cyklus prochází přes všechny hodnoty rozdílů jasu a přes jednotlivé barevné složky pole *PoleDPD* a aplikuje na ně předem definované podmínky. Tímto postupem dojde k nadefinování jednoho bitmapového obrazu fázových diferencí s barevným kódováním. Výsledek této úpravy je zobrazen uživateli v podobě jedné položky v panelu DPD v hlavním okně programu. Proměnná *PoleDPD* je vynulována, řídící proměnná cyklu (pořadí vstupních obrazů) je inkrementována o jedničku a celý výpočet začne znova. K ukončení celého výpočtu dochází v okamžiku, kdy řídící proměnná cyklu nabude větší hodnoty, než je celkový počet obrazů vybraných uživatelem ke zpracování.

### Závislost doby výpočtu DPD na rozlišení obrazů



Obrázek 7: Závislost doby výpočtu DPD na rozlišení obrazů. Jak je z tohoto grafu patrné, tato závislost je téměř lineární. Celkový počet zpracovávaných obrazů: 189. Barevná hloubka obrazů: 24 bitů.

Z hlediska výpočetní náročnosti je výpočet DPD nejnáročnější částí celého programu. Doba potřebná ke zpracování se odvíjí od mnoha faktorů. Předně je to celkový počet obrazů ke zpracování a jejich rozlišení (více bodů v obrazu, jenž je třeba zpracovat),

a s tím související délka kroku DPD (čím kratší, tím více obrazů ke zpracování). Celková výpočetní doba DPD dále také závisí na hardwarovém vybavení počítače na kterém je výpočet realizován. Na obr. 7, je možno shlédnout srovnání doby výpočtu DPD pro různá rozlišení obrazů. Jak je z tohoto grafu patrné, závislost doby výpočtu na rozlišení obrazů je téměř lineární.

### 5.3 Vytváření bitmapových souborů

Tvorbou bitmapových souborů se v našem případě zabýváme v okamžiku ukládání DPD. V první řadě požadujeme, aby každá vypočtená DPD byla na disku volně přístupná v podobě bitmapového souboru pro případné další zpracování. V druhé řadě účelem našeho programu je i prohlížení vypočtených DPD, a to by ve stávajícím řešení jinak, než převodem na bitmapu nebylo možné.

Vytvoříme instanci třídy `System.Drawing.Bitmap`, která bude reprezentovat náš nový obraz. Tato třída vyžaduje specifikaci parametrů nové bitmapy — výšku, šířku a počet bitů na barvu. Metodou `Bitmap.LockBits` zpřístupníme jednotlivé bity pro zápis a pomocí cyklu můžeme přistupovat k jednotlivým pixelům bitmapy. Metodou `Color.ToArgb` přiřazujeme hodnoty složek RGB jednotlivých pixelů na odpovídajících pozicích  $x, y$ , které získáme např. výpočtem DPD. Takto vytvořená bitmapa je zatím pouze v paměti počítače a kdybychom program vypnuli přišli bychom o ni. Proto je výhodné vytvořený obraz uložit metodou `Bitmap.Save` do samostatného obrazového souboru na disk v některém ze standardně podporovaných grafických formátů.

### 5.4 Vytváření videosouborů

Tvorba videa ze série obrazů představuje komplexnější programátorský problém, jehož vlastní řešení přesahuje rámec této bakalářské práce, proto je výhodnější použít některé z již předpřipravených řešení. K řešení tohoto problému byla zvolena externí knihovna **AVI<sup>3</sup>file format library** a mírně pozměněný algoritmus `AviWriter`, jehož autorkou je softwarová vývojářka Corinna John a který může být volně šířen a používán.

Algoritmus tvorby videa v našem programu lze s využitím těchto externích knihoven popsat následujícím způsobem. Vytvoříme instanci třídy `AviWriter`, která otevře datový proud a pomocí cyklu vkládáme metodou `AddFrame` jednotlivé obrazy, tak jak za sebou mají ve videu následovat. Všechny obrazy z nichž je video tvořeno musí mít stejné rozměry, aby nedošlo k přetečení obrazového zásobníku. Ošetření proti rozdílným rozměrům vstupních obrazů je provedeno přímo ve vstupním výběrovém dialogu, viz odstavec 6.4. Proto se tímto problémem v dalších částech programu nemusíme dále zabývat. Po načtení všech obrazů datový proud uzavřeme a pomocí třídy `AVISaveOptions` nastavíme parametry výsledného videosouboru (počet snímků za sekundu, kompresní poměr), který bude následně zapsán na disk ve formě videosouboru s příponou AVI.

---

<sup>3</sup>Audio Video Interleave — multimediální kontejner, který rozděluje data souboru do bloků, každý blok je identifikován pomocí speciální značky [10].

## 6 Orientace v prostředí DPDView

Program DPDView implementuje metodu dynamických fázových diferencí popsanou v odstavci 2.6. Je tedy určen ke zpracování obrazů již zrekonstruované a navázané fáze z DHM. Umožňuje libovolně volit krok DPD, zobrazit a prohlížet zdrojové i již upravené obrazy z dané série pozorování. Uživatel dále může vytvořit náhled videa ze zpracovaných souborů a následně tento náhled uložit do videosouboru ve standardním formátu. Hlavním rysem programu je jednoduché a intuitivní grafické uživatelské prostředí, jehož ovládání bude vysvětleno v této kapitole.

### 6.1 Instalace

Program DPDView sám o sobě instalaci nevyžaduje. Rozbalením instalačního balíčku vznikne adresář DPDView, který je možné nakopírovat na libovolné místo v počítači či počítačové síti. Uvnitř tohoto adresáře se nachází doprovodné knihovny a spustitelný soubor **DPDView.exe**, kterým se program spouští. Program žádným způsobem nemodifikuje systémové soubory, adresáře a registry ani žádným jiným způsobem neovlivňuje systémová nastavení. **Instalaci vyžaduje** běhové prostředí (*runtime*) **.NET Framework 3.5**, pokud není v počítači již nainstalováno. Aktuální verze 3.5 (v době psaní této práce) je pro majitele operačního systému Windows k dispozici zdarma jako samostatná komponenta, která se do systému doinstaluje. Bývá také šířena na CD či DVD různých počítačových časopisů, jako součást různého software a lze ji také stáhnout samostatně nebo prostřednictvím služby Windows Update. Prostředí .NET Framework je navrženo tak, aby program napsaný pro nižší verzi fungoval i s vyšší. Zjednodušeně lze tedy říci, že stačí na počítač nainstalovat nejnovější verzi prostředí .NET Framework.

### 6.2 Systémové požadavky

Program byl napsán ve vývojovém prostředí Visual Studio 2008 primárně jako aplikace pro operační systém Microsoft Windows. Podporovány jsou architektury

- x86,
- x64,
- ia64 (Windows Server 2008).

Mezi podporované verze operačního systému Windows patří

- Windows XP,
- Windows Vista,
- Windows 7,
- Windows Server 2003/2008.

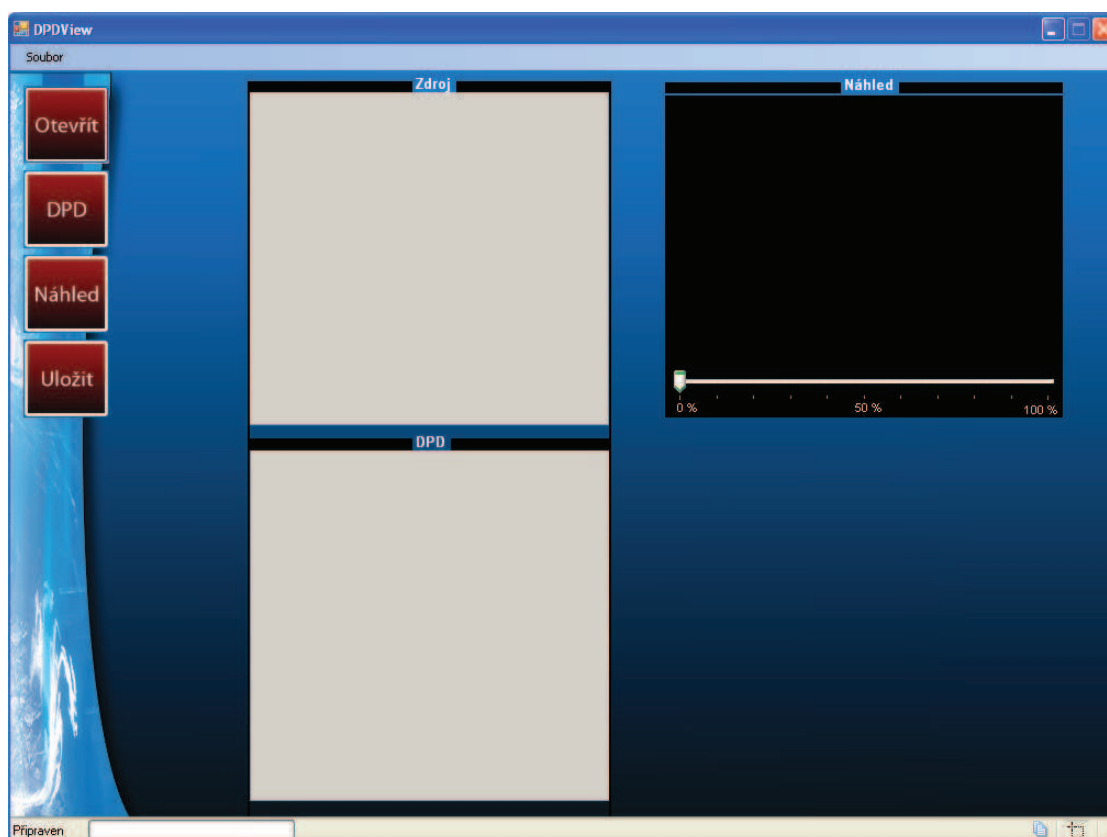
Podpora pro unixové platformy (Linux, BSD, Mac OS X, Solaris, AIX) není oficiálně umožněna, avšak existují komunitní nástroje (DotGNU, WINE), které umožňují spustit tento program i tam.



Minimální požadavky na hardwarové vybavení počítače se odvíjí od hardwarových požadavků pro instalaci prostředí .NET Framework 3.5, tedy 400 MHz CPU, 96 MB RAM, 1024×768 high-color 32bit monitor a alespoň 500 MB volného místa na pevném disku počítače (ovšem spustitelný soubor programu dosahuje velikosti pouze 1 MB). Posledním požadavkem programu je právo pro zápis do svého adresáře a podadresářů.

### 6.3 Základní obrazovka programu

Po spuštění programu uživatele uvítá hlavní okno celého programu, kde především z důvodů zrychlení a zjednodušení práce s tímto programem probíhá 90% uživatelem vykonávaných úkonů. Jak je patrné z obr. 8 je toto okno rozděleno do několika částí. V horní

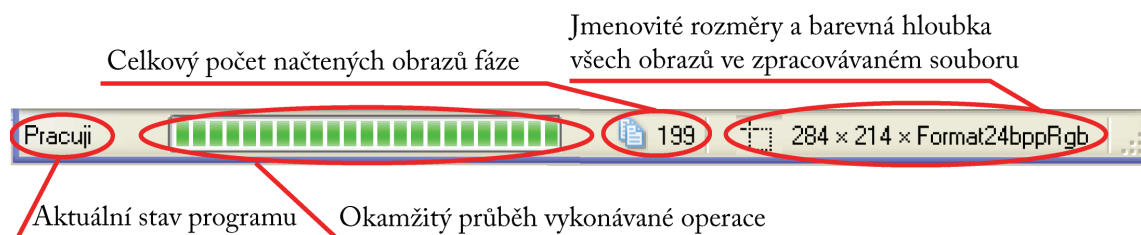


Obrázek 8: Základní obrazovka programu DPDView bez načtených dat.

části se nachází programová lišta. Na levém okraji centrální plochy je umístěn oddělovací pruh s tlačítky umožňující rychlý přístup k volbám nabízených funkcí programu. Centrální plocha hlavního okna programu obsahuje dva posuvné panely sloužící k zobrazení zdrojových obrazů fáze a vypočtených DPD. Dále se zde nachází okno pro přehrávání náhledu DPD s posuvníkem umožňujícím libovolné časové posunutí přehrávaného náhledu. Tento posuvník zároveň slouží jako indikátor průběhu přehrávání.

U spodního okraje programu je v celé délce hlavního okna umístěn stavový řádek (komponenta Status Bar), který poskytuje základní informace o stavu programu a datech se kterými se právě pracuje viz obr. 9.





Obrázek 9: Popis jednotlivých položek stavového řádku.

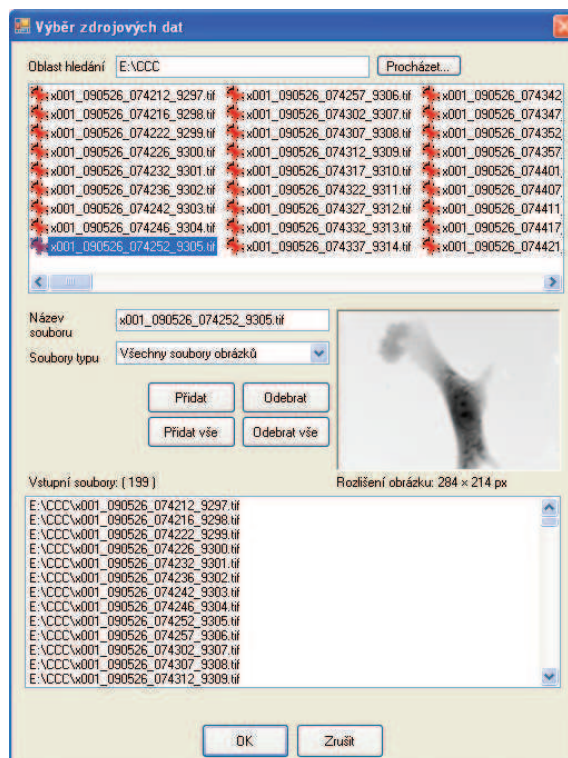
## 6.4 Vstupní soubory

DPDView umožňuje načítat 24bitové barevné obrazové soubory standardních formátů, jmenovitě JPG, TIFF, PNG, GIF, BMP. Vstupní soubory lze načíst tlačítkem **Otevřít**, případně v horní liště přes menu **Soubor**→**Otevřít**. Pro pohodlnou správu vstupních souborů, byl navržen jednoduchý souborový dialog, kde se po výběru příslušného zdrojového adresáře z adresářové struktury pevného disku zobrazí jednotlivé soubory v horním zobrazovacím poli.

Pro přidání jednoho nebo více souborů do seznamu souborů ke zpracování slouží ovládací tlačítka umístěná v prostřední části dialogu. K rychlé orientaci mezi soubory slouží okno náhledu, v němž se zobrazuje právě uživatelem označený obrazový soubor. Zobrazení náhledu je ještě doplněno o výpis rozměrů právě uživatelem označeného obrazu. Toto se jeví jako velmi praktické z toho důvodu, že uživatel musí svým výběrem vybrat pouze takovou sadu obrazů, ve které mají všechny obrazy stejné rozměry. Tato podmínka vyplývá z odstavce 5.2, kde k určení polohy odpovídajícího pixelu ve dvou po sobě následujících obrazech fáze používáme souřadnice vztažené ke stejnému počátku souřadných os.

Možné situaci, kdy by musel uživatel ručně kontrolovat rozměry pro celou sérii měření, jsme předešli implementováním kontrolní funkce, která při kliknutí na potvrzovací tlačítko **OK**, zkontroluje daný výběr a obrazy, které se rozměrově liší od ostatních zvýrazní červenou barvou. Uživatel je následně vyzván, aby svůj výběr změnil tak, že ve spodním výběrovém okně nebude žádný soubor označen jako rozměrově odlišný. Tímto ve své podstatě jednoduchým opatřením jsme vyřešili veškeré další problémy, které by mohly vzniknout v důsledku různých rozměrů jednotlivých obrazů ve vybrané sérii obrazů ke zpracování, ať již v případě výpočtu DPD, viz odstavec 5.2 či vytváření videa, viz odstavec 5.4.

V této části je ještě nezbytně nutné zmínit podmínku na abecední řazení názvů jednotlivých obrazů ve zpracovávaném souboru měření. Celá tato bakalářská práce je založena na zpracování **časosběrných** pozorování fázového kontrastu buňky. Proto zde hraje časová návaznost velkou roli a jednou z možností, jak tuto návaznost zajistit, je použití abecedně navazujících názvů vstupních obrazů. Další možností určení pořadí obrazů určených ke zpracování by bylo pomocí systémových funkcí zjišťovat poslední datum změny daného obrazu (případně čas vytvoření obrazu), ovšem tento způsob ztrácí význam v okamžiku, kdy si uvědomíme, že dochází k softwarové rekonstrukci obrazu fáze a následným úpravám popsaných v odstavci 2.4, při kterých je čas úpravy všech obrazů fáze změněn na téměř stejnou hodnotu (např. během 1 sekundy počítač zpracuje 5 souborů naráz).



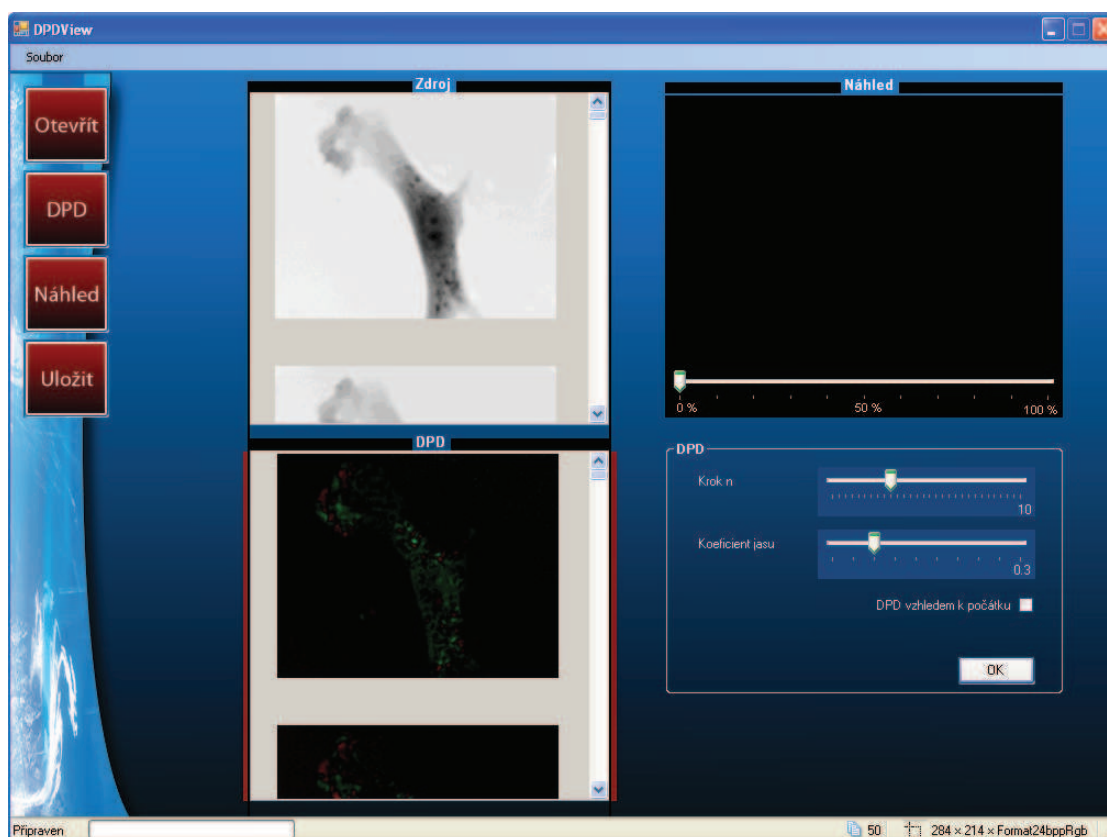
Obrázek 10: Navržený souborový dialog pro jednoduchou správu vstupních souborů.

## 6.5 Výpočet DPD

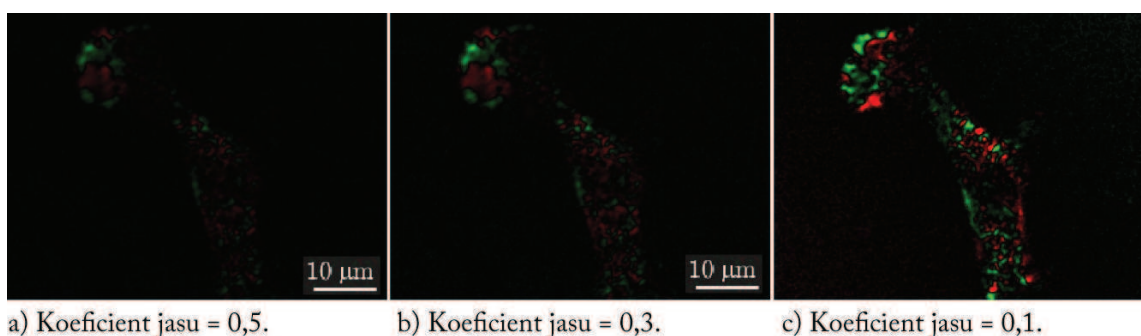
V tomto odstavci se dostáváme k ústřední funkci celého DPDView, a tou je samotný výpočet dynamických fázových diferencí. Příslušná konfigurační část hlavního okna programu se aktivuje kliknutím na tlačítko **DPD** nebo pomocí nabídky **Soubor**→**DPD**. Zde je nutné upozornit na to, že tyto volby se stanou aktivní pouze v případě provedení úspěšného výběru vstupních souborů popsaného v odstavci 6.4. Tímto postupem předcházíme situaci, kdy by bylo možné ihned po startu programu nechat počítat DPD, aniž by byly vybrány obrazy ke zpracování, což by mohlo mít za následek pád programu. Jedinou možností, jak upozornit uživatele, že není něco v pořádku, by bylo chybové hlášení, což by neprospělo uživatelskému komfortu, na který je při návrhu tohoto programu kladen velký důraz.

V bíle orámovaném konfiguračním dialogu **DPD** je možné měnit typ DPD. Zatržením boxu **DPD vzhledem k počátku** vynutíme jen tento typ DPD, viz odstavec 2.6. Výsledné programem vypočtené DPD jsou zobrazeny ve sloupci **DPD** v hlavním okně programu. Jednotlivé obrazové položky tohoto seznamu je možno nalézt v pracovním adresáři programu ve složce PhaDn, kde  $n$  je délka uživatelem zvoleného kroku. Na obr. 11 je zobrazeno základní okno programu s načtenými daty.

V konfiguračním dialogu je dále umožněno změnit koeficient jasu. Tento koeficient udává číslo kterým je vydělena příslušná hodnota příslušné barevné složky každého pixelu v obrazu. Tímto koeficientem se dá snadno přímo v prostředí DPDView měnit barevný jas výsledných DPD a ve svém důsledku i barevný jas náhledu videa vygenerovaného z těchto zesvětlených (ztmavených) DPD. Na obr. 12 je srovnání vlivu změny koeficientu jasu na jas vypočtených DPD.



Obrázek 11: Základní obrazovka programu DPDView s načtenými obrazy.



Obrázek 12: Srovnání změn jasu obrazů vlivem změny koeficientu jasu.

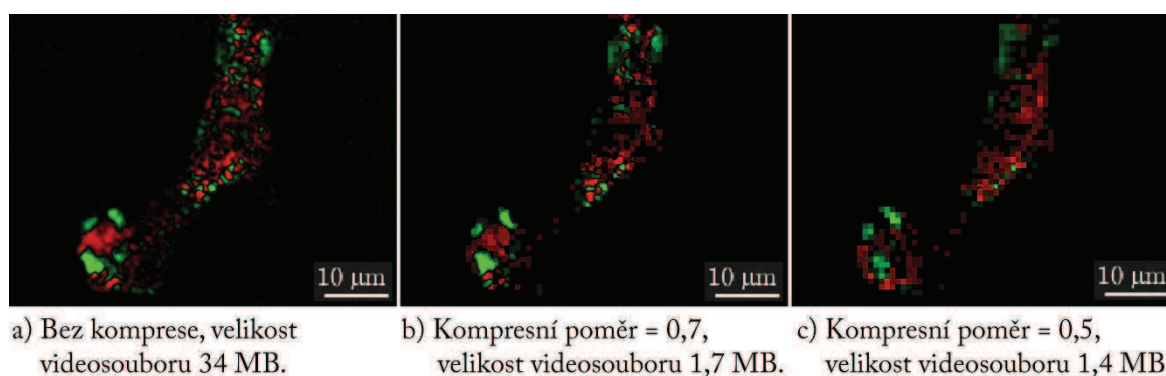
## 6.6 Náhled videa

Další důležitou funkcí programu je tvorba náhledu videa, který poskytuje přímou informaci o změnách suché hmoty buňky v čase. Přehrávací okno náhledu lze aktivovat opět v levém menu tlačítkem **Náhled** nebo přes nabídku **Soubor**→**Náhled**. Po kliknutí na některý z těchto ovládacích prvků se zaktualizuje konfigurační dialog v pravém dolním rohu hlavního okna programu, kde je možné změnit počet snímků za sekundu z kterých bude video náhledu vytvořeno, či si nechat náhled přehrát v externím okně. Přehrávání v externím okně může být výhodné pokud jsou změny v DPD malé a je třeba si náhled zvětšit. Pro zjednodušení a zrychlení práce s programem je umožněno měnit parametry náhledu v libovolném okamžiku i během přehrávání náhledu, není třeba čekat až se celý náhled videa přehraje až do konce. K okamžitému vymazání přehrávacího okna náhledu

a uvolnění přehrávaného náhledu videa z paměti slouží tlačítko **Stop**.

## 6.7 Uložení videa do souboru

Protože realizace tohoto programu vyplynula z potřeby prezentace výsledků pozorování biologických objektů na vzdáleném biologickém pracovišti, je umožněno ukládání výsledků DPD do videoformátu AVI a sice bezkompresním a kompresním<sup>4</sup>. Do ukládacího dialogu se je možno dostat tlačítkem **Uložit** nebo přes nabídku **Soubor**→**Uložit**. Při velkém počtu zpracovávaných souborů, by se mohly vyskytnout problémy s velikostí výsledného videosouboru, jehož velikost by se při použití bezkompresní metody mohla pohybovat v řádech stovek megabytů. Proto je při ukládání možné využít volby uložení do komprimovaného videoformátu, který ovšem na rozdíl od bezkompresního formátu snižuje vizuální kvalitu takto uloženého videosouboru, jak je patrné na obr. 13. Součástí této volby je i možnost uživatelského nastavení kompresního poměru, tedy nalezení kompromisu mezi kvalitou a velikostí. Toto se jeví jako velmi praktické při zasílání náhledu emailem nebo přenášení náhledů výsledků pozorování na přenosných discích i pro případy velmi rozsáhlých souborů obsahující stovky obrazů, které by mohli v maximální obrazové kvalitě dosahovat velikosti v řádech gigabytů.



Obrázek 13: Srovnání změn vizuální kvality videosouboru vlivem změny kompresního poměru.

<sup>4</sup>Kompresí dat je proces překódování vstupního (zdrojového) toku dat na jiný datový tok (výstupní, komprimovaný), který má menší velikost. Kompresí dat se dosáhne redukci nebo odstraněním opakování v datech [11].

## 7 Závěr

V rámci předkládané bakalářské práce byl vytvořen počítačový program, který zpracovává obraz fáze získané z transmisního digitálního holografického mikroskopu. V průběhu vývoje tohoto programu byly přidány některé další doplňující funkce nad rámec zadání. Program v podobě, v jaké byl odevzdán s touto bakalářskou prací umožňuje

- načítat 24bitové obrazy ve formátech TIFF, JPEG, PNG, BMP a GIF,
- načítat řádově stovky obrazů zrekonstruované fáze buňky,
- vypočítat a zobrazit výsledek obou typů dynamických fázových diferencí,
- nastavit délku kroku obou DPD,
- nastavit koeficient jasu DPD,
- ukládat obrazy DPD na disk ve formátech TIFF, JPEG, PNG, BMP, GIF,
- generovat a přehrávat náhled videa z vypočtených DPD,
- nastavit snímkovací frekvenci vytvářeného náhledu videa,
- ukládat náhled videa na disk ve formátu AVI, s možností nastavení vizuální kvality (a tím i velikosti videosouboru) ukládaného videa.

Program byl úspěšně testován na několika sadách souborů zrekonstruované fáze buňky a ukázal se jako vhodný nástroj pro výpočet a zobrazení dynamických fázových diferencí.

Během vývoje tohoto programu se objevily další možnosti na rozšíření, ale buď z časových nebo koncepčních důvodů nebyly implementovány. Mezi první rozšíření by patřila změna načítacího algoritmu obrazů, kvůli zmenšení nároků na operační paměť počítače při práci s velkým počtem obrazů. Ze současného přístupu, kdy program nejprve všechna vstupní data načte k „sobě“ do paměti a po zbytek času už k nim jen přistupuje. Místo toho by bylo vhodnější načítat data jen dvou právě odečítaných obrazů a ty pak následně hned po provedení požadované operace vypouštět z paměti kvůli uvolnění místa v paměti pro další dvojici.

Další rozšíření by se týkalo umožnění načítání 16bitových černobílých obrazů, což by vedlo k přesnějšímu (jemnějšímu) rozlišení barevného kódování výsledků DPD.

Ke zlepšení uživatelského pohodlí by mohlo přispět ukládání daného pozorování jako *projektu* — souboru, který by program otevřel a ve kterém by byla uložena veškerá nastavení a cesty k pracovním adresářům, které souvisí s jedním konkrétním pozorováním.

## Seznam použitých symbolů

Symbol	Význam	Jednotky
$\lambda$	obecná vlnová délka	m
$k$	vlnový vektor	rad m <sup>-1</sup>
$t$	čas	s
$\phi$	fáze z DHM, tedy fázový rozdíl mezi vlnou z předmětové a referenční větve	rad
$\Phi$	kvantitativní fáze po numerických úpravách	rad
$n$	index lomu	
$\delta$	rozdíl optické dráhy	m
$\varrho$	hustota	kg/m <sup>3</sup>
$\alpha_s$	specifický refraktivní přírůstek	m <sup>3</sup> /kg
$C$	koncentrace roztoku	kg/m <sup>3</sup>
$W$	suchá hmota	kg
$S_{i,j}$	plocha pixelu	px <sup>2</sup>
$\Phi_D$	dynamická fázová difference	rad
$\Phi_{D1}$	postupná dynamická fázová difference	rad
$\Phi_{D2}$	dynamická fázová difference vzhledem k počátku	rad
$x, y, z$	souřadnice pixelu v obraze	px
$f$	hodnota jasů pixelu	



## Literatura

- [1] JANEČKOVÁ H., VESELÝ P. and CHMELÍK R.: *Proving Tumour Cells by Acute Nutritional/Energy Deprivation as a Survival Threat: A Task for Microscopy*. Anti-cancer Research 29, 2339-2346, 2009.
- [2] UHLÍŘOVÁ, H.: *Mikroskopie časově proměnných biologických objektů*, dizertační práce, ÚFI FSI VUT v Brně, 2009.
- [3] JANEČKOVÁ, H.: *Interferenční mikroskopie biologických vzorků*, diplomová práce, ÚFI FSI VUT v Brně, 2006.
- [4] DOBEŠ, M.: *Zpracování obrazu a algoritmy v C#*, Nakladatelství BEN, Praha, 2008.
- [5] SHARP, J.: *Microsoft Visual C# 2008 Krok za krokem*, Nakladatelství Computer Press, a.s., Brno, 2008.
- [6] VIRIUS, M.: *C# pro zelenáče*, Nakladatelství a vydavatelství Neocortex, Praha, 2002.
- [7] <http://www.microsoft.com/NET/>, dne 20. dubna, 2010.
- [8] MAREŠ, A.: *1001 tipů a triků pro C#*, Nakladatelství Computer Press, a.s., Brno, 2008.
- [9] <http://en.wikipedia.org/wiki/DirectShow>, dne 14. května, 2010.
- [10] [http://cs.wikipedia.org/wiki/Audio\\_Video\\_Interleave](http://cs.wikipedia.org/wiki/Audio_Video_Interleave), dne 17. května, 2010.
- [11] SALOMON D.: *Data Compression, the Complete Reference*, 3rd edition, Springer, 2004.