**Slide 1**

Dobrý den, já jsem David Rendl a spolu s docenty Šroubkem a Šloufem pracujeme na metodě pro měření a analýzu práškových difraktogramů.

**Slide 2**

Tato metoda, nazvaná 4D STEM powder nanobeam difraction funguje tak že za pomocí skenovacího elektronového mikroskopu vybaveného 2D pixelovaným detektorem pořídíme difrakční záznamy z vybrané oblasti vzorku a ty dále analyzujeme. Odtud také pochází ten název, 4D STEM, protože měřením získáme 2D pole 2D difrakčních záznamů. Tyto záznamy by bylo za normálních okolností velmi obtížné analyzovat, krystaly ve vzorku jsou náhodně orientované, takže i difrakční obrazce jsou náhodně orientované a analýza je sice možná, ovšem ne pro běžného uživatele nebo studenta. My vytváříme z těchto difrakčních záznamů práškové difraktogramy, tím že jednotlivé difrakční obrazce posčítáme. Zpracování práškového difraktogramu už je poměrně snadné. Prvním krokem ve zpracování je odfiltrování nevhodných souborů, tady na obrázku a lze vidět, jak to dopadne, když se snímky nasčítají rovnou bez filtrování. My potřebujeme vzít pouze takové snímky které mají hodně difrakčních peaků. Dalším krokem je pak dekonvoluce, o které se budeme dnes bavit víc.

**Slide 3**

Než přejdeme k dekonvoluci, tak zde je ukázka filtrování souborů. V původních verzích této metody se filtrování provádělo pouze na základě entropie, chceme snímky s co nejvyšší entropií. Aktuálně se snímky pro sumaci vybírají spolu s entropií i na základě dalších kritérií, jako je například počet peaků a jejich intenzita.

**Slide 4**

Posledním krokem před sečtením difrakčních záznamů je dekonvoluce. Momentálně se v knihovně stemdiff používá Richardson-Lucyho algoritmus a já bych rád krátce zmínil, jak funguje a jak se k němu dojde. Máme nějaký naměřený snímek I, rozptylovou funkci P a chtěli bychom zrekonstruovat ostrý snímek O. Předpokládáme, že snímky vznikají Poissonovským procesem, což je v mikroskopii rozumný předpoklad, takže máme věrohodnostní funkci p(I|O) což je jen sdružená distribuční funkce pro Poissonovo rozdělení, tu zlogaritmujeme a výsledný funkcionál L(O) minimalizujeme, čímž dostaneme tuto podmínku. Zde ještě využijeme předpokladu, že snímky se mezi iteracemi už prakticky nebudou měnit po dosažení určitého počtu iterací, tedy že k+1 iterace ku k-té iteraci je rovna jedné, a za tuto jedničku dosadíme a získáme tak iterativní postup pro dekonvoluci snímků, který se aktuálně používá v té naší metodě.

**Slide 5**

Momentálně pracujeme na vylepšení, a sice na implementaci regularizovaných verzí pro Richardson-Lucyho algoritmus, které jsou robustnější vůči šumu. Ty získáme podobným způsobem jako v předešlém případě, jen s tím rozdílem že přidáme regularizační člen, buď L2 normu, pak se bavíme o Tikhonov-Millerově regularizaci, nebo L1 normu, to pak nazýváme totální variací. Stejně jako v předchozím případě získáme iterativní postup pro dekonvoluci.

**Slide 6,7**

Rozdíl v regularizovaných a neregularizovaných verzích si ukážeme na nějakém obyčejném snímku, zde máme snímek Leny rozmazaný Gaussovskou rozptylovou funkcí s přičteným šumem. Jak vidíte, tak Richardson-Lucyho algoritmus není příliš odolný vůči šumu a snímek vyjde degradovaný více než byl původně. Abych byl fér, tak pro tento příklad jsem použil vysoký počet iterací, abych zvýraznil tento efekt, ale když se podíváme na výsledky pro regularizované verze pro stejný počet iterací, tak ty výsledky jsou mnohem lepší. Nevýhoda L2 regularizace je že i když potlačuje šum, tak zase trochu rozmazává ten náš snímek, takže jde proti tomu, čeho se snažíme docílit, oproti tomu L1 regularizace zachovává hrany, ale zase vyhlazuje jemné struktury, které jsou blízko hranici šumu.

**Slide 8,9,10**

Jak jsem říkal předtím, tak pro dekonvoluci musíme znát rozptylovou funkci. Tu není úplně jednoduché získat, ale zde máme tu výhodu, že můžeme předpokládat, že všechny difrakční peaky by v ideálním světě měly být bodové zdroje, takže můžeme brát rovnou ty jako rozptylovou funkci. Tady na obrázku vlevo je rozptylová funkce získaná z centrálního maxima v difraktogramech, opět nám to tu trochu kazí šum, takže můžeme využít například jádrový odhad, abychom získali hladkou funkci. Zde vidíme, jak funguje tohle vyhlazování. Další možností je nafitovat nějakou vhodnou distribuci, tady máme například fit 2D normálního rozdělení, ale experimentujeme i s fitování Voigtova profilu, což je konvoluce Gaussova a Lorezntzova rozdělení. Z teorie se dá ukázat, že difrakční maxima by měla mít právě Voigtovo rozdělení, ale v praxi se výsledky moc neliší od Gaussovského fitu.

**Slide 11,12,13**

Nyní se můžeme podívat jak vypadají samotné dekonvoluce práškových difraktogramů, zde máme jeden snímek z datasetu vybraného k sumaci, práškový difraktogram bez dekonvoluce, a poté výsledky pro dekonvoluci s 10 iteracemi a s 50 iteracemi. Pro lepší představu, co se během dekonvoluce děje zde máme vykreslenou závislost intenzity na vzdálenosti od středu snímku, součet bez dekonvoluce je zobrazený černě, ostatní barvy jsou potom pro 10,30 a 50 iterací. Co je zajímavé tak mezi 30 a 50 iteracemi není prakticky žádný rozdíl. To podstatné ovšem je že dekonvolucí získáme ostřejší maxima. To je ostatně vidět i zde, tady jsou vykreslené intenzity pro difraktogram bez dekonvoluce a s 80 iteracemi. Tady je vidět že zhruba 80 pixelů od středu je ještě jedno lokální maximum které by se bez dekonvoluce kompletně ztratilo.

**Slide 14**

Ještě bych zde rád demonstroval, jak důležitý je dobrý odhad rozptylové funkce. Zde máme znovu vykreslené intenzity na vzdálenosti od středu, ovšem tentokrát pro 10 iterací Richardson-Lucyho algoritmu a pro různé odhady rozptylové funkce. Vidíme že jádrový odhad centrálního maxima funguje vcelku dobře, ale nafitovaná Voigtova distribuce nám vytváří falešná maxima blízko středu. Tenhle efekt je známý jako ringing a je typický pro dekonvoluce

**Slide 15,16,17**

Nakonec bych rád zmínil, jaký je plán do budoucna, my bychom se totiž časem chtěli odprostit od dekonvolucí. Rádi bychom byli schopni detekovat difrakční maxima, zaznamenat jejich polohu a intenzitu a rekonstruovat práškové difraktogramy z toho. K tomu budeme chtít využít segmentační neuronovou síť, a pro tu potřebujeme velké množství dat. Aktuálně plánujeme trénovací data nasimulovat, a to tak že si vygenerujeme snímek s centrálním maximem, přidáme do předem definovaných vzdáleností od středu nahodile difrakční maxima, přičteme neuniformní pozadí a přičteme šum. Výsledkem jsou simulované difraktogramy, můžete porovnat zde, vlevo jsou simulovaná data, vpravo je vzorek z naměřených dat. My bychom pak mohli natrénovat neuronovou síť pro segmentaci difrakčních obrazců i z velmi silně zašuměných snímků a neomezovala by nás dekonvoluce.