

Obrázek 1: Ukázkový snímek z videa

Millikanův experiment – automatické zpracování obrazu

1. Zpracování dat

1.1. Zpracování videí

Ukázkový snímek z jednoho z videí lze vidět na obrázku 1. Ruční sledování kapek ve videu by zabralo mnoho času, proto máte k dispozici jednoduchý program na automatické trackování kapiček, jehož autorem je Tomáš Vítek. Program je napsán v jazyce Python.

Program pro trackování videa se jmenuje `kapkatrack.py` a nápovědu k jeho použití lze získat příkazem `python kapkatrack.py -h`. Program je možné spustit s následujícími parametry:

- `--min-B` - minimální hodnota jasu pro detekci kapky
- `--max-B` - maximální hodnota jasu pro detekci kapky (nejlepší nechat na 255)
- `--brightness` - míra změny jasu jasu původního videa (1 = původní)
- `--contrast` - míra změny kontrastu původního videa (1 = původní)
- `--blur` - míra rozmazání původního videa (0 = vůbec, musí být celočíselné)

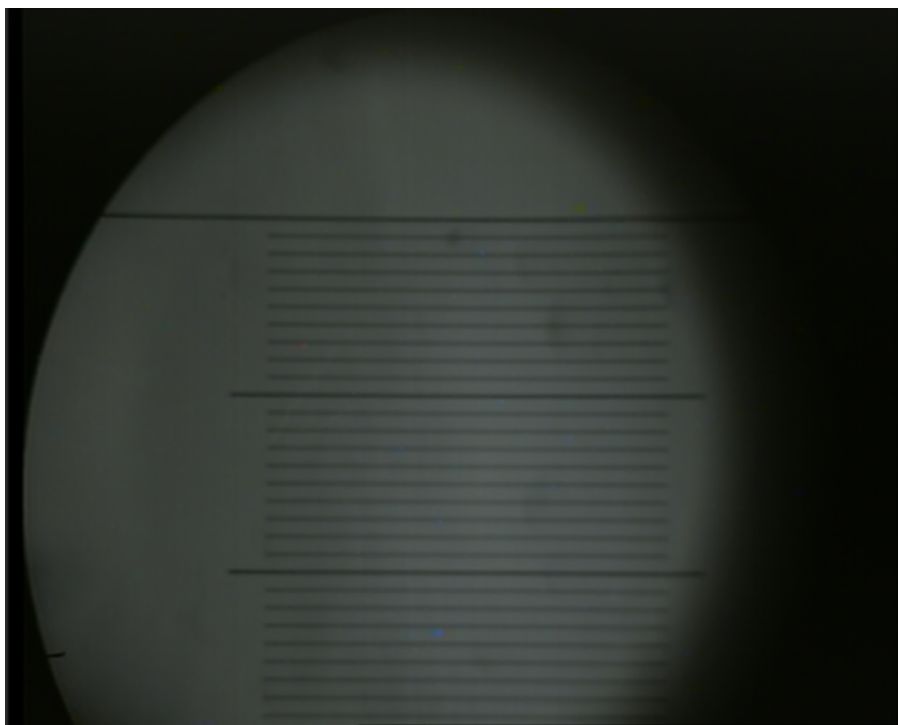
Parametry jsou dobrovolné a při jejich nezadání se použijí výchozí hodnoty. Výjimkou je parametr `--min-B`, při jehož nezadání se na počátku objeví okno, ve kterém si uživatel parametr sám zvolí.

Součástí příkazu musí být samozřejmě i cesta k videu/videím, příklad spuštění tedy může být `python3 kapkatrack.py -min-B 100 -contrast 1.5 -brightness 1.2 data/vids/*mpg`

Aby bylo sledování kapiček o něco spolehlivější, po jejich detekci pomocí minimální/maximální hodnoty je každé přiřazen tracker z knihovny `opencv`, kterou při práci s videem používal a kterou je třeba mít nainstalovanou (v pip balíčky `opencv-python` a `opencv-contrib-python`).

Program postupuje tak, že nejprve udělá průměr všech snímků videa, čímž získám jeho pozadí (ukázka na obrázku 2) a pak u každého snímku toto pozadí odečte (zkázka na obrázku 3).

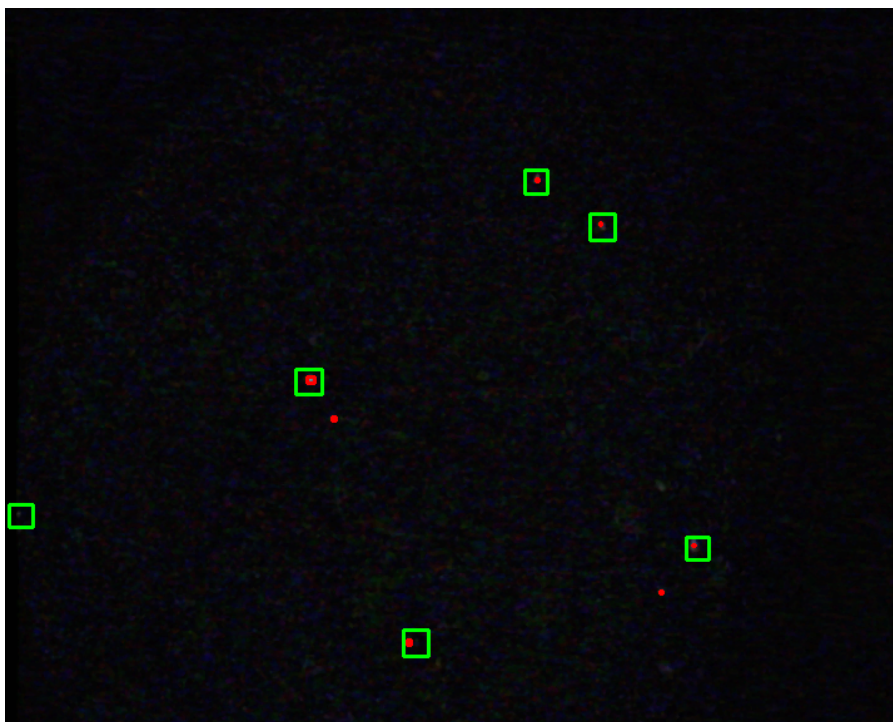
Poté už dochází k samotnému vyhledání a trackování kapiček, ukázku je možné vidět na obrázku 4. Souřadnice trackovaných kapiček jsou poté uloženy do souborů `[puvodni_video]_x.csv` pro souřadnici `x`, podobně pro `y`.



Obrázek 2: Ukázka zprůměrovaného pozadí videa



Obrázek 3: Ukázka videa po odstranění pozadí



Obrázek 4: Ukázka trackování kapiček ve videu. Hodnoty jasu v nastavených mezích pro detekci jsou vyznačeny červeně, trackovaná oblast pomocí opencv trackerů je vyznačena zeleně.

1.1.1. Zpracování získaných souřadnic

Na zpracování závislosti souřadnic kapiček na čase z předchozí části lze použít další program v pythonu, pojmenovaný `kapkal.py`. Ten nejprve trajektorie kapiček trochu vyhladí pomocí `np.convolve` (opraví tak i některé chyby trackeru), poté se pokusí najít snímek, ve kterém nejvíce kapiček změnilo směr své rychlosti (tj. snímek, ve kterém došlo k převrácení napětí). Nakonec vybere jen ty kapičky, které v tomto snímku v ose y rychlost skutečně převrátily, pro ty je pak z obou stran proveden lineární fit přímé části trajektorie od a do okamžiku otočení a jsou tak získány jejich dvojice rychlostí.

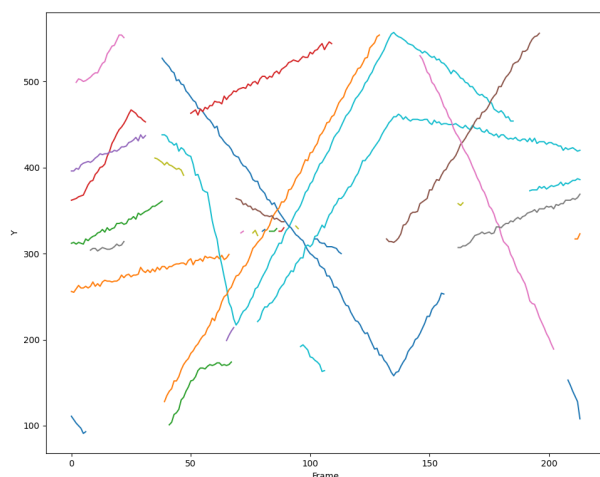
Nakonec jsou ještě z výsledných dat vyřazeny všechny kapičky, které mají v x -ové ose v intervalu fitu příliš velkou směrodatnou odchylku. Ta totiž může naznačovat, že v trackeru došlo k nějaké chybě, nebo že se kapička z nějakého důvodu (např. pohyb vzduchu) příliš rychle pohybovala ve směru osy x .

Na obrázku 5 je ukázka výstupních dat kapkatrackeru, který je zároveň vstupem kapkalu. Jde na něm vidět, že většina zachycených trajektorií kapiček je pro nás irelevantní, a je potřeba vybrat pouze ty lineární, jejichž kapičky navíc byly nabitě a v okamžiku převrácení napětí tak otočily svou rychlost do opačného směru. Na obrázku 6 jsou zobrazeny trajektorie právě takových kapiček, které byly vybrány z předchozího výstupu trackeru a bylo vyhodnoceno, že prochází okamžikem obrácení napětí. Pro ně byly obě části trajektorie před a po změně napětí nafitovány přímkami a z nich určena rychlost (v pixelech za snímek), která poté byla uložena do souboru `[puvodni_video]_drops.csv`.

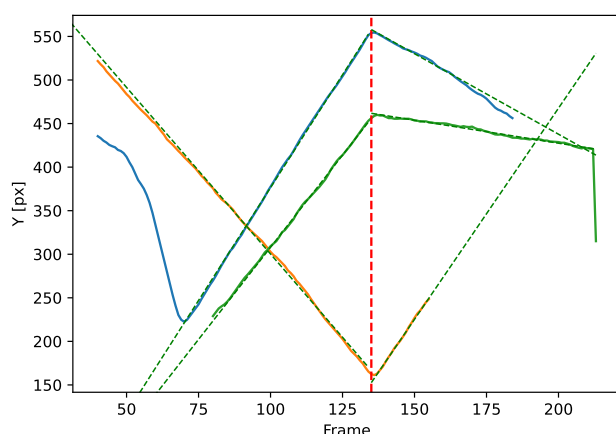
Mělo by být poznamenáno, že detekce okamžiku převrácení napětí je zcela automatická, stejně jako nalezení správné části trajektorie pro fit rychlosti kapky, jediným potřebným vstupem od uživatele může být doladění níže uvedených parametrů kapkalu, aby lépe fungoval na daný set videí.

Stručnou dokumentaci všech parametrů programu `kapkal.py` je možné získat spuštěním programu s parametrem `-h`. Pro přehlednost jsou uvedeny i zde:

- `--fit-std-threshold` - Maximální povolená směrodatná odchylka rychlosti kapičky (slouží i na určení části trajektorie kapičky pro fit)
- `-s` nebo `--speed` - Doba v sekundách, po kterou se při vyhodnocování ukazuje každý graf. Při hodnotě 0 se grafy nebudou zobrazovat vůbec.
- `--smooth-factor` - Počet framů, přes které probíhá konvoluce při vyhlazování trajektorie kapičky.



Obrázek 5: Ukázka výstupu dat trackeru (trajektorií kapiček) v y ose při vstupu do programu `kapkal.py`.



Obrázek 6: Ukázka profiltrování a fitu trajektorií kapiček v y ose. Fit je vyznačen zelenou přerušovanou čarou. Červená přerušovaná čára značí detekovaný okamžik obrácení napětí.

- `-f` nebo `--min-fit-frames` - Minimální počet framů, přes které může být fitována rychlost kapičky (pokud se nepodaří, kapička je vyřazena)
- `--inversion-offset` - Počet framů od detekce okamžiku obrácení napětí, přes které se vyhodnocuje rychlost kapičky. Okamžik obrácení napětí nemusí být vždy detekován dokonale přesně a trajektorie kapičky nemusí být bezprostředně po obrácení ustálená, tato hodnota je tedy lepší nenulová.
- `-x` nebo `--x-std-threshold` - Maximální povolená směrodatná odchylka rychlosti kapičky v ose x v úseku fitu rychlosti. Pokud je přesažena, kapička bude vyřazena.
- `--save-plots` - Pouze přepínač bez parametru. Když je přítomen, výstupem bude kromě csv souborů s rychlostmi kapiček také graf trajektorií kapiček v y ose se zobrazenými fity (např. obrázek 6).

I když se to jeví jako hodně parametrů, u většiny z nich by mohlo stačit použít defaultní hodnoty, příklad příkazu pro analýzu dat tedy může být tak jednoduchý jako `python3 kapkal.py data/vids/*.mpg`.