

Przemysław MAZUREK

ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY W SZCZECINIE, KATEDRA PRZETWARZANIA SYGNAŁÓW I INŻYNIERII MULTIMEDIALNEJ
26. Kwietnia 10, 71-126 Szczecin

Redukowanie czasu obliczeń transformaty Hougha z wykorzystaniem preselekcji implementowane na GPGPU

Dr inż. Przemysław MAZUREK

Adiunkt w Katedrze Przetwarzania Sygnałów i Inżynierii Multimedialnej Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie. Autor ponad 150 artykułów z zakresu cyfrowego przetwarzania sygnałów, estymacji ruchu, biopomiarów oraz przetwarzania obrazów biomedycznych.



e-mail: przemyslaw.mazurek@zut.edu.pl

Streszczenie

W artykule przedstawiono algorytm rzadkiego próbkowania okręgu dla transformaty Hougha, w celu preselekcji okręgów do dalszego przetwarzania. Rozwiązanie pozwala na skrócenie czasu obliczeń, rzędu dwóch razy dla implementacji z wykorzystaniem programowanej karty graficznej (GPGPU). Może ono być zastosowywane do wykrycia jednokolorowej kryzy wokół lustra sferycznego lub półsferycznego służącego do pomiaru oświetlenia. Dzięki temu można analizować obraz wideo zawierający pomiary z różnych obszarów.

Słowa kluczowe: Transformata Hougha, Próbki światła, Pomiary oświetlenia, GPGPU.

Reducing the computation time of the Hough Transform implemented on GPGPU using preselection of circles

Abstract

The Hough Transform (HT) is a very powerful algorithm for the detection of shapes. It is used for the detection of the flange of a light probe (Fig. 1). GPGPU processing is necessary for HT computation. The proposed algorithm (Fig. 4) is based on the preselection of circles for further computation (Fig. 3). Selection of the circles based on a limited set of pixel samples allows reduction of the processing time up to two times. The reduction is dependent on the image content, but the test shows that the influence of the number of the tested radiuses is less. About one second of the processing time is necessary, so larger images need tens of seconds for computation (Fig. 5). This algorithm uses exhaustive search over all positions and radiuses. One of the interesting applications of the HT is the estimation of the position and diameter of a circle related to the flange of a light probe [2, 3]. Light probes are used for computer graphics applications and they allow application of the real world measurements of light using the Image Based Lighting (IBL) technique. The GPGPU implementation use CUDA based code (CUDA v4.0) [10,11] and 16x8 organization of threads is assumed. The technique proposed for the time and spatial synchronization of the multiple threads for more synchronous read operation from a global memory is applied [7]. The global memory is large and slow, so a GPGPU-global memory interface is the main bottleneck of the system. Synchronization of accesses allows coalescence read operations.

Keywords: Hough Transform, light probe, light measurements, GPGPU.

1. Wprowadzenie

Transformata Hougha (HT – ang. Hough Transform) jest algorytmem wykorzystywanym do detekcji kształtów [1], także zaszumionych. Kształty są definiowane jako trajektorie, dla których wykonuje się całkowanie wartości obrazu. Najczęściej wykorzystywana jest ona do detekcji linii oraz okręgów. HT estymuje dopasowanie kształtu, położenie, orientację oraz skalę, w zależności od realizacji. Algorytm jest bardzo wymagający obliczeniowo, ponieważ testowane są wszystkie przypadki. Estymacja paramet

ów okręgu jest bardziej wymagająca, ponieważ konieczne jest analizowanie okręgów o różnych promieniach. Wykorzystanie GPGPU (programowalnych kart graficznych), odpowiedniego algorytmu i właściwej implementacji pozwala na analizę obrazu, w tym obrazu wideo, także w czasie rzeczywistym.

2. Próbki oświetlenia

Algorytm estymacji parametrów okręgu powinien wykorzystywać wiedzę aprioryczną dotyczącą analizowanego obrazu, co pozwala na skrócenie czasu obliczeń. Dedykowany danej aplikacji algorytm ma możliwość bardziej efektywnego jakościowo lub czasowego działania.

Jednym z interesujących zastosowań HT w analizie obrazu jest estymacja obszaru, w którym znajduje się próbnik światła. Zwykle jest to sferyczne lub półsferyczne lustro, odbijające oświetlenie otoczenia [2, 3]. Estymacja ta jest nowym zagadnieniem badawczym, dlatego ilość odniesień literaturowych jest niewielka. Ekstrakcja tego obszaru, będącego kołem, a następnie transformacja do postaci półsfery umożliwia wykorzystanie informacji o wektorach światła na potrzeby grafiki i animacji komputerowej. Technika ta to IBL (ang. Image-Based Lighting) – oświetlenie sceny przez obraz, który jest filtrem, znajdującym się na sferze wokół sceny, i gdzie światło spoza sfery wchodzi do jej środka, podlegając zmianie jasności i barwy [2, 3]. Techniki pozyskiwania obrazów sferycznych (panoram) jest wiele, (np. seria zdjęć obiektywem szerokokątnymi, rybie-oko). Lustrzany próbnik można przemieszczać i realizować pomiar w różnych miejscach [12-14]. Na rys. 1 przedstawiono półsferyczny próbnik oświetlenia z niebieską kryzą. Rozmiar próbniaka może być różny, od kilku cm do np. 60 cm.



Rys. 1. Próbki oświetlenia
Fig. 1. A light probe

Próbki z kryzą zawiera jednolitej barwy pierścien, który może być wykryty za pomocą metod kluczowania chromatycznego obrazu, a także metodami estymacji kształtu [4, 5]. Niestety, kluczowanie zawiedzie, jeśli na obrazie będzie podobna barwa w innych obszarach. Konieczne jest wtedy wykorzystanie innej barwy, co komplikuje cały proces. Uzyskanie niezależności jest możliwe, poprzez zastosowanie metod cyfrowego przetwarzania obrazu, praktycznie dla wszystkich przypadków.

W [6] oraz [7] zaproponowano i rozwinięto metodę estymacji kryzy, z wykorzystaniem informacji na temat barwy oraz detekcji krawędzi kryzy, dzięki czemu w nowej przestrzeni 2D podkreślone są oba te aspekty w tych samych miejscach. Pozwala to na

zwiększenie jakości estymacji, która realizowana jest następnie z wykorzystaniem transformaty Hougha. Ponieważ w obrazie jest tylko jeden próbnik to zadaniem jest znalezienie maksimum w przestrzeni transformaty H:

$$H_{\max}(x, y, r) = \max H(x, y, r) \quad (1)$$

Transformata Hougha realizuje operacje:

$$H(x, y, r) = \sum_i I(x + r \cdot \sin(\alpha_i), y + r \cdot \cos(\alpha_i)) \quad (2)$$

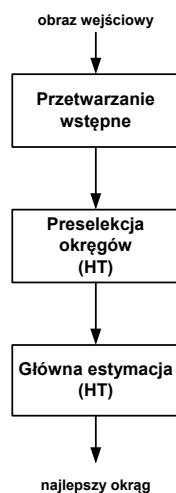
I – obraz,
 x, y – środek okręgu,
 r – promień,
 i – indeks kąta,
 α – kąt,

z określoną rozdzielczością dla położenia (x, y) oraz promienia r , co jest przyczyną znacznego kosztu obliczeniowego. W przypadku zastosowania inwersji [8] przestrzeni poszukiwań zadaniem jest znalezienie minimum:

$$H_{\min}(x, y, r) = \min_r H(x, y, r) \quad (3)$$

Rozwiązanie to jest o tyle dobre, że w trakcie akumulacji (3) można odrzucić okręgi, dla których wartość tymczasowa jest za duża.

Ponieważ okręgi są wyznaczane stopniowo, to możliwe jest zapamiętanie najlepszej (w tym przypadku najmniejszej wartości) i testowanie w trakcie obliczeń, czy nie została ona przekroczona. W przypadku implementacji równoległej z wykorzystaniem GPGPU, konieczna jest synchronizacja wyników, co nie jest proste technicznie, zwłaszcza dla starszych kart. W [8] wykorzystano wyznaczanie HT w dwóch etapach. W pierwszym realizowana jest pewna liczba kroków (kolejnych punktów na okręgu), po czym porównywane są wyniki dla kilkudziesięciu czy kilkuset przypadków z najlepszym dotychczasowym. Jeśli wszystkie wartości są większe, to drugi krok nie jest wykonywany i następuje odrzucenie wszystkich obecnie analizowanych. Jeśli jeden jest mniejszy, to obliczenia dla reszty okręgów kontynuuje się, a po ich zakończeniu ponownie sprawdza. Jeśli najlepszy wynik jest mniejszy od najlepszego, to staje się on najlepszym i zapamiętanym. Proces powtarza się dla kolejnych zestawów okręgów w celu wyłonienia najlepszego.



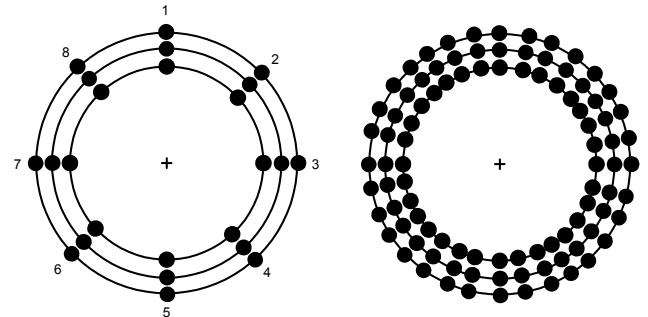
Rys. 2. Schemat przetwarzania
 Fig. 2. Schematic of processing

Ostatecznie wynik (położenie oraz promień) jest podstawą dla algorytmu optymalizacji gradientowej w celu dokładnego dopasowania subpikselowego.

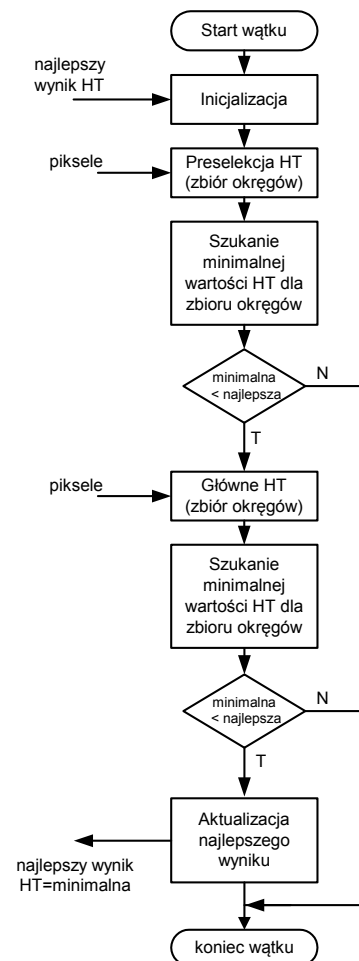
Metoda [8] pozwala na uzyskanie 16% redukcji czasu obliczeń, pozwala na opracowanie bardziej doskonałej techniki.

3. Preselekcja okręgów

Proponowana metoda preselekcji okręgów (rys. 2) bazuje na rzadszym próbkowaniu obrazu (rys. 3). Wybór okręgów, a dokładnie zestawu, dokonuje się podobnie jak dla metody [8] jak na rys. 4.



Rys. 3. Preselekcja i zwykłe próbkowanie HT (pokazano trzy okręgi)
 Fig. 3. Preselection and main HT sampling (three circles shown)



Rys. 4. Algorytm preselekcji dla HT
 Fig. 4. Algorithm of the preselection based HT

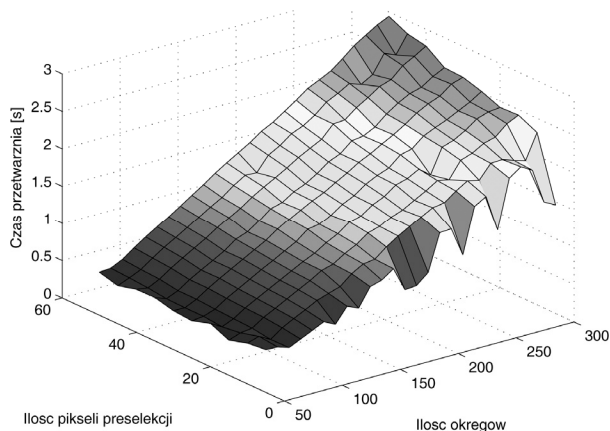
Wartość minimalna jest początkowo wybierana jako największa możliwa liczba. Algorytm nie próbuje okręgu w sposób optymalny, ponieważ ponownie te same punkty używane są podczas głównej estymacji. Okazuje się, że próba optymalizacji implementacji, poprzez wykorzystanie poprzedniego wyniku z etapu preselekcji w głównej estymacji nie pomaga, ponieważ kod jest bardziej złożony i efektywność czasowa maleje.

4. Implementacja oraz czas przetwarzania

Zasadniczą kwestią jest wpływ wyboru liczby punktów na okręgu. Zbyt duża prowadzi do zmniejszenia szybkości, mała natomiast wpływa na jakość pracy. Wyznaczenie minimalnej liczby punktów jest trudne, ponieważ zależy ona od zawartości obrazu. Z doświadczeń wynika, że powinno być to kilkanaście punktów. W przeciwnym przypadku testowanie w pierwszym etapie nie korzysta z wiedzy na temat okręgu, co może spowodować zły wynik w skrajnym przypadku, przez podjęcie decyzji na podstawie wartości tylko jednego piksela. Warto zauważyć, że jest to efekt szumów w obrazie, a akumulacja realizowana przez HT spełnia dodatkowo rolę filtru odszumiającego. Dlatego też ważne jest wykorzystanie większej liczby punktów podczas preselekcji.

Na rys. 5 przedstawiono czas obliczeń dla obrazu o rozdzielczość 512x512 pikseli. Wykorzystano kartę graficzną z procesorem G80 (NVIDIA GeForce 8800GTS). Oprogramowanie napisano z wykorzystaniem CUDA [10, 11] w języku C z rozszerzeniem CUDA. Dla tej karty testowano różne organizacje wątków w bloku. Najlepsza z nich to organizacja 16x8. Największym wyzwaniem jest taka implementacja, która efektywnie wykorzystuje dostęp do pamięci, ponieważ architektura CUDA ma szybkie procesory, ale mało pamięci wewnętrznej. Obraz przechowywany jest w tym przypadku w pamięci karty (pamięć globalna karty graficznej), ale nie samym GPU. Parametry interfejsu pamięci są głównym ograniczeniem wydajności algorytmu [10, 11].

W celu zwiększenia wydajności, wykorzystano niewielką pamięć podręczną (cache), znajdującą w środku GPGPU podczas adresowania pamięci globalnej (texture cache). Ponieważ wymuszono organizację wątków, taką, by próbkowanie obrazu było realizowane w sąsiednich adresach, uzyskuje się wysokich stopień trafień, a to daje możliwość przyspieszenia obliczeń.



Rys. 5. Czas obliczeń dla preselekcji HT
Fig. 5. Computation time for the preselection based HT

W implementacji wykorzystano niestandardowy przydział wątków od pikseli obrazu. Wątki przydzielane są do piksela, który stanowi środek zestawu okręgów. Wynikiem pośrednim dla danego bloku są synchronizowane przez porównanie wartości. Synchronizacja wyniku bloku jest realizowana za pomocą funkcji atomowych, gwarantujących niepodzielność operacji [10, 11]. Na karcie wykorzystanej do badań nie można zrealizować synchronizacji danych wątków za pomocą instrukcji atomowych, wewnątrz pamięci GPGPU. Konieczna jest dlatego synchronizacja przez pamięć globalną, co jest rozwiązaniem wolniejszym.

5. Wnioski

Liczba analizowanych okręgów wpływa w przybliżeniu liniowo na czas obliczeń. Inaczej jest z ilością punktów (pikseli) okręgu, które są równomiernie rozmieszczone. Dla małej ilości okręgów zysk nie jest zauważalny, jednak dla dużej liczby okręgów wyraźnie występuje zmniejszenie czasu obliczeń, nawet dwukrotnie. Pokazuje to, że zaproponowane rozwiązanie jest znacznie bardziej wydajne od rozwiązania [8], a także od zwykłej metody realizującej transformację Hougha.

Wyniki zależą także od zawartości obrazu. Badania wpływu zawartości samego obrazu na czas obliczeń, wymaga zastosowania metody Monte Carlo (testowania dla bardzo dużej liczby różnych obrazów zarejestrowanych w różnych warunkach). W praktyce należy spodziewać się, że zysk algorytmu jest zmienny, jednak czas obliczeń jest skrócony, względem zwykłej metody.

Realizacja estymacji w czasie rzędu sekund pozwala na analizę obrazu wideo, w którym położenie próbnika oświetlenia jest zmienne. Algorytm może być także wykorzystany do inicjalizacji algorytmu śledzenia próbnika oświetlenia [9].

Praca została wsparta z wykorzystaniem oprogramowania i sprzętu projektu UE EFRR ZPORR Z/2.32/1/1.3.1/267/05 "Badawczo-Dydaktyczne Centrum Nowoczesnych Technologii Multimedialnych Politechniki Szczecińskiej".

6. Literatura

- [1] Hough P.V.C., Arbor A.: Method and Means for Recognizing Complex Patterns, US Patent no. 3,069,654, 1962.
- [2] Debevec P.: Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography, SIGGRAPH 98, 189-198, 1998.
- [3] Debevec P.: Light Probe Image Gallery, <http://www.pauldebevec.com/Probes/>
- [4] Frejlichowski D.: An Experimental Comparison of Seven Shape Descriptors in the General Shape Analysis Problem. ICIAR 2010, Part I, Lecture Notes in Computer Science, vol.6111, 294-305, 2010.
- [5] Frejlichowski D.: Analysis of Four Polar Shape Descriptors Properties in an Exemplary Application, ICCVG 2010, Part I, LNCS vol.6374, 376-383, 2010.
- [6] Mazurek P.: Estimation of position of the light probe device for photorealistic computer animation purposes, Elektronika – konstrukcje, technologie, zastosowania, 42-44, R. LII no.1, 2011.
- [7] Mazurek P.: Circle parameters estimation using Hough transform implemented on GPGPU, Measurement Automation and Monitoring, vol.57, no.8, 886-898, 2011.
- [8] Mazurek P.: Optymalizacja transformaty Hougha dla okręgu z wykorzystaniem techniki odrzucenia implementowana na GPGPU, Measurement Automation and Monitoring, vol.58, no.7, 587-589, 2012.
- [9] Mazurek P.: Light estimation using light probe devices, Electrical Review, R.89 no. 3b/2013, 312-314, 2013.
- [10] NVIDIA CUDA C Programming Guide v.4.0, NVidia, 2011.
- [11] NVIDIA CUDA, CUDA C Best Practices Guide v.4.0, NVidia, 2011.
- [12] Reinhard E., Ward G., Pattanaik S., Debevec P.: High Dynamic Range Imaging. Acquisition, Display, and Image-Based Lighting, Morgan Kaufmann, 2005.
- [13] Unger J., Gustavson S., Ollila M., Johannesson M., A Real Time Light Probe, In Proceedings of the 25th Eurographics Annual Conference, Short Papers and Interactive Demos, 17-21, 2004.
- [14] Unger J., Gustavson S., Kronander J., Larsson P., Bonnet G., Kaiser G., Next Generation Image Based Lighting, SIGGRAPH 2011, Talk, Vancouver, Canada, August 7-11, 2011, <http://webstaff.itn.liu.se/~jonun/web/papers/2011-Siggraph/SiggraphTalk2011.pdf>