

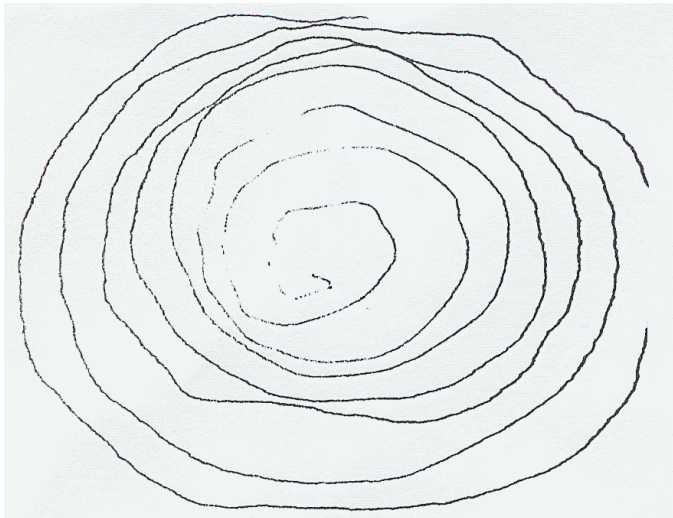
# ***Analýza spirál***

Jan Doležel,  
[dolezj2@fel.cvut.cz](mailto:dolezj2@fel.cvut.cz)

Vedoucí práce: ing. Miroslav Skrbek, Ph.D.

# 1. Úvod - čeho chceme dosáhnout

Ve spolupráci s fakultní nemocnicí v Motole připravujeme aplikaci pro detekci parkinsoniků a dalších pacientů trpícími třesem v horních končetinách. Analýza je prováděna na spirále, kterou pacient nakreslí na tablet připojený k počítači. V současnosti analýzu provádí lékař na spirále kreslené tužkou (perem) na papír.



**obr. 1 – Spirála kreslená parkinsonikem**

části se věnujeme vlastní analýze spirály a ve druhé převodu nakreslených spirál do digitální podoby, do formátu vhodného pro analýzu. Tímto postupem však ztrácíme část informace z kreslené spirály. Tablet umožňuje zaznamenat kromě souřadnic právě kresleného bodu i tlak pera na podložku a časový údaj sejmuté hodnoty. O tyto dynamické parametry jsme u spirály kreslené na papír ochuzeni. A je možné (spíše pravděpodobné), že v sobě obsahují podstatné informace o pacientovi.

Celý proces analýzy by měl být zautomatizovaný, bez zbytečné přítěže pro pacienta i lékaře. Program by měl provádět diagnózu pacientů z nakreslené spirály a stanovit závažnost postižení tremorem (třesem). Navíc by měl umožňovat správu a evidenci pacientů, jejich návštěv u lékaře, evidenci léku, které jim byly předepsány, zjistit pokroky či naopak nedostatky při léčbě.

Zatím se podařilo dosáhnout několika dílčích úspěchů. Z důvodu dočasné absence tabletu, byl vývoj rozdělen do dvou směrů. V první

## 2. Vlastní analýza

Do algoritmu pro testování vstupuje vektor dat ve formátu <x>, <y>, <čas> a <tlak> z/do textového souboru tak, jak je umožňuje snímat tablet. Každá položka vektoru reprezentuje jeden bod na spirále, jejich posloupnost pak celou spirálu směrem od středu k okraji.

Aplikace v současnosti umožňuje nahrávat tato data ze souboru, zobrazit je uživateli, vygenerovat spirálu dle zadaných parametrů, nakreslit spirálu myší a opět uložit do souboru. V aplikaci je naimplementován algoritmus testování spirály dle Seth L. Pullmana, MD, FRCPC popsany v článku *Spiral Analysis: A New Technique for Measuring Tremor With a Digitizing Tablet* (*Movement Disorders*, Vol. 13, Supplement 3, 1998).

Aplikace také umí tento spočítaný výsledek exportovat do souboru, kam se ukládá ve formátu <x>, <y>, <čas>, <tlak>, <r>, < $\theta$ >, < $\Delta\theta$ >, < $\Delta r/\Delta\theta$ >, < $\Delta^2 r/\Delta\theta^2$ >, <extrém>, <inflex>, včetně parametrů použitých při generování a výsledných hodnot do textového souboru. Poslední dvě čísla na řádku nabývají pouze 1 (bod je inflexní/extrém) nebo 0 (jinak).

### Generování spirál

Archimédovu spirálu lze popsat pomocí vzorců v polárních a kartézských souřadnicích:

- v polárních souřadnicích:

$$r = a\theta$$

$$r = \sqrt{x^2 + y^2}$$

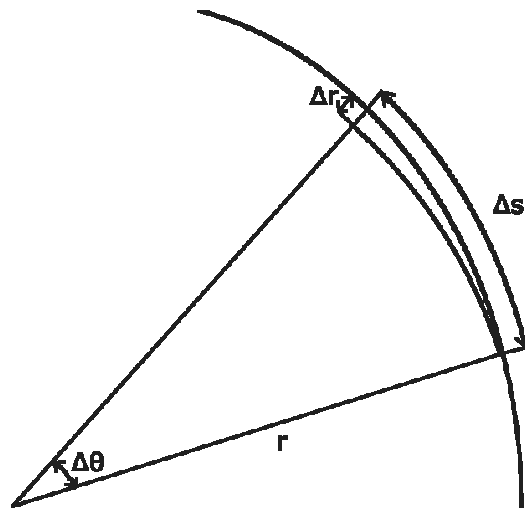
- v kartézských:

$$x = a\theta \cos(\theta + c)$$

$$y = a\theta \sin(\theta + c)$$

kde  $r$  je poloměr,  $a$  je parametr „rozevření“ spirály,  $\theta$  je úhel,  $x$ ,  $y$  jsou kartézské souřadnice a  $c$  vyjadřuje počáteční úhel.

Spirálu budeme generovat iteračně, s konstantní tangenciální rychlostí. Vydeme z těchto vzorců:



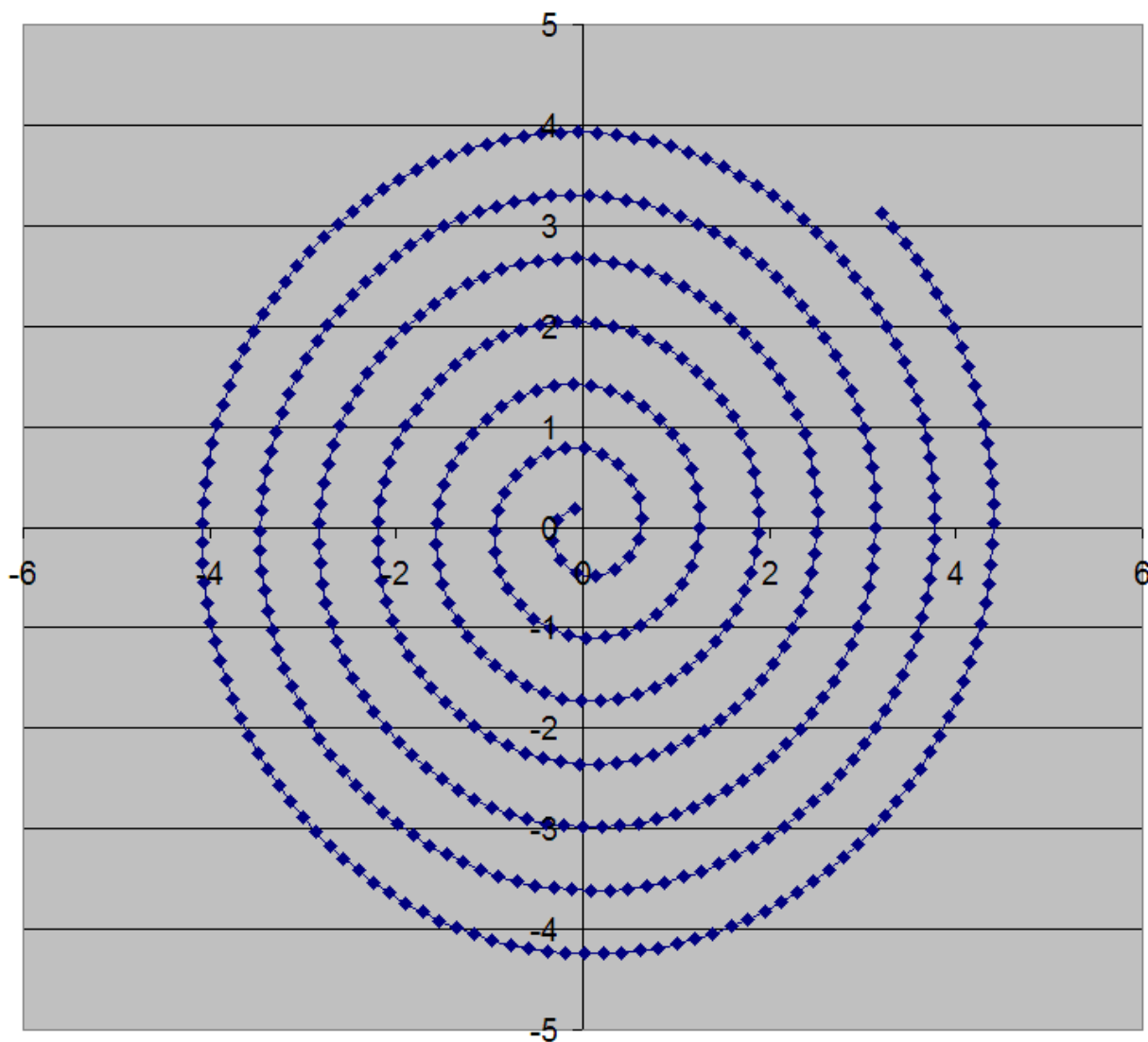
obr. 2 – Výsek na spirále

$$v = a \cdot \omega \cdot \sqrt{1 + \theta^2}$$

$$\Delta s = a \cdot \Delta \theta \cdot \sqrt{1 + \theta^2}$$

a jednoduchou úpravou dostaneme:  $\Delta \theta = \frac{\Delta s}{a\sqrt{1 + \theta^2}}$

kde  $\Delta \theta$  je přírůstek úhlu mezi sousedními body vyjádřený v závislosti na  $\Delta s$ , což je vzdálenost sousedních bodů na spirále a  $\theta$  je velikost úhlu z minulé iterace.



obr. 3 - Dokonalá spirála

Na tuto spirálu pak můžeme namodulovat šum. Spirálu můžeme deformovat pomocí jedné úpravy a dvou parazitních signálů.

**1)** rozdílnými koeficienty u  $x$  a  $y$  se nám spirála deformuje do tvaru elipsy, tedy:

$$x = a\theta \cos(\theta + c)$$

$$y = b\theta \sin(\theta + c)$$

**2)** na spirálu namodulujeme sinusoidy:

$$x = a\theta_1 \cos(\theta_1 + c) + A \sin(\theta_2) \cos(\theta_1)$$

$$y = b\theta_1 \sin(\theta_1 + c) + A \sin(\theta_2) \sin(\theta_1)$$

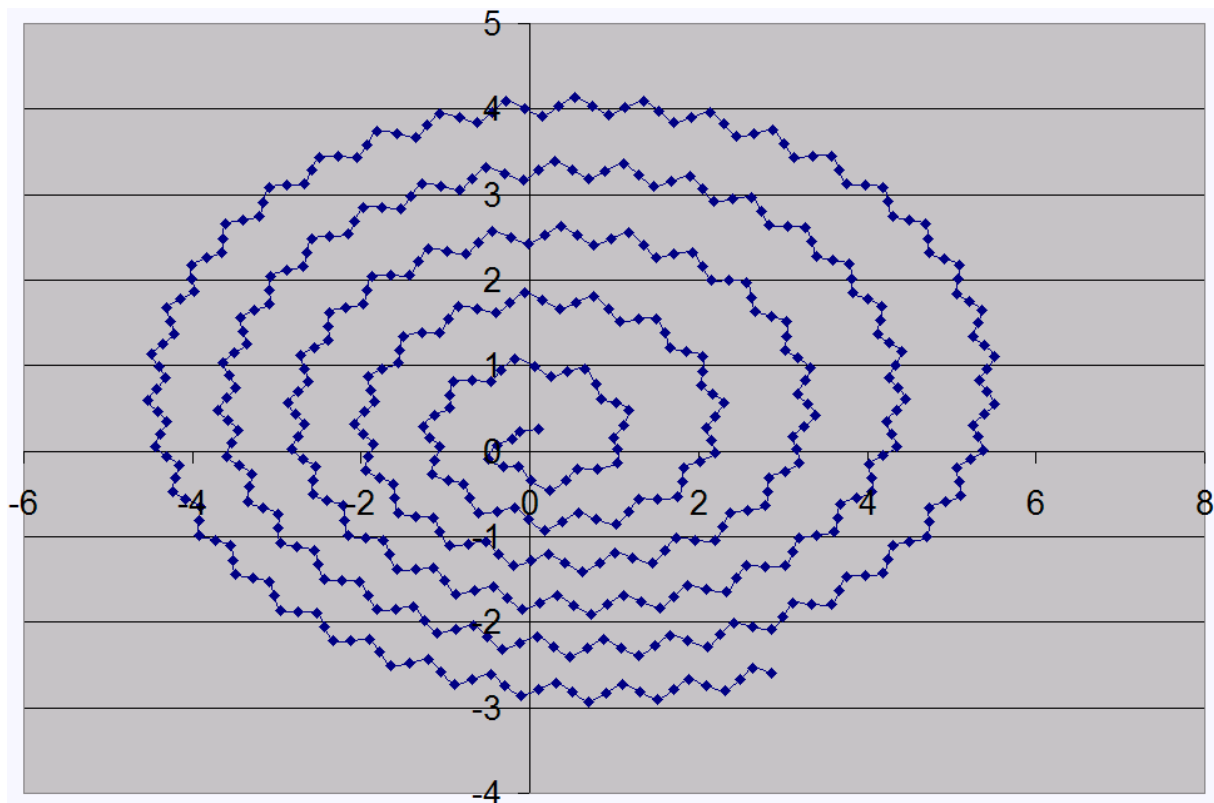
$\theta_1$  je úhel používaný výše,  $\theta_2$  je úhel sinusoidy,  $A$  její amplituda

3) spirálu je možno také „pokřivit“ namodulováním sinusoidy jiným způsobem:

$$x = a\theta_1 \cos(\theta_1 + c) + A \sin(\theta_2) \cos(\theta_1) + B \sin(\theta_1 / p)$$

$$y = b\theta_1 \sin(\theta_1 + c) + A \sin(\theta_2) \sin(\theta_1) + B \sin(\theta_1 / p)$$

zde přibyly další parametry –  $B$  je amplituda druhého sinusového signálu, a  $p$  je jeho poměr vůči úhlu generování spirály



obr. 4 - Deformovaná spirála

Celý tento postup je možno shrnout do algoritmu generování spirály:

```
done = false,  $\theta_1 = \theta_2 = 0$ , time = 0, pres = 1;
```

```
while(!done) {
     $x_1 = a * \theta_1 * \cos(\theta_1)$ ;
     $y_1 = b * \theta_1 * \sin(\theta_1)$ ;
     $x_2 = A * \sin(\theta_2) * \cos(\theta_1)$ ;
     $y_2 = A * \sin(\theta_2) * \sin(\theta_1)$ ;
     $z = B * \sin(\theta_1 / p)$ ;
```

```
     $x = x_1 + x_2 + z$ ;
```

```

y = y1 + y2 + z;
createNewRow(x, y, time, pres);

```

```

Δθ1 = Δs / 2 / (a + b) / √(1+θ12);
θ1 += Δθ1;
θ2 += Δθ2;
time += 5;
if(θ1 > θ) {
    done = true;
}
}

```

tučně jsou vyznačeny parametry, které lze ovlivnit. Popsány byly výše. Spirála je generována s konstantní tangenciální rychlostí a s konstantními časovými rozestupy mezi body 5ms.

Pro generování spirály lze doporučit parametry, které byly otestovány v praxi:

Při čtyřech otáčkách:

a = b = 200,      Δs = 50,      dpi = 1000 - odpovídá datům z tabletu  
a = b = 10,      Δs = 5,      dpi = 72 - odpovídá datům z myši

Při použití nastavení pro tablet:

<b>B</b>	<b>p</b>	<b>Popis</b>
100	0,01	Podobné namodulované sinusoidě
100	0,3333	Deformace do čtverce
1000	40	Spirála je natěsnána ke straně
1000	0,1 – 1	Podivné deformace
1000	0,9; 1; 1,1	Spirála je uvnitř zploštělá
1000	1 – 10	Uvnitř zploštělá a natočená
10000	60	Šroubovice

A = 100,      θ<sub>2</sub> v poměru k π - hezká namodulovaná sinusoida

## Testování spirál

Jak bylo zmíněno výše, je k testování použit algoritmus podle Pullmana. Tento algoritmus využívá statistické metody. Vypočítává 5 parametrů – I<sub>1</sub> až I<sub>5</sub> a tři z nich využívá k výpočtu DOS (degree of severity), která by měla odpovídat „modified United Parkinson’s Disease Rating Scale“ v rozsahu 0 až 4. Stupnice je rozdělena na části: 0-1 normální, 1-2, lehká, 2-3 střední, 3-4 závažná.

Jednotlivé parametry se vypočtou podle následujících vzorců:

$$i_1 = \frac{1}{\theta} \sum_{j=1}^J \left( \frac{\Delta r}{\Delta \theta} - \bar{r}_\theta \right)^2$$

$$I_1 = \ln i_1$$

což je vlastně standardní odchylka. Proměnné mají tento význam:  $\theta$  je celkový úhel,  $J$  počet bodů spirály,  $\Delta r$  rozdíl vzdálenosti od středu mezi dvěma následujícími body,  $\Delta\theta$  je pak jejich rozdíl úhlů. Poslední proměnná je pak průměrná derivace  $r$  podle  $\theta$  přes všechny body (odpovídá koeficientu  $a$  v polárních souřadnicích – viz generování spirály). Neboli proložíme „ideální“ spirálu, jenž nejlépe odpovídá nakreslené a spočítáme odchylku...

$$i_2 = \frac{1}{\theta} \sum_{j=1}^J \left( \frac{\Delta \frac{\Delta r}{\Delta \theta}}{\Delta \theta} - \bar{dr}_\theta \right)^2$$

$$I_2 = \ln i_2$$

$I_2$  vyjadřuje standardní odchylku druhé derivace. Nová proměnná vyjadřuje průměrnou druhou derivaci  $r$  podle  $\theta$  ( $i_2$  by ideálně měla být rovna nule).

$$I_3 = (\theta / R - 14\pi) / 2\pi$$

kde  $R$  je maximální poloměr. Tento parametr vyjadřuje počet otáček spirály vztažený k maximálnímu poloměru a 7 otáčkám... Význam tohoto parametru úplně nechápu.

$$I_4 = \frac{1}{2(J-1)} \sum_{j=2}^J \left( \text{sign} \left( \frac{\Delta r_{j-1}}{\Delta \theta_{j-1}} - \bar{r}_\theta \right) - \text{sign} \left( \frac{\Delta r_j}{\Delta \theta_j} - \bar{r}_\theta \right) \right) \cdot 100\%$$

Vyjadřuje procentuální počet extrémů (maxim a minim) na spirále vzhledem k počtu všech bodů.

$$I_5 = \frac{1}{2(J-1)} \sum_{j=2}^J \left( \text{sign} \left( \frac{\Delta \frac{\Delta r_{j-1}}{\Delta \theta_{j-1}}}{\Delta \theta_{j-1}} - \bar{dr}_\theta \right) - \text{sign} \left( \frac{\Delta \frac{\Delta r_j}{\Delta \theta_j}}{\Delta \theta_j} - \bar{dr}_\theta \right) \right) \cdot 100\%$$

Poslední parametr je procentuální počet inflexních bodů vzhledem k počtu všech bodů.

Formule DOS, se pak vypočte jako  $\text{DOS} = 0.4615 * I_1 + 0.0544 * I_5 - 0.2331 * (I_1)^2 - 0.0726 * (I_2)^2 - 0.001 * (I_5)^2 + 0.2539 * I_1 * I_2 + 1.3668$

Tento postup lze opět vyjádřit formou algoritmu:

```
createNewVector(res);
time0 = time = row[0].time;
x0 = row[0].x; y0 = row[0].y;
theta = r_bar_theta = dr_bar_theta = d2rfi = drfi1 = drfi = 0;

x1 = (row[0].x - x0) / dpi;
y1 = (row[0].y - y0) / dpi;
r_max = r1 = sqrt(x1^2 + y1^2);
Delta_theta = theta1 = atan2(y1, x1);
res.add(createNewRow(x1, y1, 0, row[0].pres, r1, theta, Delta_theta, drfi, d2rfi));
```

```

for(int j = 1; i < row.size; i++) {
    if(row[j].time < time) {
        continue;
    }
    x2 = (row[j].x - x0) / dpi;
    y2 = (row[j].y - y0) / dpi;
    r2 =  $\sqrt{x_2^2 + y_2^2}$ ;
    rmax = max(rmax, r2);
    Δr = r2 - r1;
    θpom = θ2 = atan2(y2, x2);
    if(abs(θ2 - θ1) > π) {
        if(θ1 < 0) {
            θ1 += 2 * π;
        }
        else {
            θpom += 2 * π;
        }
    }
    Δθ = θpom - θ1;
    if(abs(Δθ) > 0.000001) {
        drfi = Δr / Δθ;
        d2rfi = (drfi - drfi1) / Δθ;
        θ += Δθ;
        res.add(createNewRow(x2, y2, row[j].time - time0,
            row[j].pres, r2, θ, Δθ, drfi, d2rfi));
        if((abs(drfi) > 1000) || (abs(d2rfi) > 1000)) {
            continue;
        }
        r̄θ += drfi;
        d2r̄θ += d2rfi;
    }

    r1 = r2;
    x1 = x2;
    y1 = y2;
    θ1 = θ2;
    drfi1 = drfi;
    time += step;
}

r̄θ /= res.size;
d2r̄θ /= res.size;
θ = abs(θ);
i1 = i2 = i4 = i5 = 0;

for(int j = 1; i < res.size; i++) {
    i1 += (res[j].drfi - r̄θ)2;
}

```



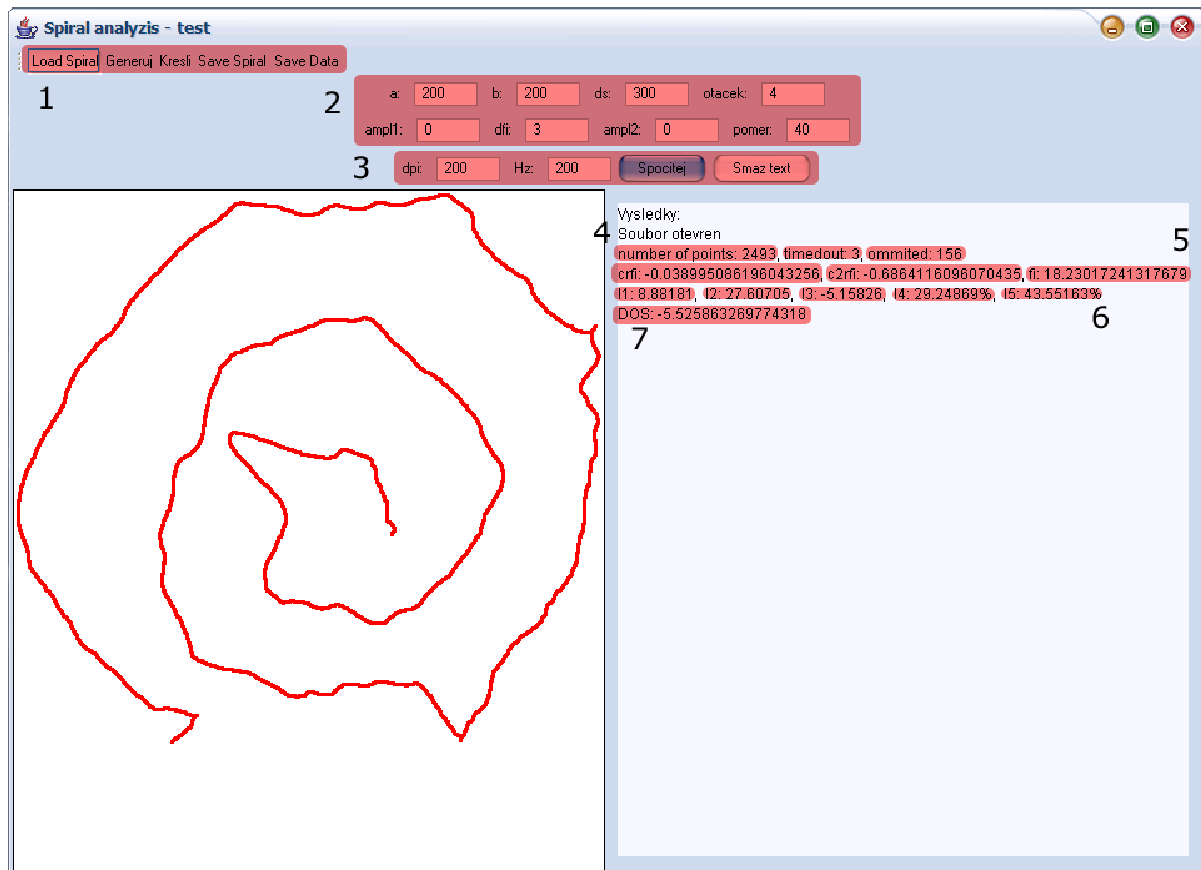
```

    i2 += (res[j].d2rfi - d $\bar{r}_\theta$ )2;
    i4 += abs(sign(res[j].drfi -  $\bar{r}_\theta$ ) - sign(res[j-1].drfi -  $\bar{r}_\theta$ ));
    i5 += abs(sign(res[j].d2rfi - d $\bar{r}_\theta$ ) - sign(res[j-1].d2rfi - d $\bar{r}_\theta$ ));
}

I1 = ln(i1 /  $\theta$ );
I2 = ln(i2 /  $\theta$ );
I3 = (( $\theta$  / rmax) - (14 *  $\pi$ )) / (2 *  $\pi$ );
I4 = i4 / (res.size - 1) * 50;
I5 = i5 / (res.size - 1) * 50;
DOS = (0.4615*I1 + 0.0544*I5 - 0.2331*(I1)2 - 0.0726*(I2)2 -
        0.001*(I5)2 + 0.2539*I1*I2 + 1.3668);

```

## Návod k použití



**obr. 5 - Okno aplikace pro testování spirál**

- 1) Load Spiral – nahraje spirálu ze souboru ve formátu <x> <y> <čas> <tlak>
- Generuj – vygeneruje spirálu dle parametrů 2. okýnka
- Kresli – umožní kreslit spirálu myši (po skončení klepnout na konec kreslení)
- Save Spiral - uloží spirálu do souboru ve formátu <x> <y> <čas> <tlak>
- Save Data – uloží vypočítaná data do souboru (nejdříve nutné provést výpočet)

- 2) Hodnoty použité při generování spirály odpovídají parametrům popsaným výše pro generování spirály: a (a), b (b), ds ( $\Delta s$ ), otáček ( $\theta/2\pi$ ), ampl1 (A), d $\theta$  ( $\Delta\theta$ ), ampl2 (B), pomer (p)
- 3) Hodnoty použité při výpočtu. dpi je rozlišení v kterém byla spirála nakreslena, Hz je frekvence s jakou mají být vybírány. Při generování je použita frekvence 200 Hz (vzdálenost dvou bodů je 5ms). Tlačítko „Spočítej“ spustí výpočet a tlačítko „Smaž text“ vymaže konzoli s výstupem.
- 4) Number of points je celkový počet bodů na spirále, timeout je počet bodů, které byly vynechány kvůli použité frekvenci, ommited je počet bodů, které příliš vybočovali z řady a byly při výpočtu  $\bar{r}_\theta$  a  $d\bar{r}_\theta$  vynechány (při výpočtu  $I_x$  vynechány nebyly)
- 5) crfi je  $\bar{r}_\theta$ , c2rfi je  $d\bar{r}_\theta$ , fi je vypočítaný úhel  $\theta$
- 6+7)  $I_x$  jsou vypočítané hodnoty koeficientů a DOS výsledná hodnota

Uvedené textové značení je použito i ve výstupních souborech a má stejný význam. V panelu nalevo od výsledků se zobrazuje aktuální spirála. Ať už nahraná vygenerovaná nebo (na)kreslená.

## Problémy, k zamyšlení...

Největším problémem výše uvedeného řešení je, že nedává správné výsledky (jak potvrdily experimenty). Vypočítaná DOS se pohybuje přibližně v rozsahu -20 až do +13. Evidentně se do rozsahu 0-4 nelze vejít.

Na testovaných datech (poskytnuté Motolem) se rozsahy DOS pohybovaly v rozmezí od -20 do +6, podle toho, jaké rozlišení (dpi) a jak hrubě (step – frekvence bodů) bylo při výpočtu použito. (O převodu spirál do digitální podoby pojednává další část.) Spirály kreslené myší se do stupnice taktéž nevešly. Nevypozoroval jsem ani žádnou závislost vypočtené DOS na předpokládané závažnosti postižení pacienta. DOS má spíše náhodný charakter.

Tento problém by mohl být důsledkem dílčích problémů:

Problémem je vůbec definovat a najít střed spirály. V současném řešení se za střed považuje první zaznamenaný bod. Ale střed by mohl být také střed elipsově obálky spirály, popřípadě i jinak definované body. Metoda výpočtu DOS uvedená výše je na určení středu velice citlivá.

Otázkou také je, jaký druh signálu pacienti postižení parkinsonovou chorobou produkují a nakolik je uvedená metoda na tento parazitní signál citlivá. Možná bude také nutné přidat do aplikace další generátor šumu, pokud tento nepůjde popsat již existujícími prostředky. U tohoto šumu můžeme zkoumat například jeho frekvenci a frekvenční spektrum, zda se projevuje po celé spirále, či jen v určitých úsecích a další. Velkou pomocí při dalších analýzách bude i tablet, který poskytuje navíc informace o rychlosti kreslení a tlaku pera na podklad.

Další problémy a pozorování:

- Namodulovaný parazitní signál nedrží tangenciální rychlost generování.
- Vyšší frekvence namodulovaných signálů zvětšuje vypočtenou chybu. Možná, že je to správné chování...
- Vstupní parametry (x,y) algoritmu jsou velice ovlivněny vzorkováním. Pro výpočet se používají souřadnice přepočtené na centimetry.

Je také možné, že tato metoda nebude dávat nikdy uspokojivé výsledky. Dalšími možnými postupy testování se věnuji v dokumentu Semestrální práce z 36NAN – Analýza spirál. Pro konkrétní závěry však není dostatek testovacích dat.

Aplikace v současném stavu umožňuje pouze testování spirál uvedenou metodou, jejich nahrání a uložení. Neobsahuje žádné prostředky evidence pacientů, ani zatím neumožňuje komunikaci s tabletem.

# 3. Digitalizace spirál

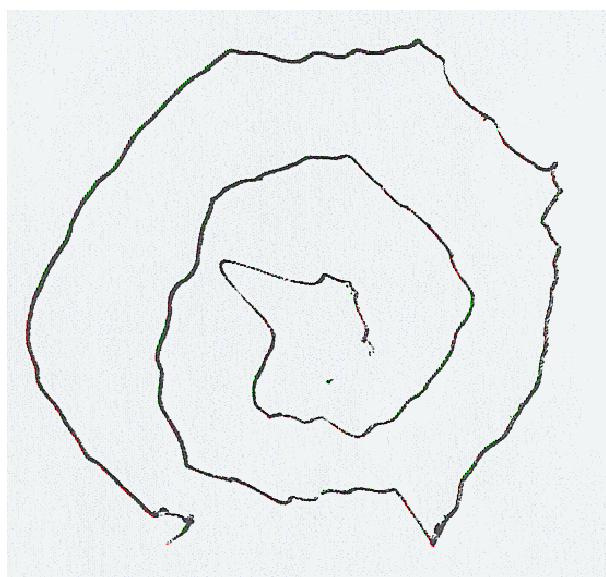
Tento program by měl umožňovat automatický převod spirál nakreslených na papír do digitální podoby vhodné pro analýzu – posloupnost souřadnic x a y. Jeden takový bod nalezené spirály odpovídá jednomu pixelu naskenovaného obrázku. Tento proces se skládá ze sedmi kroků:

1. naskenování spirály a uložení jako obrázku v počítači
2. nahrání obrázku do aplikace
3. aplikování operace threshold (práh)
4. 2x aplikování operace dilatace (ztluštění čáry)
5. 10x aplikování operace skeleton (ztenčení čáry)
6. nalezení spirály
7. uložení spirály do souboru

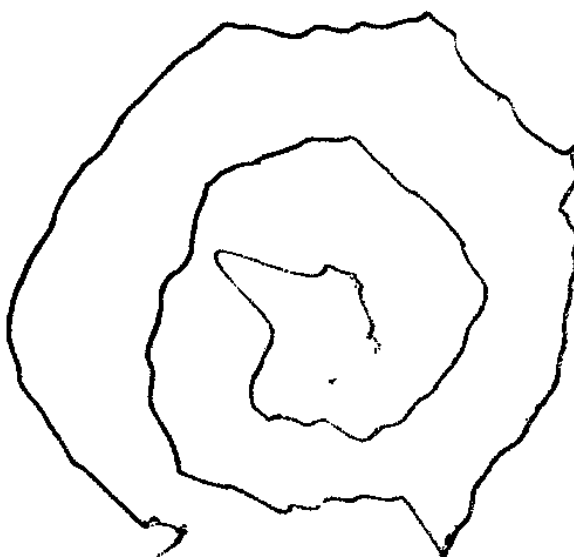
## Filtry

Aplikace pro manipulaci s obrázky využívá několika filtrů (threshold, dilate, skeleton) z balíku JH Labs (<http://www.jhlibs.com/ip/filters.html>). Filtr threshold převede barevný obrázek do obrázku, který obsahuje pouze dvě barvy. Jako parametr se filtru předává číslo (0-255) – práh. Je-li některá ze složek RGB pixelu vyšší než tento práh, je výsledný pixel černý. V opačném případě bude bílý.

Druhé dva filtry pracují pouze s obrázky v dvoubarevné formě.



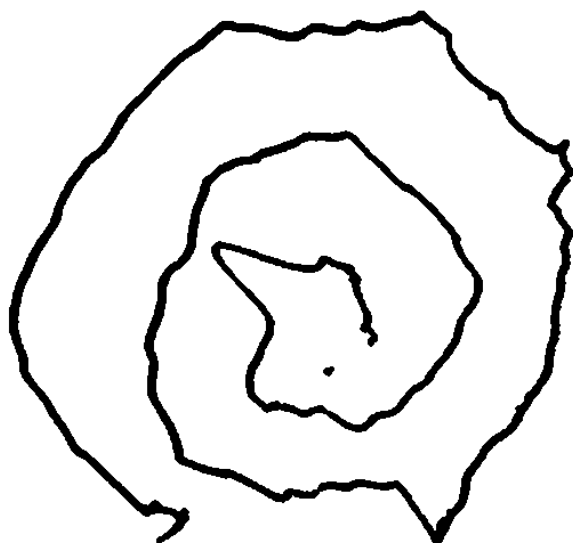
obr. 6 - Původní spirála



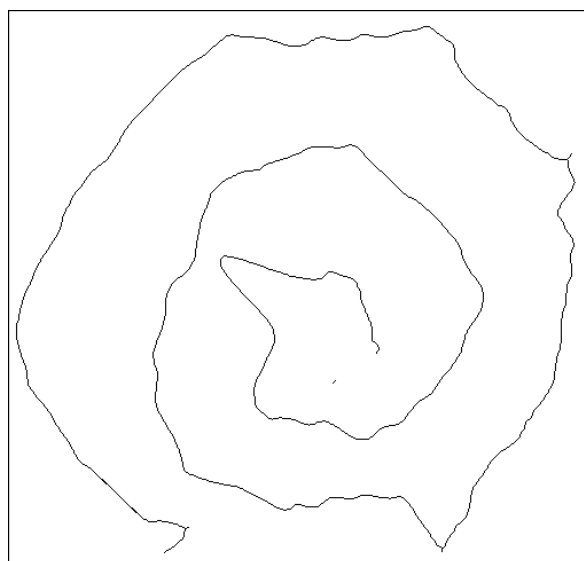
obr. 7 - Spirála po operaci threshold

Operace dilatace začerní pixely v okolí (standardně jen sousední pixely) každého černého pixelu. Po aplikování tohoto filtru bude každá černá plocha (i jednotlivý pixel) o jeden pixel širší. Po prahování zbudou na spirále artefakty v podobě prázdných (bílých) míst, spirála se může přerušit a objeví se výběžky (větvičky). Operace dilatace by měla tato místa vyhladit a eliminovat. Právě dvě tyto operace provedené po sobě se ukázaly jako optimální (jedna bývá nedostatečná a při třech se už ztrácejí jemnější details).

Skeleton je velice podobný filtru eroze, ale na rozdíl od něj se zastaví na čáře široké přesně jeden pixel. Eroze ubírá z okrajů černých ploch pixely a jelikož se na žádné hranici nezastaví, dojde k umazání části spirály. Což je nepřijatelné. Každé aplikování skeletonu ubere z okraje plochy jeden pixel (pokud by se ovšem plocha nerozpadla na více částí). Praxe ukázala, že deset těchto operací po sobě je dostatečných.



obr. 8 – Spirála po dilataci



obr. 9 – Několika násobné opakování skeletonu

Po aplikování těchto filtrů v tomto pořadí, by nám měla zůstat spirála přesně jeden pixel široká.

## Hledání spirály

Vstupem do algoritmu hledání spirály je dvoubarevný obrázek s nepřerušenu spirálou tlustou právě jeden pixel. Při nedodržení této podmínky se algoritmus zacyklí nebo spirálu nenajde. Hledání spirály probíhá v několika částech:

1. nalezení (alespoň jednoho bodu) spirály
2. nalezení konce spirály
3. nalezení celé spirály
4. určení počátečního bodu

Při hledání bodu spirály se postupuje z levého horního rohu po diagonále. Narazíme-li na černý bod, pokusíme se vypravit po spirále. Není-li to artefakt (posloupnost je dostatečně dlouhá – více jak 5 bodů), narazili jsme na spirálu a došli na jeden z jejích konců (záleží na tom, zda je spirála levotočivá, nebo pravotočivá).

Vydáme se tedy po spirále zpátky a dojdeme na druhý konec. Který z těchto krajních bodů je blíže středu obrázku, ten prohlásíme za střed spirály.

Vstupem pro procházení spirály je aktuální bod, poslední navštívený bod a hloubka zanoření (počet rozvětvení spirály). Dále se pokračuje takto:

- 1) Najdi body vhodné pro pokračování – v okolí 1 pixelu kolem aktuálního bodu jsou nalezeny černé body, které nepatří do již nalezené části.
- 2) Vyber z nich nejbližší bod, kterým se má pokračovat, zařaď ho do posloupnosti a přejdi do něj
- 3) Pokud se spirála rozděljuje, vyber dva vhodné sousední body (spirála se nemůže dělit do více směrů než dvou). Rekurentně vyvolej procházení na tyto dva body a zvyš hloubku zanoření. Po návratu z rekurze vezmi kratší posloupnost, přidej jí na konec dosud nalezené, invertuj ji a znovu přidej (cesta tam a zase zpátky), přidej na konec delší posloupnost ukončí toto procházení a vrať vytvořenou posloupnost.
- 4) Pokud nebyl nalezen žádný další bod, ukončí toto procházení a vrať posloupnost.

Implementace algoritmu hledání spirály je rozsáhlá a pro větší detaily odkazují na zdrojové kódy ve třídě `spiral.digitalize.Spiral`.

## Problémy, k zamyšlení...

Aplikace trpí několika neduhy. První, na co narazíme je, že operace skeleton z balíku filtrů JH Labs není naimplementována bez chyb. Může se stát, že se zacyklí nebo vyhodí výjimku (Out of memory, Null pointer, Stack overflow). Vypadá to, že se to stává kvůli jednomu nebo pár bodům (nejspíše nějaké větvičky), kvůli kterým je asi špatně detekován konec některé z vnitřních smyček algoritmu (kterých je v něm, dle zběžné prohlídky, pozhnaně). K nápravě stačí před použitím filtru skeleton (toho který se zacyklí) použít jednu filtr dilatace. Avšak to již vylučuje plně automatický převod spirály.

Operace skeleton také vytváří na okraji obrázku rámeček. Spirála se proto nesmí dotýkat okraje, nebo být blízko něj, protože po použití operace dilatace a skeleton by se algoritmus pro hledání spirály na rámečku zacyklil.



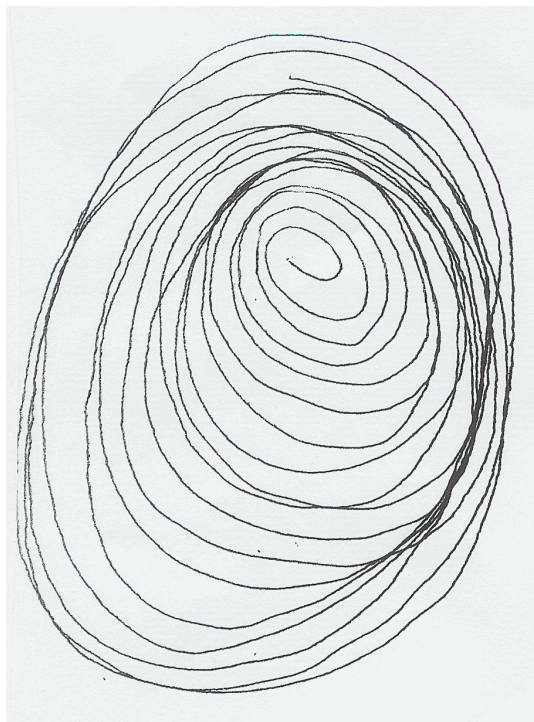
Algoritmus hledání nepočítá s možností, že se v nějakém bodě spirála rozvětjuje do tří cest. Tato možnost je velice nepravděpodobná, není však vyloučena. Může nastat zejména v případech krátkých (1-5 bodů) větviček. Na to opět pomáhá jedna operace dilatace následovaná dostatečným počtem skeletonů. Ale stejně jako výše je nutný zásah uživatele.

**obr. 10 –  
Rozvětvení do tří  
cest**

Algoritmus také nenajde celou spirálu, pokud je i po dvojitém aplikování dilatace stále přerušená. Najde jen její část. Tyto spirály vznikají zejména při „odfláknutém“ kreslení a kreslení tužkou. Při používání pera, tenké fixy nebo jiného dostatečně výrazného nástroje, by neměly vznikat.

Existují i spirály, které je již z principu nemožné převést do podoby sledu souřadnic  $(x,y)$ , protože se tah protíná a nelze potom vysledovat historii kreslení. Tyto spirály bude možné snímat jen pomocí tabletu.

Posledním problémem, kterým je program zatím vybaven spočívá v tom, že celou analýzu je nutné provést dvoukolově. Nejdříve provést sadu filtrů a teprve po té, co skončí, spustit algoritmus hledání spirály.



**obr. 11 – Spirála, kterou nelze převést**

## 4. Závěr

Podářilo se napsat dvě spolupracující aplikace. Naimplementovaný diagnostický algoritmus však nesplnil očekávání. Možná se ho podaří opravit, upravit nebo úplně nahradit funkční metodou diagnózy parkinsoniků. Až bude tato metoda správně zařazovat pacienty, bude nutné aplikaci rozšířit o evidenci a databázi pacientů, jak byla popsána v úvodu.

Druhá aplikace již celkem obstojně, za menší pomoci uživatele, dokáže převádět spirály kreslené na papír do digitální podoby (posloupnost souřadnic). Ještě bude třeba celý proces upravit, aby byl plně automatický, ale již nyní je program použitelný.

Dále by bylo vhodné se zaměřit na sehnání dalších dat – zřejmě nakreslených spirál, převést je do sledu souřadnic a najít metodu, která je bude správně analyzovat. Současným problémem je, že dvacet spirál, které jsou momentálně k dispozici, je pro tento úkol velice málo.

Také by bylo vhodné získat tablet, připojit ho k aplikaci a získávat data z něj. Tyto informace by měly mnohem větší vypovídací hodnotu než data statická, kreslená na papír.



# 5. Literatura a zdroje

- Seth L. Pullman, MD, FRCPC. Spiral Analysis: A New Technique for Measuring Tremor With a Digitizing Tablet (*Movement Disorders*, Vol. 13, Supplement 3, 1998)
- Jerry's Java Image Processing Pages: <http://www.jhlab.com/ip/>