

---

# MemSkel - uživatelský manuál

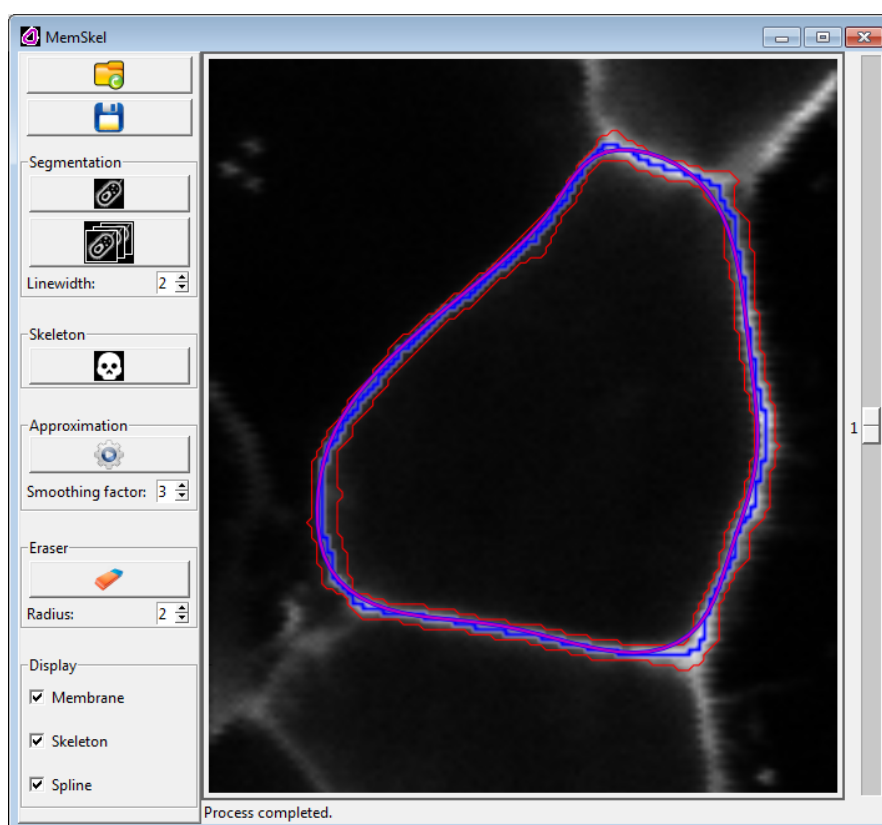
---

Tomáš Ryba  
tryba@kky.zcu.cz

30. června 2015

# 1 Úvod

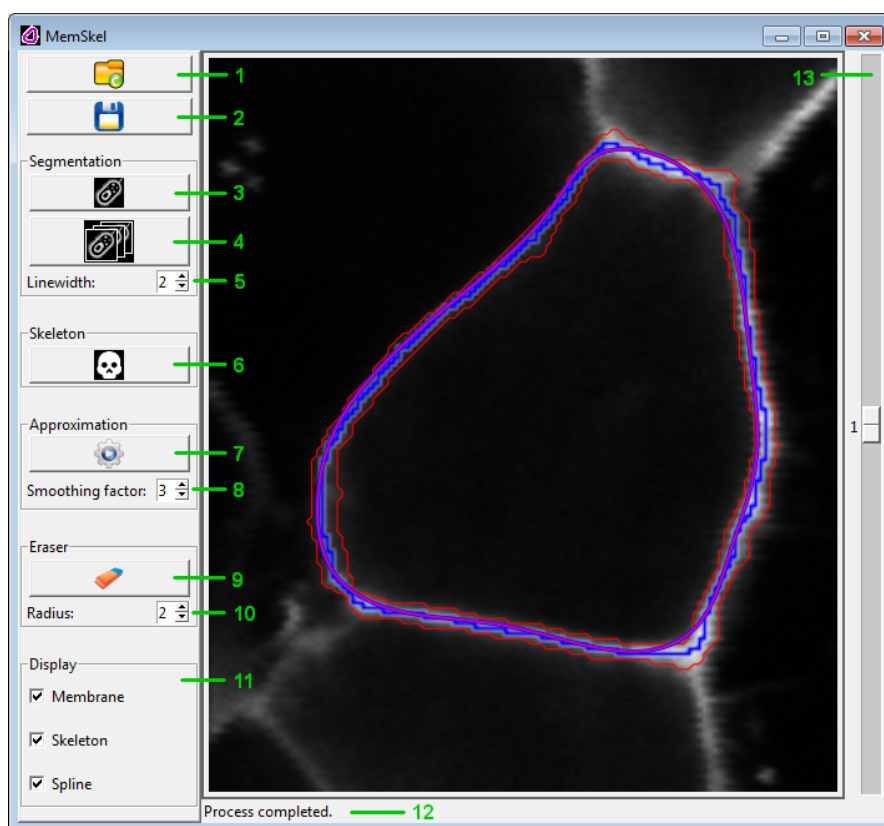
Program MemSkel (**M**embrane **S**keletonization) je určen pro segmentaci buněčné stěny. Jedná se o interaktivní metodu segmentace obrazu, která ke svému běhu potřebuje alespoň jednu výchozí pozici v obraze. Tu uživatel do obrazu zadá pomocí myši. Takto označené body (pixely) se nazývají seedy (z anglického výrazu *seed points*). Výsledek segmentace je možný manuálně upravit nástrojem guma či přidáním dalších seedů. Jakmile je uživatel s výsledkem spokojen, je možné spustit skeletonizaci, která najde střednici segmentované membrány. Je nutné, aby byl výsledek segmentace uzavřenou oblastí, jinak proces skeletonizace selže viz níže. Výsledný skelet je možné aproximovat pomocí b-spliny s proměnnou mírou vyhlazení.



Obrázek 1: Hlavní okno programu.

## 2 Grafické uživatelské rozhraní

Aplikace je rozdělena do dvou hlavních částí: nástrojová lišta, vizualizace. V následující části jsou popsány všechny hlavní prvky grafického uživatelského rozhraní, které jsou očíslovány na obr. 2.



Obrázek 2: Hlavní okno programu s očíslovanými částmi.

V následujícím výčtu je uveden krátký popis jednotlivých částí:

1. Otevřít soubor ... otevře dialog pro načtení dat ze souboru.
2. Uložit ... otevře dialog pro uložení výsledků práce.
3. Segmentace jednoho snímku ... v případě dat obsahujících více než jeden snímek (frame, page), provede segmentaci pouze aktuálního snímku.
4. Segmentace stacku ... v případě dat obsahujících více než jeden snímek (frame, page), provede segmentaci celého stacku (všech framů).
5. Linewidth ... spinbox pro změnu šířky čáry, používané při označování membrány.
6. Skeletonization ... spustí skeletonizaci masky membrány a vypočítá její střednici.
7. Approximation ... aproximuje střednici masky membrány s mírou vyhlazení danou následujícím parametrem.

8. Smoothing factor ...spinbox pro určení míry s jakou bude aproximována střednice masky membrány.
9. Eraser ...spustí nástroj guma pro mazání masky membrány.
10. Radius ...spinbox pro změnu velikosti gumy.
11. Display ...checkboxy pro zapnutí/vypnutí vizualizace příslušných prvků.
12. Status bar ...zobrazuje informace o běhu programu.
13. Slider ...umožňuje změnu snímku načtených dat.

## 3 Používání aplikace

Následující část bude obsahovat odkazy na jednotlivé komponenty grafického rozhraní, které jsou očíslovány na obr. 2. Odkazy budou ve formě kroužku s příslušným číslem uvnitř, např. odkaz na tlačítko pro otevření souboru má tvar ①.

### 3.1 Spuštění a ukončení

Editor se spouští souborem *memSkel\_main.exe* a ukončuje se kliknutím na červený křížek v pravém horním rohu okna.

### 3.2 Načtení dat

Stiskem tlačítka ① se otevře dialog pro otevření souboru. Předpokládá se, že vstupní data budou ve formátu TIFF. Je nutné zajistit, aby byl formát kompletní, především aby obsahoval metainformace o velikosti jednotlivých snímků. Pokud bude problem s načtením dat, „přeuložení“ obrázku v prohlížeči/editoru obrázků může pomoci. Např. program *IrfanView* při možnosti *Save as...* metainformace vytvoří. Jinou možností, která v tomto případě může pomoci, je uložit data jako 8-bitová (na výstup programu tato změna nemá vliv).

Jakmile jsou data úspěšně načtena, zobrazí se v editoru.

### 3.3 Segmentace membrány

Jak již bylo řečeno, pro segmentaci membrány potřebuje algoritmus znát alespoň jeden výchozí bod (seed). Po stisku tlačítka ③ resp. ④ se spustí editační mód, ve kterém uživatel tahem myši (levé tlačítko stisknuté) označí seedy. Stiskem tlačítka enter se spustí se výpočetní algoritmus, po jehož doběhnutí má uživatel možnost podle potřeby domalovat další seedy a stiskem enteru spustit algoritmus znovu.

Po ukončení označování a stisku enteru, tzn. algoritmus doběhl a nebyly přidány další seedy, se editační mód ukončí. Poté je možné výsledek upravit pomocí nástroje guma, viz níže.

Během editačního módu je možné měnit šířku čáry pomocí spinboxu ⑤ nebo kolečkem myši.

Stiskem tlačítka ④ se spustí segmentace celého stacku, tzn. všech snímků. V tomto případě začne algoritmus po ukončení segmentace aktuálního snímku (včetně možnosti přidání seedů) výsledek „rozšiřovat“ do dalších snímků. K tomu je zapotřebí již skeleton prvního snímku, jeho výpočet se tedy spustí automaticky. Editace masky membrány je možná po doběhnutí algoritmu do konce.

### 3.4 Nástroj guma

Pomocí tohoto nástroje (tlačítko ⑨ nebo tlačítko *e*) je možné editovat binární masku membrány. Aktivita tohoto nástroje je indikována modrým kolečkem kolem kurzoru, jehož velikost koresponduje s velikostí okolí, které je editováno. Poloměr nástroje guma je možný měnit pomocí spinboxu ⑩ nebo kolečkem myši, pokud je nástroj aktivní. Po stisku levého tlačítka myši se spustí mazací režim, který vymaže všechny pixely v masce membrány, které jsou lokalizovány v oblasti dané kolečkem kolem kurzoru. Samozřejmostí je i možnost tažení myši. Aktivace mazacího režimu je indikována zbarvením tohoto kolečka do červena, viz obr. 3.

### 3.5 Skeletonizace

Jakmile je určena maska membrány, je dalším krokem určení její střednice (středové osy). Toho je docíleno stiskem tlačítka ⑥. Pro správnou funkčnost je nutné, aby byla membrána uzavřená, jinak nebude algoritmus fungovat, viz obr. 4.

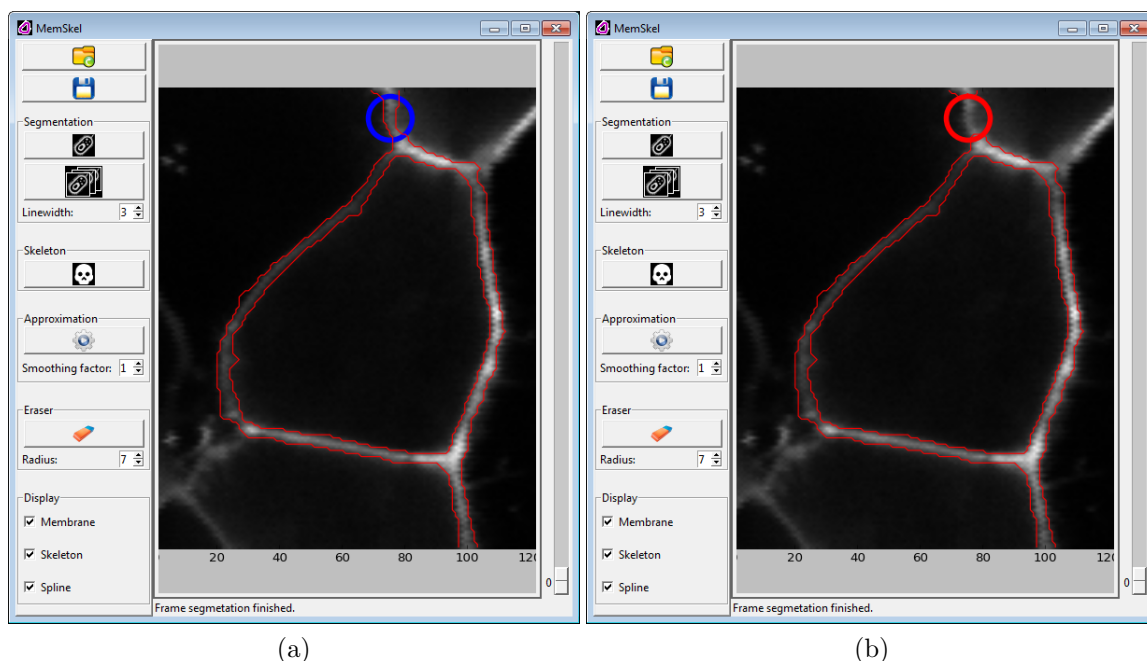
### 3.6 Aproximace

Po určení střednice je možné provést její aproximaci pomocí b-spliny stiskem tlačítka ⑦. Současně je možné měnit i míru vyhlazení pomocí spinboxu ⑧. Demonstrace tohoto nástroje je zobrazena na následujícím obrázku 5.

### 3.7 Uložení výsledků

Po dokončení práce je možné uložit výsledky pomocí tlačítka ②. Program vytvoří dva soubory:

1. binární obrázek představující výslednou masku membrány,
2. textový soubor popisující aproximační splinu (uzly).



Obrázek 3: Demonstrace nástroje guma. Na obr. 3a je znázorněn stav, kdy je nástroj guma spuštěn, obr. 3b pak znázorňuje stav při stisku levého tlačítka myši.

## 4 Ostatní

### 4.1 Status bar

Ve status baru (12), se zobrazují hlavní informace o činnosti editoru. Mezi ně patří informace o úspěšném/neúspěšném načtení dat, průběhu algoritmů, úspěšném/neúspěšném uložení výsledků apod. Při segmentaci více-snímkových dat se v pravé části zobrazí i tzv. progress bar, který udává procentuální stav výpočtu.

### 4.2 Log file

Při kritické chybě editoru, při níž dojde k jeho ukončení, se objeví okno informující uživatele o zápisu dané chyby do tzv. log souboru. Tento soubor má název *memskel\_main.exe.log* a obsahuje bližší informaci o místě v programu, které vyvolalo pád editoru. Tento soubor je důležitý pro případné hlášení dané chyby vyvojářům.

## 5 Spuštění programu ze zdrojových dat

Tato kapitola popisuje postup instalace Pythonu a balíčků, které jsou v programu využívány. Toho je zapotřebí v případě spouštění programu ze zdrojových dat.

### 5.1 Potřebné balíky

V této sekci jsou vyjmenovány všechny balíky, které program využívá a tudíž potřebuje ke svému běhu. Uvedena je i verze daných balíčků.

- Python 2.7.5
- numpy-MKL-1.7.1
- scipy-0.12.0
- matplotlib-1.1.1 (s novějšími verzemi program nefunguje - nezachytává klávesy), před instalací nutno nainstalovat následující podmiňující balíky:
  - python-dateutil-1.5
  - pytz-2013b
  - pyparsing-1.5.7
- pylibtiff-0.3.0.dev82
- Pillow-2.0.0

### 5.2 Pořadí instalace balíčků

Při samotné instalaci je zapotřebí dodržovat určité pořadí, ve kterém jsou balíky do Pythonu přidávány:

1. python-2.7.5.msi ... instaluje samotný jazyk Python; v tomto kroku nezáleží na tom, jestli se nainstaluje 32 nebo 64 bitová verze, je ale nutné další balíky volit ve stejné bitové verzi,
2. numpy-MKL-1.7.1.win32-py2.7.exe ... výpočetní jádro pythonu, práce s vícerozměrnými poli apod,
3. scipy-0.12.0.win32-py2.7.exe ... balík obsahující mmj. metody pro aproximaci,
4. python-dateutil-1.5.win32-py2.7.exe ... prerekvizita pro balík matplotlib,
5. pytz-2013b.win32-py2.7.exe ... prerekvizita pro balík matplotlib,

6. `pyparsing-1.5.7.win32-py2.7.exe` ...prerekvizita pro balík `matplotlib`,
7. `matplotlib-1.1.1.win32-py2.7.exe` ... vizualizace, novější verze balíku nefungují,
8. `pylibtiff-0.3.0.dev82.win32-py2.7.exe` ... práce s obrázky typu TIFF,
9. `Pillow-2.0.0.win32-py2.7.exe` ... práce s obrázky, načítání apod,
10. `setuptools-0.7.4.win32-py2.7.exe` ... instalace balíků ze zdrojových kódů,
11. `pymorph-0.96` ... morfologické úpravy obrázku; instalace viz níže.

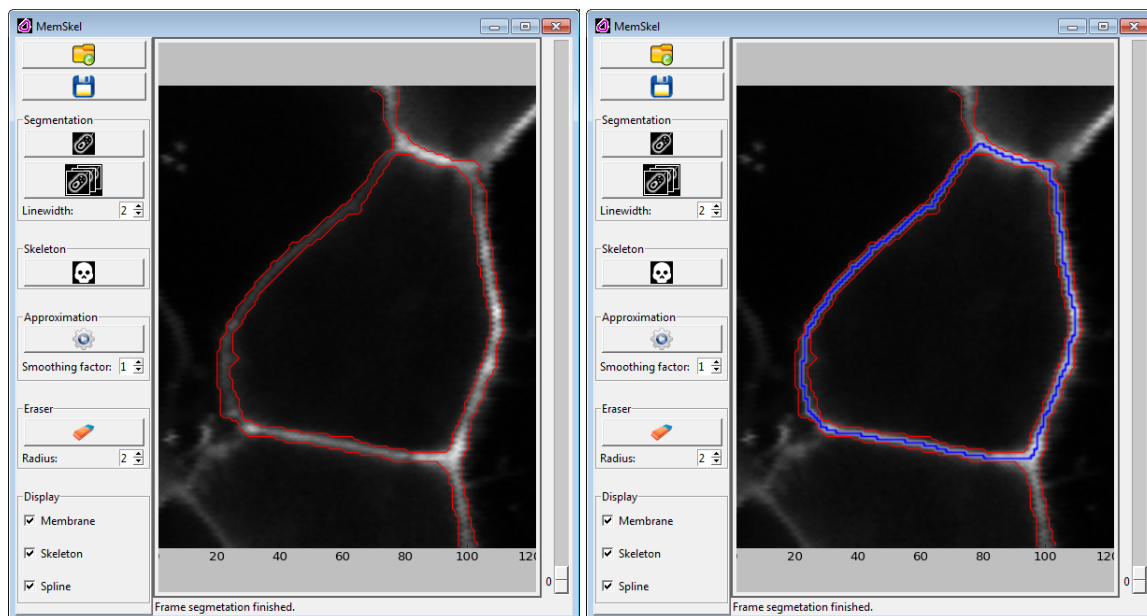
Pro instalaci balíku *pymorph* je nutné přidat python do systémové proměnné PATH. Může se stát, že nějaký instalační balík tak učiní automaticky - v takovém případě tento krok není nutný. Pro ověření je třeba spustit příkazovou řádku (command line), např. stiskem *Win+R* a vepsáním *cmd* do otevřeného okna viz obr. 6. Vepsáním příkazu *python -help* do spuštěné příkazové řádky by se měla zobrazit nápověda pro programovací jazyk viz obr. 7a. Pokud se ale zobrazí okno viz obr. 7b, znamená to, že je třeba python přidat do systémové proměnné. To lze jednoduše zařídit příkazem *setx path "%path%;c:\Python27"*, měla by se poté vypsát hláška potvrzující přidání, viz obr. 8. Po restartování příkazové řádky by se po zopakování experimentu měla zobrazit hláška viz 7a.

Jakmile je přidána cesta do systémové proměnné, je možné nainstalovat balík *pymorph* pomocí balíku *setuptools*. Je nutné v příkazové řádce přejít do adresáře *pymorph-0.96* a instalaci spustit pomocí příkazu *python setup.py install*. Pokud proběhne instalace správně, bude příkazová řádka vypadat podobně jako na obr. 9.

### 5.3 Spuštění programu

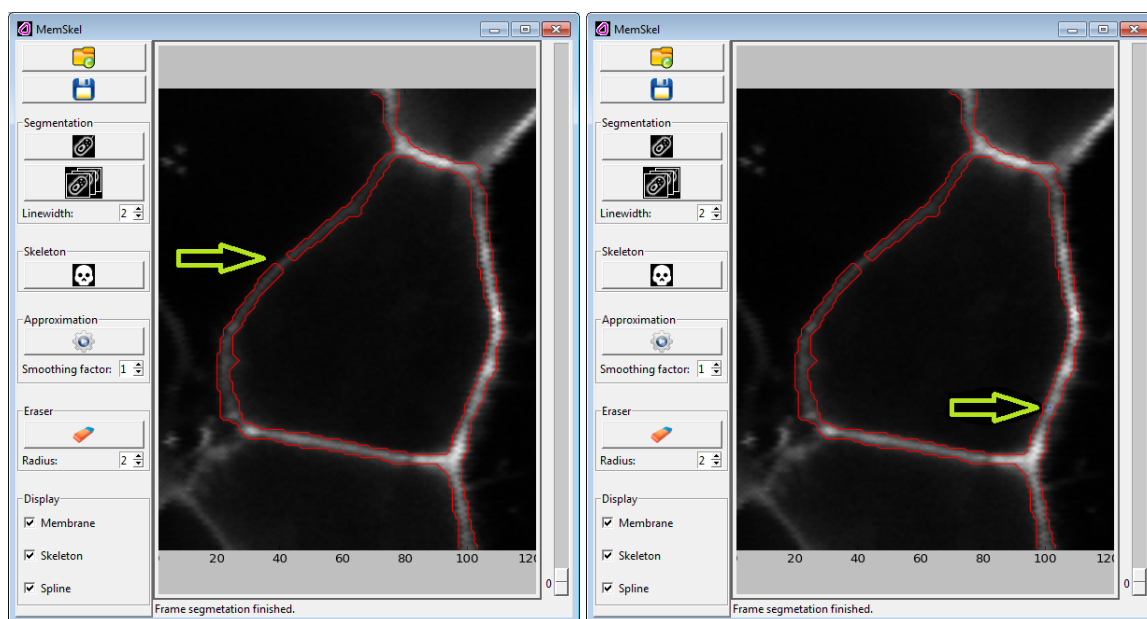
Jakmile je python přidán do systémové proměnné, je možné program *MemSkel* spustit spuštěním souboru *memSkel\_main.py*, který se nachází přímo mezi zdrojovými daty. Eventuelně je možné spustit program z příkazové řádky příkazem *python memSkel\_main.py*, je třeba nacházet se v adresáři, který daný soubor obsahuje, viz obr. 10.





(a)

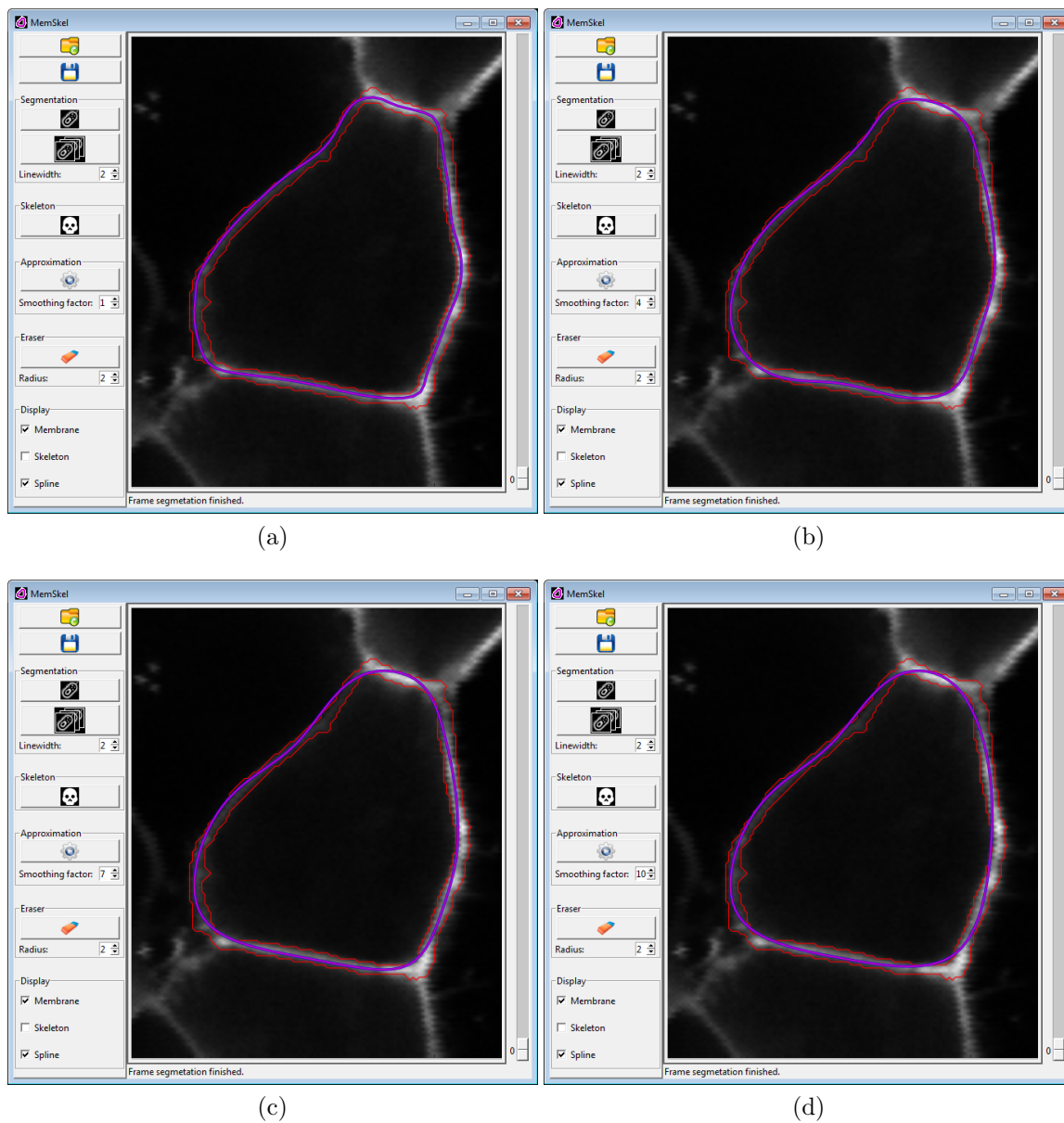
(b)



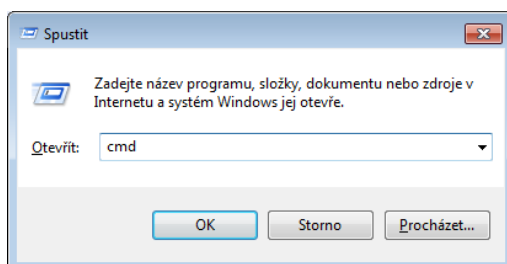
(c)

(d)

Obrázek 4: Demonstrace použití nástroje pro určení střednice. Obr. 4a je znázorněna správně segmentovaná membrána a příslušná střednice je na obr. 4b. V případě špatně segmentované membrány viz 4c, selže i výpočet střednice viz 4d.



Obrázek 5: Demonstrace aproximace splinou a různých stupňů vyhlazení: faktor = 1 ... 5a, faktor = 4 ... 5b, faktor = 7 ... 5c a faktor = 10 ... 5d.



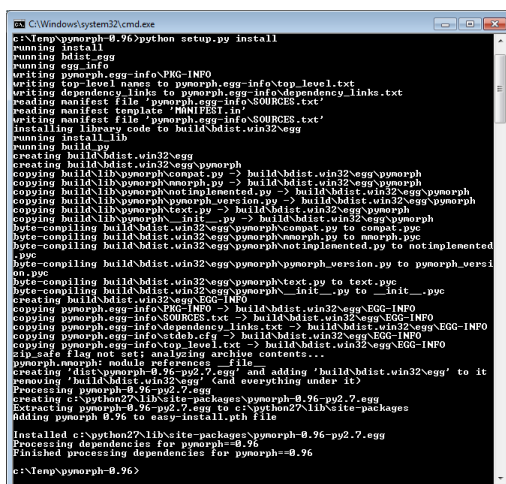
Obrázek 6: Spuštění příkazové řádky.

(a) Python zaveden v promenne PATH

(b) Python nezaveden v promenne PATH

Obrázek 7: Ověření zavedení pythonu v systémové proměnné.

Obrázek 8: Přidání pythonu do systémové proměnné.



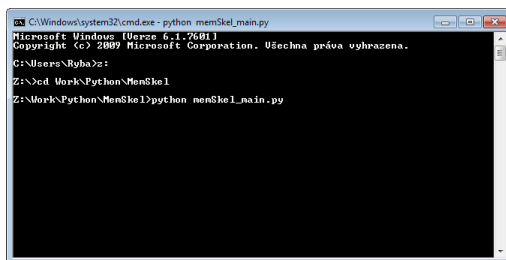
```

C:\Windows\system32\cmd.exe
c:\temp\pymorph-0.96>python setup.py install
running install
running bdist_egg
running egg_info
writing pymorph.egg-info\PKG-INFO
writing top-level names to pymorph.egg-info\top_level.txt
writing dependency links to pymorph.egg-info\dependency_links.txt
reading manifest file 'pymorph.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'pymorph.egg-info\SOURCES.txt'
installing library code to build\bdist.win32\egg
running install_lib
running build_py
creating build\bdist.win32\egg\pymorph
copying build\lib\pymorph\__init__.py -> build\bdist.win32\egg\pymorph
copying build\lib\pymorph\nmorph.py -> build\bdist.win32\egg\pymorph
copying build\lib\pymorph\notimplemented.py -> build\bdist.win32\egg\pymorph
copying build\lib\pymorph\pymorph_version.py -> build\bdist.win32\egg\pymorph
copying build\lib\pymorph\__text__.py -> build\bdist.win32\egg\pymorph
copying build\lib\pymorph\__init__.py -> build\bdist.win32\egg\pymorph
byte-compiling build\bdist.win32\egg\pymorph\__init__.py to __init__.pyc
byte-compiling build\bdist.win32\egg\pymorph\nmorph.py to nmorph.pyc
byte-compiling build\bdist.win32\egg\pymorph\notimplemented.py to notimplemented.pyc
byte-compiling build\bdist.win32\egg\pymorph\pymorph_version.py to pymorph_version.pyc
byte-compiling build\bdist.win32\egg\pymorph\__text__.py to __text__.pyc
creating build\bdist.win32\egg\EGG-INFO
copying pymorph.egg-info\PKG-INFO -> build\bdist.win32\egg\EGG-INFO
copying pymorph.egg-info\SOURCES.txt -> build\bdist.win32\egg\EGG-INFO
copying pymorph.egg-info\dependency_links.txt -> build\bdist.win32\egg\EGG-INFO
copying pymorph.egg-info\top_level.txt -> build\bdist.win32\egg\EGG-INFO
zip safe flag not set; analyzing archive contents...
pymorph.nmorph: module references __file__
creating dist\pymorph-0.96-py2.7.egg and adding 'build\bdist.win32\egg' to it
removing 'build\bdist.win32\egg' (and everything under it)
Processing pymorph-0.96-py2.7.egg
creating c:\python27\lib\site-packages\pymorph-0.96-py2.7.egg
Extracting pymorph-0.96-py2.7.egg to c:\python27\lib\site-packages
Adding pymorph 0.96 to easy-install.pth file

Installed c:\python27\lib\site-packages\pymorph-0.96-py2.7.egg
Processing dependencies for pymorph==0.96
Finished processing dependencies for pymorph==0.96

c:\temp\pymorph-0.96>

```

Obrázek 9: Instalace balíku *pymorph*.


```

C:\Windows\system32\cmd.exe - python memSkel_main.py
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2009 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\Ruba>
Z:\Work\Python\MemSkel
Z:\Work\Python\MemSkel>python memSkel_main.py

```

Obrázek 10: Spuštění programu MemSkel z příkazové řádky.