Warszawa 12 I 2017

Tomasz Jakubczyk

Opiekun pracy:
dr hab. Andrzej Stachurski

Raport
z Pracowni Dyplomowej Inżynierskiej 1
semestr 16Z

1. Wstęp – temat i cel pracy

Tematem pracy jest „Minimalizacja odległości płaskiej krzywej zamkniętej generowanej przez 6-parametrową procedurę od obrazu rozproszeniowego Mie'go”. Celem tej pracy jest wytworzenie programu wspierającego użytkownika oprogramowania **PikeReader** służącego do analizy wyników eksperymentu fizycznego. Oprogramowanie to jest omówione szczegółowo w pracy [1], której jestem współautorem.

Eksperyment fizyczny prowadzony w IF PAN w Laboratorium Nanocząstek i Klasterów Zespołu Spektroskopii Laserowej wykorzystuje jako metodę pomiarową rozpraszanie światła laserowego na kropli cieczy lewitującej w pułapce elektrodynamicznej. Rozproszone na kropli światło rejestrowane jest za pomocą matrycy CCD poprzedzonej układem optycznym. Do opisu rozpraszania światła na sferycznych kroplach służy teoria Mie'go. Rozwiązując zagadnienie odwrotne dla każdego obrazu rozproszeniowego można znaleźć promień oraz współczynnik załamania kropli. Tym samym, analizując w ten sposób sekwencję obrazów (nagrany film) można znaleźć ewolucję promienia i współczynnika załamania. Do rozwiązywania zagadnienia odwrotnego służy inne oprogramowanie napisane w Laboratorium, działające obecnie na platformie CUDA, dla którego PikeReader przygotowuje dane.

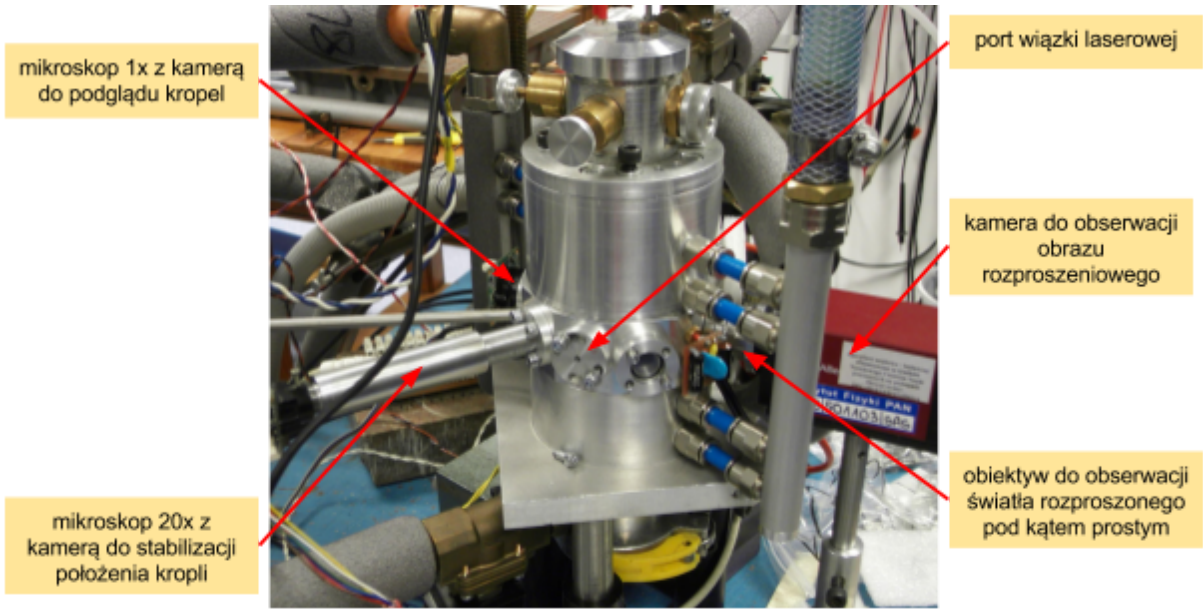

Fig. 1 Zdjęcie komory klimatycznej do obserwacji rozpraszania światła pod kątem prostym.

Układ optyczny jest bardzo prosty. Generalnie składa się z dwóch takich samych soczewek i umieszczonego między nimi kompozytowego polaryzatora foliowego. Jego przednią aperturą jest okienko pułapki. Kropla znajduje się w ognisku pierwszej soczewki. Do celów Ray Tracing’u może być traktowana jako punktowe źródło światła. Obraz powstający na matrycy CCD nie jest obrazem kropli lecz obrazem rozproszeniowym Mie’go (skaterogramem). Ponieważ użyto tylko dwóch zwykłych soczewek sferycznych (z warstwami przeciwodblaskowymi), układ jest źródłem dużej aberracji sferycznej dla

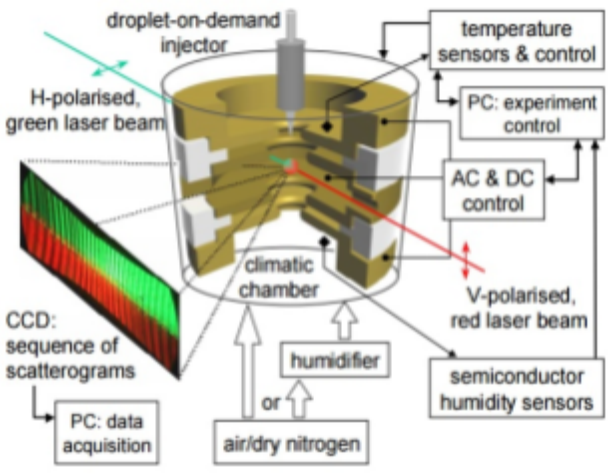

Fig. 2 Widok pułapki z jej częściowym przekrojem i schemat układu pomiarowego.

promieni pozaosiowych i znacznej aberracji chromatycznej. Jego zaletą jest jednak duży kąt widzenia przy bardzo zwartej budowie. Informacja zawarta w obrazie nie jest tracona skutkiem aberracji i można ją odzyskać poprzez odpowiednie przetwarzanie obrazu.

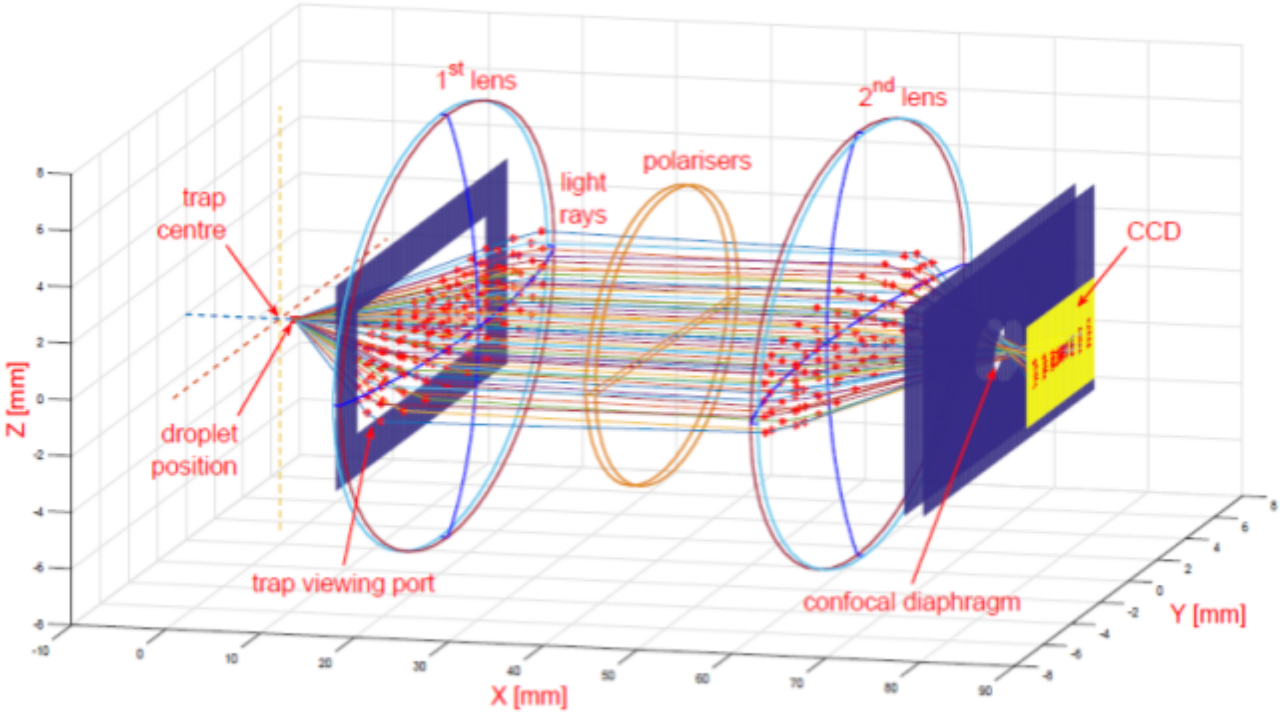

Fig. 3 Bieg promieni poprzez istotne elementy układu optycznego

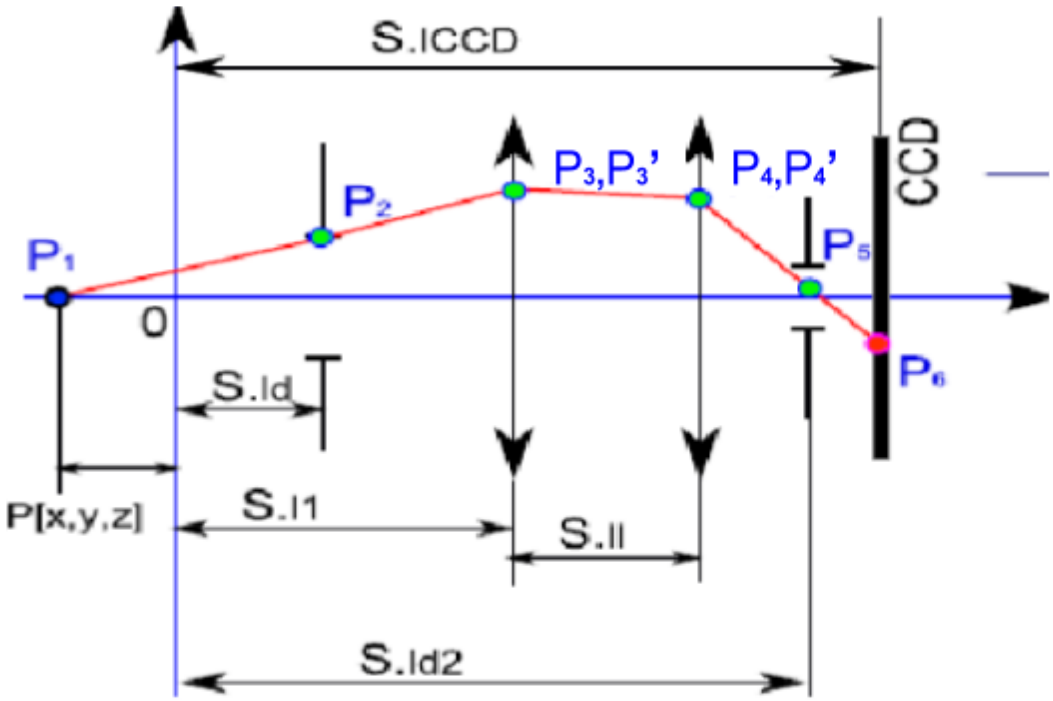

Fig. 4 Schemat przekroju układu optycznego. Punkty opisują kolejne węzły ray tracing'u: P_1 – położenie kropli, P_2 – punkt w obszarze okienka pułapki, P_3,P_3' oraz P_4, P_4'– punkty na powierzchniach łamiących soczewki, P_5 – punkt w obszarze przesłony konfokalnej, P_6 – punkt na matrycy CCD

Kształt obrazu otrzymywanego na matrycy CCD operacyjnie definiowany jest przez kształt jego obwodu. Jest on projekcją cienia rzucanego przez brzegi okienka w pułapce pod wpływem

oświetlenia przez światło rozproszone na kropli. Zależy on przede wszystkim od kształtu okienka, od położenia soczewek względem kropli i względem siebie, a także od położenia CCD. W kolejnych eksperymentach pozycja kropli względem środka pułapki może być zmienna i zazwyczaj jest, natomiast pozycja matrycy CCD zależy od ustawienia kamery w układzie doświadczalnym i zmienia się rzadziej. W istniejącym już oprogramowaniu dostępna jest funkcja, która przy pomocy Ray Tracing'u propaguje teoretyczny kształt okienka na matrycę, dostarczając w ten sposób swoistego celownika.

Oprogramowanie **PikeReader,** napisane w Matlabie, dostarcza celownika ręcznie ustawianego za pomocą 6 parametrów – pozycji kropli i matrycy CCD. Ręczne ustawianie parametrów jest jednak dość trudne i czasochłonne. Moduł automatycznie znajdujący te parametry może znacząco poprawić wygodę i szybkość analizowania wyników eksperymentu.

2. Przeprowadzone badania problemu

W celu zrozumienia problemu przeczytałem kod istniejącego już oprogramowania i zapoznałem się z teorią stojącą za eksperymentem w zakresie potrzebnym do pracy nad zagadnieniem.

Po przeanalizowaniu zebranych informacji postanowiłem sprawdzić jak dostępna funkcja fizycznego ray tracingu poddaje się optymalizacji. Po napisaniu odpowiedniej funkcji celu, solverom Matlaba rzeczywiście udało się uzyskać zadowalające wyniki dla ręcznie zaznaczonych punktów brzegowych.

Okazało się, że kod rysujący celownik jest złożeniem kilku funkcji (procedur) dość chaotycznie połączonych w pliku z wywołaniami (callbacks) interfejsu graficznego.

Dostępne procedury złożyłem w jedną funkcję generującą celownik na podstawie danych 6 parametrów. Funkcja ma postać f(PkX,PkY,PkZ,CCDX,CCDY,CCDZ): R^6 -> R^2^n

Wynikiem wywołania funkcji z 6 argumentami jest n (n=80) punktów na płaszczyźnie. Oznacza to, że celownik przybliżony jest krótkimi odcinkami. Według opinii pracujących przy doświadczeniu fizyków nie są znane sposoby tworzenia analitycznych transformacji (jakobianów) przejścia z niezaberrowanego układu współrzędnych do układu zaberrowanego. Tym samym, choć znane są funkcje analityczne opisujące kształt brzegu okienka, nie można znaleźć analitycznego kształtu celownika. Typową metodą szukania obrazów tworzonych przez układy optyczne jest metoda macierzy ABCD (2x2), będąca formą analitycznego ray tracing'u. Działa ona jednak tylko dla optyki przyosiowej (~+/- 5° od osi optycznej) i w zasadzie osiowo-symetrycznej. Przyosiowa analiza układów nieosiowych wymaga już stosowania macierzy 6x6 [2]. Istnieją też metody macierzowe dla optyki pozaosiowej [3], podobnie jak inne klasyczne metody analizy aberracji [4]. Wszystkie one są jednak bardzo skomplikowane obliczeniowo i ich stosowanie jest dużo trudniejsze niż użycie klasyczego raytracingu, nie przynosząc żadnych korzyści. Nie ma zatem powodu ich stosowania.

Solvery Matlaba **fminsearch** (metoda simplex) i **fminunc** (metody gradientowe) potrafią optymalizować procedurę tak jak funkcję analityczną za pomocą próbkowania funkcji (scan lines).

Wybór punktów brzegowych pozornie wydaje się prosty, ale żeby dokładnie znaleźć granicę światła i cienia należy rozpoznać dyfrakcję na brzegu okienka. Światło laserowe rozproszone na kropli pozostaje spójne i kiedy dociera do brzegu okienka ulega dyfrakcji. Na obrazie brzegu okienka uzyskiwanym na matrycy CCD, na brzegu cienia, powstaje wzór dyfrakcyjny. W kierunku prostopadłym do brzegu celownika wzór dyfrakcyjny można opisać całką Fresnela. Do rozpoznania położenia brzegu wystarczy jednak wskazać punkt po stronie cienia w takiej odległości od pierwszego maksimum dyfrakcyjnego jak odległość pierwszego maksimum i pierwszego minimum.

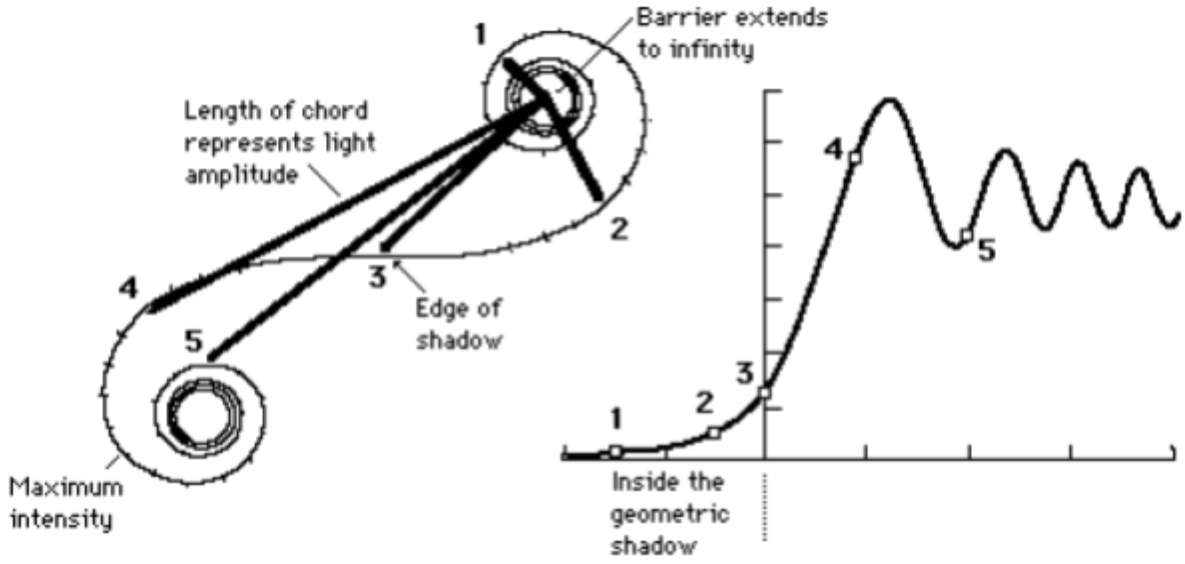

Fig. 5 Dyfrakcja na krawędzi, spirala Cornu, całka Fresnela

Istnieje wiele metod znajdowania krawędzi na obrazach. Popularnością cieszy się metoda Cannego. Niestety, ze względu na występowanie zjawisk falowych oraz duże zmiany natężenia światła dla różnych kątów padania powoduje, że dostępne metody znajdowania krawędzi nie dają zadowalających wyników. Metoda Cannego zależnie od ustawień znajduje wiele krawędzi w środku obrazu ze względu na występujące prążki interferencyjne, oraz nie znajduje krawędzi po lewej i prawej stronie obrazu gdzie natężenia światła są znacznie mniejsze. Problem wyrównywania jasności obrazu jest rozwiązywany w aplikacji PikeReader za pomocą fizycznego ray tracingu przez obliczenie teoretycznego rozkładu natężenia światła, ale wymaga właśnie dokładnego podania pozycji kropli i matrycy CCD. Te z kolei znajdujemy przez dopasowanie celownika do brzegu obrazu. Mamy zatem pętlę w zagadnieniu. Wymagałoby to rozwiązywania iteracyjnego.
Innym problemem jest to, że algorytm Cannego ignoruje wzór dyfrakcyjny na brzegu.
Zdecydowałem się najpierw znaleźć przybliżone położenie celownika poprzez maksymalizowanie liczby punktów jasnych w jego środku i ciemnych na zewnątrz. Po znalezieniu okolic brzegu obrazu postanowiłem przeszukać pixele wzdłuż linii (odcinków) prostopadłych do ramki celownika w celu znalezienia pierwszego maksimum i pierwszego minimum dyfrakcyjnego, a następnie wyznaczyć geometrycznie punkty brzegowe.

3. Zaimplementowana aplikacja

W wyniku prowadzonych badań zdołałem napisać w Matlabie funkcję, którą można wykorzystać jako moduł do aplikacji **PikeReader**. Funkcja **BorderRecognition** przyjmuje na wejściu obraz 640x480 i opcjonalne parametry początkowe (punkt początkowy), a zwraca na wyjściu szukane parametry pozycji kropli i matrycy CCD. Obraz wejściowy dla uzyskania lepszej jakości danych zwykle złożony jest z sumowanych klatek z filmu.

W pierwszej kolejności napisałem program, w którym można ręcznie zaznaczyć punkty brzegowe. Minimalizowana funkcja celu wyznaczała średniokwadratową odległość celownika od ręcznie zadanych punktów dla 6 argumentów. Wyniki okazały się zadowalające i postanowiłem napisać program, który sam rozpoznawałby brzeg obrazu bez udziału użytkownika.

Zdecydowałem się napisać funkcję skalaryzującą **BrightnesScalarization** maksymalizującą liczbę

punktów jasnych wewnątrz ramki i ciemnych na zewnątrz. Szybko okazało się, że nie jest oczywiste jak odróżnić punkty jasne od ciemnych. Napisałem funkcję TresholdValue która znajduje taki próg, że zachowana jest proporcja punktów jasnych do ciemnych (~0.3). Wadą tego rozwiązania jest ignorowanie faktycznego położenia brzegu, co wobec natury obrazu rozproszeniowego daje rozwiązanie zbyt przybliżone.

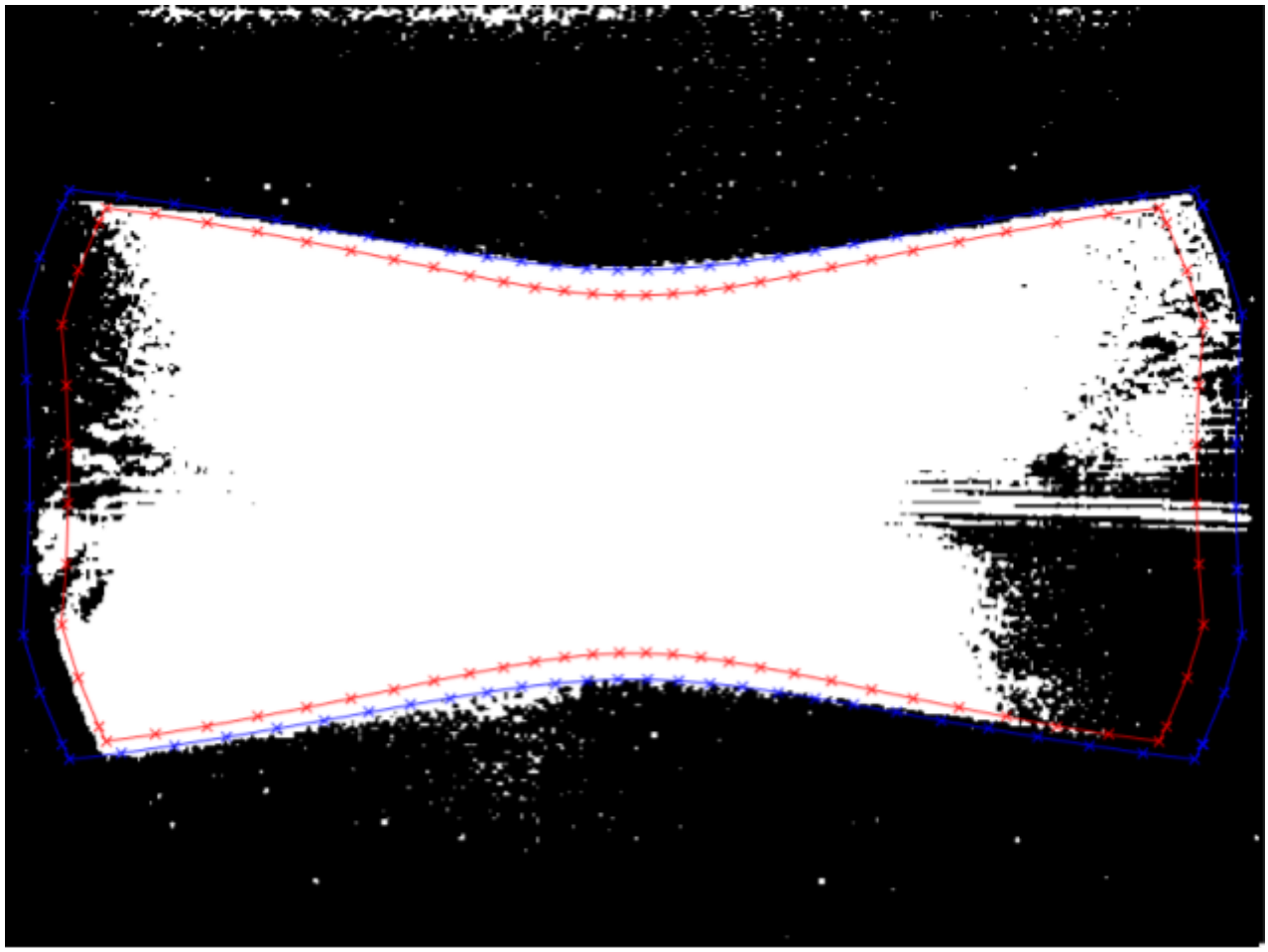

Fig. 6 Dopasowanie celownika do obrazu zbinaryzowanego.

W celu uzyskania dokładniejszych wyników napisałem funkcję **FindBorderPoints** znajdującą punkty brzegowe na podstawie analizy wzoru dyfrakcyjnego dla wszystkich wygenerowanych punktów brzegu i zwracającą najlepsze z nich.

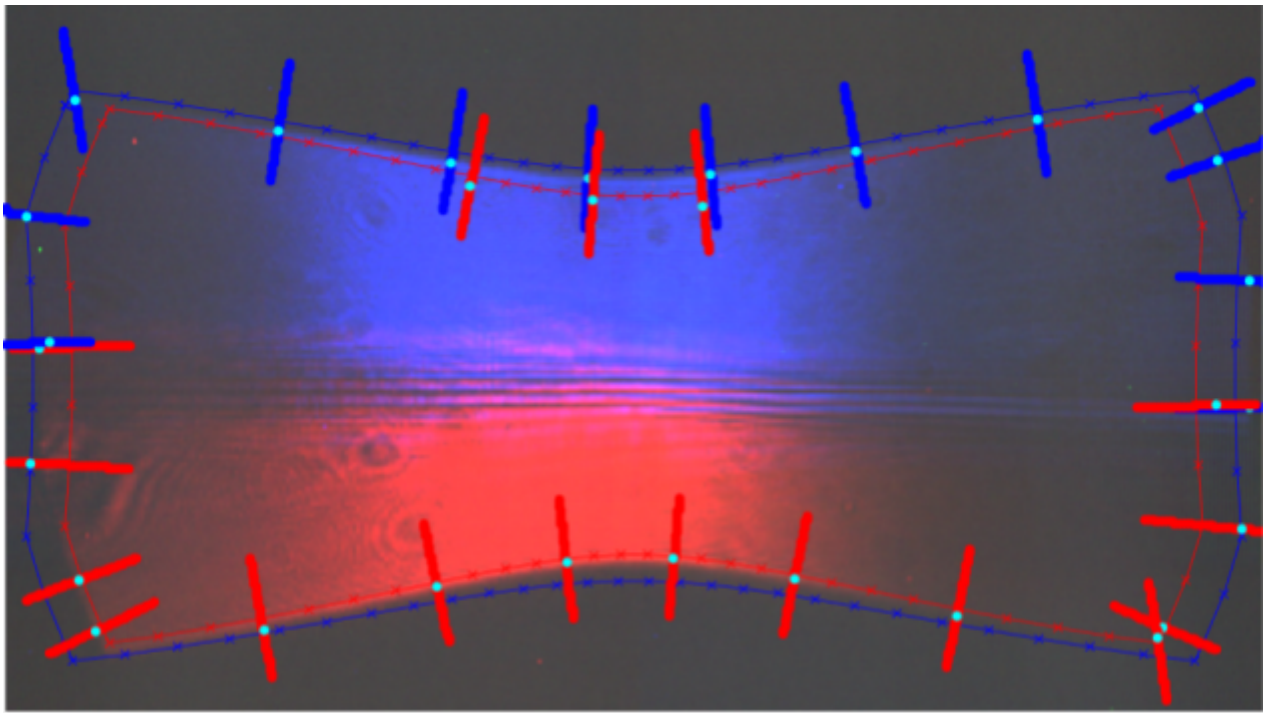

Fig. 7 Punkty granicy światła i cienia na przeszukiwanych odcinkach wybrane do dokładnego poszukiwania brzegu.

W wybranych punktach celownika znajdowane są równania prostych prostopadłych do ramki. Za styczną do ramki w wybranym punkcie przyjmuję prostą równoległą do prostej przechodzącą przez lewego i prawego sąsiada wybranego punktu i przechodzącą przez wybrany punkt. Następnie znajdowane są współrzędne pixeli leżących na prostej prostopadłej do ramki w wybranym punkcie w zadanej odległości. Wybieranie współrzędnych pixeli na odcinku odbywa się za pomocą algorytmu Bresenhama. Do dalszego etapu wybieram odcinki na których występuje najwyższa dynamika wartości, czyli te z największymi różnicami między największą i najmniejszą wartością.

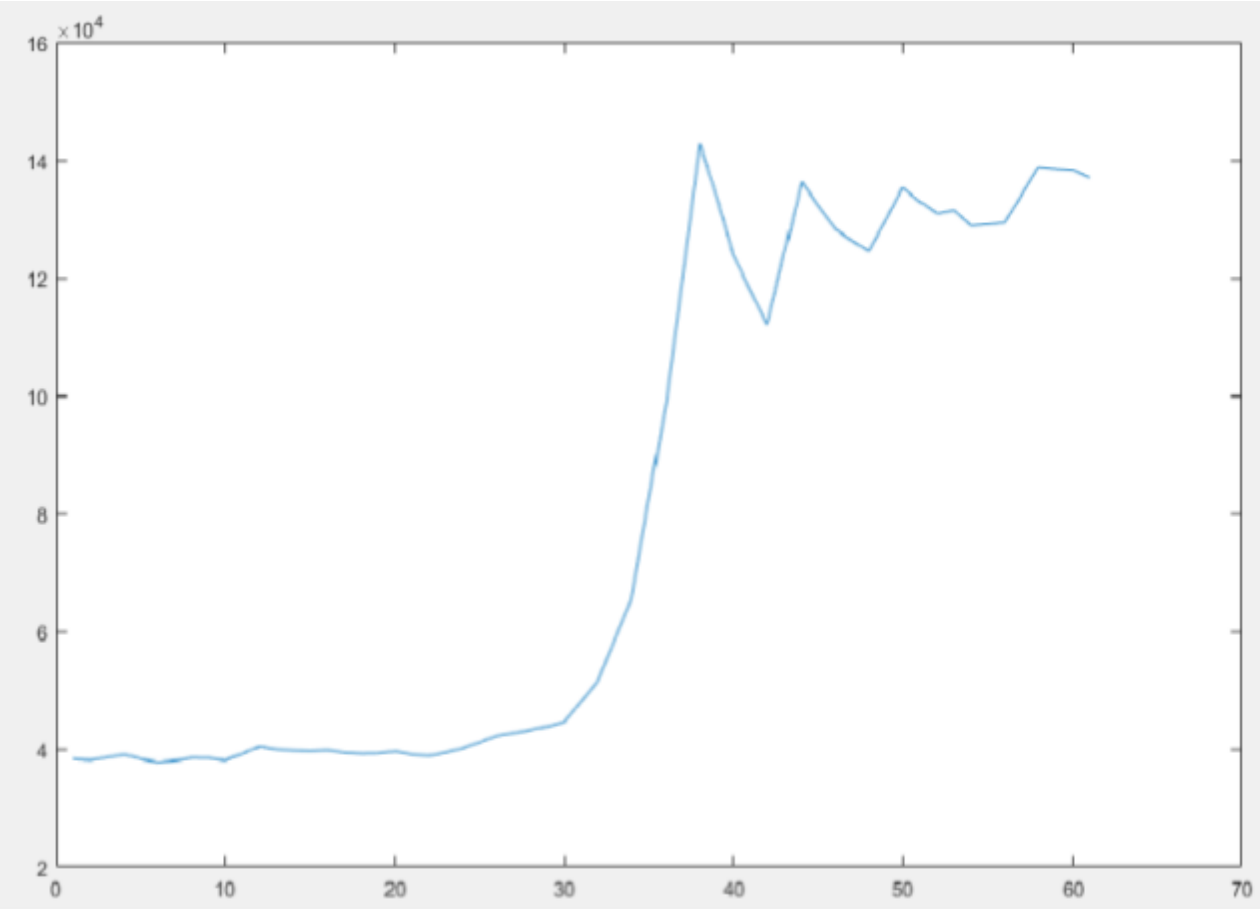

Fig. 8 Wzór dyfrakcyjny na przeszukiwanym odcinku.

Uzyskane w ten sposób odcinki przeszukiwane są w poszukiwaniu minimów i maksimów. Następnie wyliczam różnice wysokości między kolejnymi minimami i maksimami. Krzywa o największym skoku wartości najprawdopodobniej jest zboczem wzoru dyfrakcyjnego i na niej musi znajdować się granica światła i cienia. Maksimum na szczycie tej krzywej jest pierwszym maksimum dyfrakcyjnym, a następne minimum jest pierwszym minimum dyfrakcyjnym. Za pomocą tych danych wyliczana jest geometryczna granica światła i cienia: odległość 1-szego minimum od 1-szego maksimum jest przenoszona na stronę cienia przy pomocy wzoru na środek odcinka. Wszystkie wybrane punkty brzegowe są zwracane i wykorzystane do minimalizacji funkcji celu **MeanSquaredDistance**.

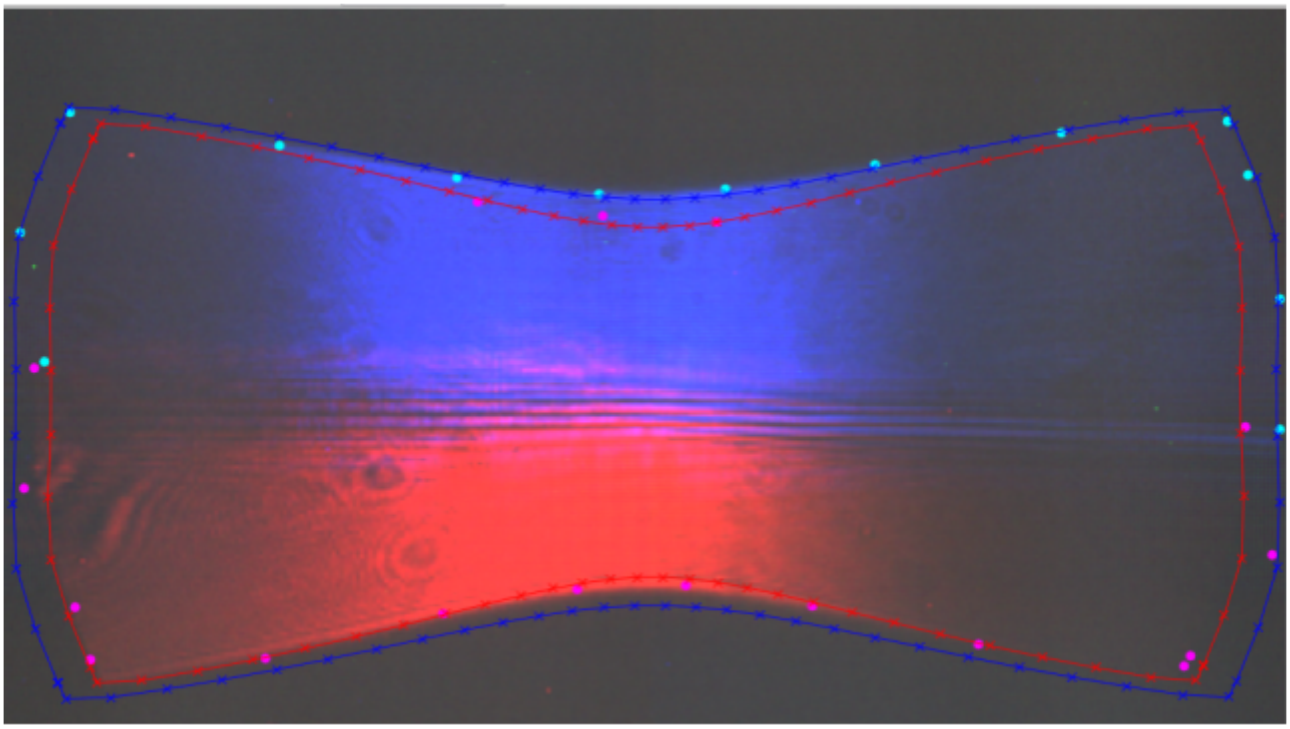

Fig. 9 Dopasowanie celownika do znalezionych punktów brzegu.

4. Dokumentacja

Omawiany kod znajduje się w sieciowym repozytorium GitHub po adresem: https://github.com/sigrond/BorderRecognition.
Wygenerowałem automatycznie dokumentację kodu za pomocą m2HTML. Dokumentacja dostępna jest na GitHubie i można ją znaleźć na stronie https://sigrond.github.io/BorderRecognition/ .

5. Literatura

1. G. Derkachov, T. Jakubczyk, D. Jakubczyk, J. Archer and M. Woźniak, Fast data preprocessing with Graphics Processing Units for inverse problem solving in light-scattering measurements, *Journal of Quantitative Spectroscopy and Radiative Transfer*, http://dx.doi.org/10.1016/j.jqsrt.2017.01.008
2. P.-D. Lin, C.-C. Hsueh, 6×6 matrix formalism of optical elements for modeling and analyzing 3D optical systems, Appl. Phys. B (2009) 97: 135. doi:10.1007/s00340-009-3616-7
3. W. Brower, Matrix Method in Optical Instrument Design (Benjamin, Elmsford, 1964)
4. A. E. Conrady, Applied Optics and Optical Design, Courier Corporation, 2014.