

Kamil STASZEK

AKADEMIA GÓRNICZO-HUTNICZA, KATEDRA ELEKTRONIKI, Al. Mickiewicza 30, 30-059 Kraków
EVATRONIX S. A., ul. Przybyły 2, 43-300 Bielsko-Biała

Opis i implementacja algorytmu wyznaczania rozkładu fazy obrazów prążkowych w układzie FPGA

Mgr inż. Kamil STASZEK

W 2011 roku ukończył studia magisterskie na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo – Hutniczej w Krakowie. Aktualnie jest doktorantem w Katedrze Elektroniki AGH. Główne zainteresowania naukowe dotyczą przetwarzania sygnałów cyfrowych oraz techniki mikrofalowej.



e-mail: kstaszek@agh.edu.pl

Streszczenie

Niniejszy artykuł przedstawia opis algorytmu wyznaczania rozkładu fazy z obrazów prążkowych. Jest to jeden z kroków przetwarzania obrazów stosowanych powszechnie w profilometrii optycznej, dzięki której na podstawie zdjęcia można odtworzyć trójwymiarowy kształt obiektu. Algorytm oparty został na zespolonej transformacji falkowej i pozwala na pełną automatyzację procesu skanowania. Duży nacisk został położony na optymalizację pod kątem sprzętowej implementacji w układzie FPGA. Wykazana została dobra odporność zarówno na szumy jak i niską jakość obrazu prążkowego.

Słowa kluczowe: obraz prążkowy, ekstrakcja fazy, transformacja falkowa, FPGA.

Hardware implementation of phase extraction algorithm in fringe pattern analysis

Abstract

This paper presents a phase extraction algorithm used in fringe pattern analysis. It is one of steps in optical profilometry process used to obtain full three-dimensional information about the measured object shape. The algorithm is based on the complex wavelet transform and allows full automation of the process. Since the fringe patterns represent non-stationary signals, application of time-frequency analysis provides better results than the commonly used Fourier transform. Six variants of the wavelet transform (one- and two-dimensional) were simulated in order to minimise the required hardware resources in FPGA and test their robustness. As a result the one-dimensional minimised transform of variable size was chosen. A further part of the paper is focused on particular blocks of implementation. The presented wavelet coefficient computational method decreases the number of necessary operations during transform computing and enables significant reducing of memory requirements. The applied *arcus tangens* block with logarithmic division of approximation intervals gives correct extracted phase values in the entire four-quarter interval. The solution mentioned above leads to great minimisation of the described block without any impact on the accuracy. The last part of the paper presents processing results. The emphasis was put on the noise robustness and influence of the fringe pattern poor quality on the processing error. As it is shown, the algorithm works correctly even with only three gray levels and high noise level present in the input picture.

Keywords: fringe pattern, phase extraction, wavelet transform, FPGA.

1. Wstęp

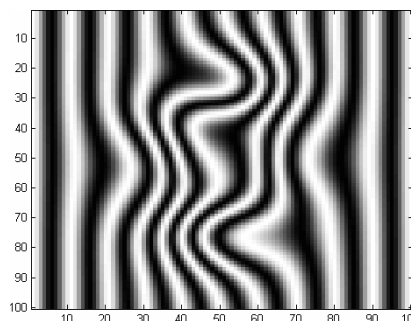
Obrazy prążkowe, zwane również interferogramami, są szeroko stosowane w wielu dziedzinach nauki i techniki. Znaleźć je można w badaniach astronomicznych, medycynie [1] holografii oraz w przemysłowych systemach kontroli jakości. Jednym z zastosowań obrazów prążkowych jest profilometria optyczna. Jest to bezdotykowa metoda pomiarowa pozwalająca na zdalny pomiar obiektu oświetlonego prążkami. Ich zagięcie (zniekształcenia fazowe) na nierównej powierzchni fotografowanego przedmiotu niesie informację o jego kształcie.

Optyczny skaner 3D składa się z projektora wyświetlającego prążki oraz jednej lub dwóch kamer ułożonych nierównolegle w stosunku do źródła prążków. Fotografowany obiekt zostaje oświetlony prążkami, co umożliwia uzyskanie jego cyfrowego modelu. Cały proces prowadzący do uzyskania informacji o kształcie składa się z wielu etapów.

Skanowanie rozpoczyna się od kalibracji stanowiska pomiarowego, gdyż do obliczeń konieczna jest znajomość położenia kamery i projektora względem siebie. W dalszej części następuje właściwa fotografia. Rzeczywisty obraz prążkowy wymaga kilku-etapowej korekcji, gdyż w oryginalnej wersji zazwyczaj nie nadaje się do dalszych etapów przetwarzania. Po przygotowaniu obrazu następuje wyznaczenie rozkładu fazy prążków, a następnie jej usuwanie. Po odtworzeniu fazy prążków, dzięki znajomości geometrii układu pomiarowego, można wyznaczyć współrzędne każdego piksela obrazu, co w konsekwencji prowadzi do powstania chmury punktów w przestrzeni. Ich triangulacja owocuje gotową powierzchnią, a więc pożądanym skanem trójwymiarowym.

Istnieje wiele sposobów wyznaczania fazy z obrazów prążkowych. Do metod najczęściej spotykanych z pewnością zaliczyć można zastosowanie transformacji Fouriera [2]. Jej powszechność wynika niewątpliwie z popularności mechanizmu FFT. Metoda ta wymaga jednak odpowiedniej filtracji obrazu, której automatyzacja jest bardzo kłopotliwa. Ponadto, ze względu na niestacjonarny charakter interferogramu, lepszych rezultatów należy spodziewać się po zastosowaniu metody analizy czasowo-częstotliwościowej.

Do metod dających najlepsze efekty a zarazem możliwych do zautomatyzowania zaliczyć można te, oparte na transformacji falkowej. Posiadają one tę zaletę, że do przetwarzania jednego punktu obrazu wykorzystują jego bliskie sąsiedztwo. Owocuje to mniejszym wpływem ewentualnego szumu na wyniki. Jednocześnie nie wykorzystują całego obrazu (lub też całego wiersza), jak w przypadku transformacji Fouriera, przez co nie występuje niekorzystne uśrednianie widma sygnału. Transformacja falkowa pozwala na analizę obrazu zarówno w domenie częstotliwości, jak i położenia, co ze względu na możliwy bardzo niejednorodny charakter odkształceń jest cechą bardzo pożądaną. Niestety są to również algorytmy wymagające dużej ilości obliczeń.



Rys. 1. Testowy obraz prążkowy
Fig. 1. Test fringe pattern

2. Opis algorytmu wyznaczania rozkładu fazy

Opisywany algorytm opiera się na transformacji falkowej z wykorzystaniem różnego rodzaju zespolonych falek. Ich zestawienie znajduje się w dalszej części rozdziału. Istotny jest fakt, iż zastosowana falka musi być zespolona, gdyż za pomocą falki rzeczywistej nie jest możliwe uzyskanie informacji o fazie sygnału. Niezależnie od typu samej transformacji pozostała część algorytmu jest w każdym przypadku taka sama. Rysunek 1 przedstawia testowy obraz prążkowy.

Pod kątem omawianego zastosowania transformacje falkowe w ogólności podzielić można na jedno- i dwuwymiarowe. W przypadku jednowymiarowym przekształcenie opisuje wzór:

$$W(s, a) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(x) \overline{\psi\left(\frac{x-a}{s}\right)} dx \quad (1)$$

a falka opisana jest równaniem:

$$\psi(x) = \exp\left(j2\pi f_0 \frac{x-a}{s}\right) \exp\left[-m\left(\frac{x-a}{s}\right)^2\right] \quad (2)$$

natomiast w przypadku dwuwymiarowej transformacji przekształcenie przyjmuje postać:

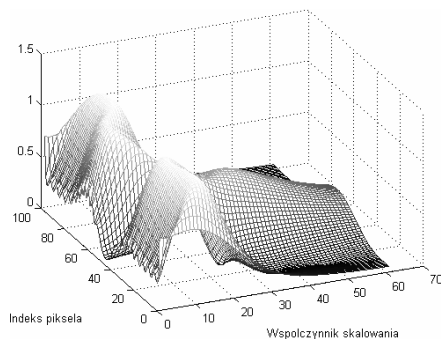
$$W(s, a, b) = \frac{1}{s} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \overline{\psi\left(\frac{x-a}{s}, \frac{y-b}{s}\right)} dx dy \quad (3)$$

a falka opisana jest równaniem:

$$\psi(x, y) = \exp\left(j2\pi f_0 \frac{x-a}{s}\right) \exp\left\{-m\left[\left(\frac{x-a}{s}\right)^2 + \left(\frac{y-b}{s}\right)^2\right]\right\} \quad (4)$$

gdzie: s – współczynnik skalowania falki, m – współczynnik szybkości zaniku obwiedni falki, f_0 – częstotliwość podstawowa falki, a – przesunięcie falki względem osi X, b – przesunięcie falki względem osi Y.

W wyniku przetworzenia pojedynczego wiersza uzyskuje się zespolony wektor dwuwymiarowy (rys. 2). Pierwszym wymiarem jest indeks piksela x , natomiast drugim współczynnik skalowania s . W ten sposób dla każdego piksela obrazu powstaje jednowymiarowy zespolony wektor o długości równej ilości zastosowanych falek (z różnymi współczynnikami skalowania - dekodowanie różnych częstotliwości). Moduł każdej z wartości pozwala na stwierdzenie jaka częstotliwość występuje w otoczeniu danego punktu - im jest on większy, tym częstotliwość falki jest bliższa częstotliwości w danym miejscu na obrazie.



Rys. 2. Moduł transformaty pojedynczego wiersza
Fig. 2. Absolute value of single verse transformation

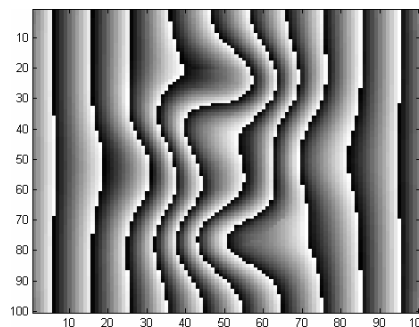
Znajdując zatem współczynnik skalowania s odpowiadający największemu modułowi transformaty dla danego piksela można określić jaka występuje w tym miejscu częstotliwość [2]. Dzięki temu, iż zastosowana falka jest zespolona można również odtworzyć chwilową fazę fali skojarzoną z tym pikselem wg poniższego wzoru:

$$\varphi(x, y) = \arctan\left(\frac{\Im[W(x, y, s_{\max})]}{\Re[W(x, y, s_{\max})]}\right) \quad (5)$$

W ten sposób odtworzona zostaje poszukiwana faza. Ze względu na zastosowanie wielowartościowej funkcji, jaką jest *arcus tangens* odtworzona faza jest charakterystycznie "poszarpana":

$$-\pi < \varphi(x, y) \leq \pi \quad \forall x, y \quad (6)$$

Jest to tzw. *wrapped phase* (rys. 3). W celu uzyskania pożądanego, tj. ciągłego kształtu fazy należy ją poddać tzw. *unwrappingowi* [3].



Rys. 3. Faza wyznaczona z testowego obrazu prążkowego
Fig. 3. Phase extracted from test fringe pattern

Analizie poddanych zostało sześć wariantów algorytmu różniących się rodzajem zastosowanych falek:

- jednowymiarowa transformacja falkowa,
- dwuwymiarowa transformacja falkowa,
- dwuwymiarowa transformacja falkowa z obrotem,
- zminimalizowana dwuwymiarowa transformacja falkowa z obrotem,
- zminimalizowana dwuwymiarowa transformacja falkowa z obrotem i zmiennym rozmiarem,
- zminimalizowana jednowymiarowa transformacja falkowa ze zmiennym rozmiarem.

Transformacja jednowymiarowa jest metodą najprostszą z omówionych. Każda z falek jest w tym przypadku jednowymiarowym wektorem, co oznacza mniejszą ilość mnożeń oraz mniejszą ilość wymaganej pamięci niż w przypadku falek dwuwymiarowych. Owocuje to również dużą szybkością. Wadą jest niższa odporność na szum. Ze względu na zerową wartość średnią szumu pożądana jest duża liczba sumowanych pikseli. Teoretycznie problem ten można rozwiązać poprzez poszerzenie obwiedni falki. Pogarsza to jednak precyzję lokalizacji dekodowanej częstotliwości. Z tego punktu widzenia korzystne jest zwężanie obwiedni. Widoczny jest tu wyraźna rozbieżność wymagań stawianych budowie falki.

Rozwiązaniem tej sytuacji są falki dwuwymiarowe. Przy stosunkowo wąskiej obwiedni sumowana jest znacznie większa ilość pikseli, gdyż brane pod uwagę są również sąsiednie wiersze. Poprawia to znacznie stosunek sygnału do szumu, nie pogarszając zdolności rozdzielczej stosowanych falek. Falki dwuwymiarowe są bardzo dobrym narzędziem do dekodowania obrazu prążkowego przy niewielkim odchyleniu prążków. Sytuacja ta ulega diametralnej zmianie w obszarach, gdzie prążki ulegają silnemu zakrzywieniu. Wynika to ze słabej korelacji "pionowej" falki z niemal "poziomymi" prążkami. Na tym przykładzie widać, iż w obrazie prążkowym ważna jest nie tylko częstotliwość, ale także kierunek ułożenia prążków.

W celu poprawienia rezultatów należy dodać jeszcze jeden parametr generowanej falki, którym jest kąt jej obrotu. Obracając falkę można ją znacznie lepiej dopasować do danego obszaru na obrazie, dzięki czemu lokalna częstotliwość może zostać poprawnie zdekodowana. Zmniejsza to znacznie błędy w okolicach największych odchylen fazy. Z przeprowadzonych testów wynika, iż ilość różnych kątów obrotu falki nie musi być duża. Niemniej jednak pojawienie się dodatkowego parametru falki zwiększa kilkukrotnie postać samej transformaty W .

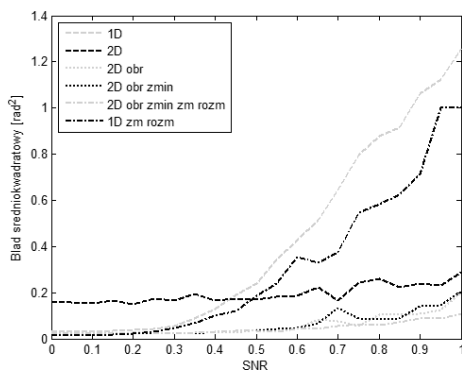
Dwuwymiarowa transformacja falkowa charakteryzuje się bardzo dużą ilością mnożeń. Biorąc pod uwagę wykładniczy zanik obwiedni falki można przyjąć, co zresztą wynika z istoty samej transformacji, iż dalekie otoczenie aktualnie przetwarzanego punktu ma minimalny wpływ na wartość transformaty. Na tej podstawie ograniczyć można rozmiar stosowanej falki i uniezależnić go od wielkości obrazu. W efekcie liczba mnożeń diame-

tralnie spada, przyspieszając znacznie algorytm. Jednocześnie efekt końcowy (wyznaczona faza) nie ulega pogorszeniu.

Stały rozmiar falki jest dopasowany do najszerzej z nich, tj. dekodującej najniższą częstotliwość. W przypadku falki o najwyższej częstotliwości (najwyższej) obwódka bardzo szybko zanika. Wówczas wynik mnożenia piksela obrazu z wartością znajdującą się na obrzeżach falki jest na tyle mały, iż jego dodanie do transformaty ma pomijalnie małe znaczenie. Prowadzi to do sformułowania kryterium, które będzie pomocne przy doborze rozmiarów falki. Jest nim graniczna wartość modułu falki. Pozwala ona na określenie takiego rozmiaru falki N , który ograniczy generowanie punktów o pomijalnie małym module (mniejszym niż wartość abs_{min}):

$$N = 2\sqrt{-\frac{s^2 \ln(abs_{min})}{m}} + 1, \quad (7)$$

Takie rozwiązanie przyspiesza realizację algorytmu, gdyż czas obliczenia transformaty jest kwadratowo zależny od rozmiaru falki. Nie powoduje przy tym pogorszenia efektu oraz odporności na szumy. Dodatkowym atutem jest ok. dwukrotnie mniejsza ilość pamięci potrzebna na zapis wykorzystywanych falek. W świetle tych faktów zminimalizowana dwuwymiarowa transformacja z obrotem i zmiennym rozmiarem jest niewątpliwie najlepszą z dotąd omówionych. Zazwyczaj jednak poziom szumów rzeczywistych obrazów prążkowych nie jest tak duży, by konieczne było stosowanie falek dwuwymiarowych [4]. Dlatego też z powodzeniem zastosowany może zostać algorytm oparty na jednowymiarowej falce ze zmiennym rozmiarem. Jest to najszybsza z opisanych metod ekstrakcji fazy. Charakteryzuje ją również najmniejszy wymagany do zapisu falek rozmiar pamięci. Dlatego też właśnie ten algorytm został wybrany do implementacji omówionej w dalszej części.



Rys. 4. Wpływ szumu na błąd przetwarzania
Fig. 4. Noise influence on processing error

3. Implementacja w układzie FPGA

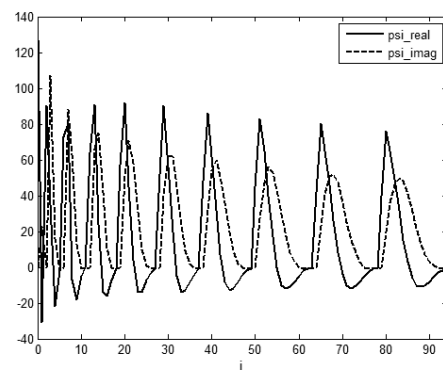
Opisany wyżej algorytm ekstrakcji fazy zaimplementowany został na platformie ewaluacyjnej wyposażonej w układ FPGA XC5VLX50 z rodziny Virtex-5. Omawiany układ posiada 48 układów mnożących w kodzie uzupełnień do dwóch oraz 48 niezależnych bloków pamięci RAM [6], z których każdy charakteryzuje się pojemnością wynoszącą 36 kB. Przetwarzany obraz posiada rozmiary 128x128 pikseli o ośmiobitowej precyzji. Na zapis jednego wiersza potrzeba zatem jedynie 128 B pamięci. Ponadto platforma posiada złącze DDR2 SODIMM pozwalające podłączyć do 256 MB zewnętrznej pamięci.

Wyznaczenie współczynników falkowych wymaga zastosowania kilku skomplikowanych funkcji matematycznych. Ich implementacja w celu wyznaczenia wartości falek przed rozpoczęciem właściwego przetwarzania obrazu prążkowego zwiększyła by czas wykonywania całego algorytmu oraz zwiększyła by wykorzystywane zasoby sprzętowe. Przyjęty zakres dziesięciu współczynników skalowania s pozwala na poprawne odtworzenie fazy z więk-

szości obrazów prążkowych. Dlatego też w implementacji sprzętowej współczynniki falkowe zapisane są jako wartości stałe.

Istotnym zagadnieniem jest precyzja, z jaką należy zapisać omawiane współczynniki. Wejściowy obraz prążkowy charakteryzuje się ośmiobitową stałoprzecinkową precyzją. Z tego względu przyjęta została ośmiobitowa precyzja zapisanych współczynników. Nie powoduje ona znaczącego obniżenia jakości przetwarzania. Błąd średniokwadratowy przetwarzania z użyciem falek zapisanych w formacie *double* (8 bajtów) wynosi 0,012509 [rad²]. W przypadku ośmiobitowego formatu stałoprzecinkowego błąd ten osiąga wartość 0,012535 [rad²]. Różnica jest więc pomijalnie mała, a zastosowanie arytmetyki stałoprzecinkowej znacznie upraszcza strukturę i przyspiesza całe przetwarzanie.

Istotną wadą tej postaci transformacji opisaną wzorami (1) i (3) jest końcowe mnożenie otrzymanej wartości całki przez odwrotność pierwiastka współczynnika skalowania s . Efektywniej jest przeskalować falki na etapie wyznaczania ich wartości (rys. 5).



Rys. 5. Wyznaczone współczynniki falkowe
Fig. 5. Computed wavelet coefficients

Ze względu na parzystość oraz nieparzystość funkcji składowych falek można zmniejszyć niemal dwukrotnie ilość mnożeń akumulowanych poprzez dodanie do siebie (z odpowiednimi znakami) wartości pikseli obrazu równooddalonych od aktualnie przetwarzanego piksela oraz wymnożenie tak utworzonej sumy przez odpowiedni współczynnik falkowy (8). Środkowy współczynnik falkowy został dwukrotnie zmniejszony dzięki czemu niezależnie od wartości oddalenia zawsze sumowane są dwie wartości pikseli (rys. 5). Eliminuje to konieczność stosowania dodatkowych układów logicznych. Tak przygotowane współczynniki falkowe umożliwiają wyznaczenie transformaty dla każdego piksela obrazu wg poniższego wzoru:

$$W(s) = \sum_n^{N-1} \varphi_s(n) \cdot [f(x-n) \pm f(x+n)] \quad (8)$$

gdzie sumowanie odbywa się dla części rzeczywistej, natomiast odejmowanie dla urojonej.

W dalszej części wyznaczany jest współczynnik o największym module. Ze względu na monotoniczność funkcji pierwiastka porównywane są kwadraty modułów liczb zespolonych - implementacja układu pierwiastkującego nie jest konieczna.

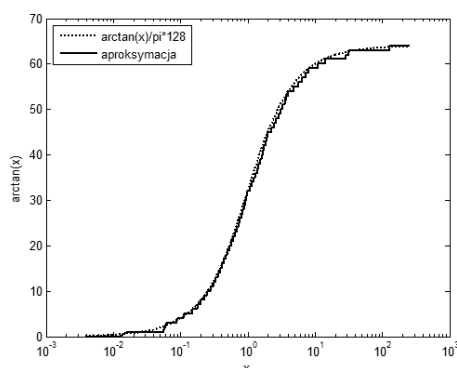
Następnym krokiem algorytmu jest dzielenie. Wybór implementacji układu nie jest tu krytyczny ze względu na fakt, iż zdecydowaną większość czasu zajmuje samo wyznaczanie transformaty. Istotne jest jednak, aby układ dzielący zwracał znaki licznika oraz mianownika w celu późniejszego określenia wartości fazy w czterocwartkowym zakresie.

Końcowym krokiem wyznaczenia fazy jest zastosowanie funkcji *arcus tangens*. Zaimplementowany został blok aproksymacji liniowej tejże funkcji dla jednej ćwiartki układu współrzędnych (ϕ). W oparciu o znajomość znaków licznika i mianownika, a więc i ćwiartkę układu, w której znajduje się dany kąt (n) z układu dzielącego, końcowa wartość fazy (Φ) zostaje wyznaczona na podstawie zależności:

$$\Phi = \begin{cases} \varphi & n = 1 \\ \pi - \varphi & n = 2 \\ -\pi + \varphi & n = 3 \\ -\varphi & n = 4 \end{cases} \quad (9)$$

Ze względu na nieskończoną dziedzinę wejściową aproksymowanej funkcji, jak również przyjętą w implementacji arytmetykę stałoprzecinkową, istotnym zagadnieniem staje się ograniczenie zakresu argumentu wejściowego. Blok dzielący dostarczający argument wejściowy omawianego bloku posiada możliwość sygnalizacji kątów szczególnych, co odpowiada wartości nieskończonej lub zerowej argumentu wejściowego. Dlatego też tenże argument wejściowy można ograniczyć. Przyjęta została 16-bitowa precyzja argumentu wejściowego (po jednym bajcie na część całkowitą oraz ułamkową). Zatem największa możliwa wartość mogąca się pojawić na wejściu to 127,996. Otrzymana wartość fazy w takim przypadku to 1,563 radiana (88,55°) i jest ona maksymalna o ile nie pojawi się flaga kąta specjalnego (np. 90°). Powoduje to błąd nie większy niż 0,5%. Nie ma on zatem praktycznego znaczenia dla całości przetwarzania. Wartości wyjściowe (wyznaczona faza) posiadają identyczną szerokość bitową jak wejściowy obraz prążkowy, tj. 8 bitów. Zapewnia to rozdzielczość na poziomie 0,0245 radiana (1,4°). Funkcja zwraca wartości kątów w kodzie uzupełnień do dwóch podzielone przez π .

Funkcja *arcus tangens* posiada znacznie bardziej jednolity przebieg, gdy pozioma oś wyskalowana jest logarytmicznie. Z tego względu zastosowanie wykładniczego podziału przedziałów aproksymacji pozwala na angażowanie znacznie mniejszych zasobów niż w przypadku liniowego podziału. Zaimplementowaną aproksymację przedstawia rys. 6.



Rys. 6. Aproksymacja funkcji *arcus tangens* z logarytmicznym podziałem
Fig. 6. *Arcus tangens* approximation with logarithmic division

Wartość błędu maksymalnego przy zaledwie 16 przedziałach aproksymacji wynosi 0,0058665 [rad] (1,2% zakresu), natomiast błąd średni wynosi 0,0013004 [rad] (0,26% zakresu). Są to więc wartości zbliżone do błędu wynikającego z 8-bitowego podziału wyznaczanej fazy (0,4%)

4. Wyniki przetwarzania

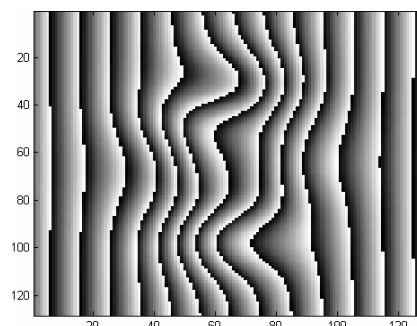
Zaimplementowany algorytm poddany został dwóm rodzajom testów sprawdzających poprawność jego działania (tab. 1,2). Pierwszy z nich przedstawia wpływ amplitudy prążków na jakość przetwarzania. Drugi natomiast weryfikuje wpływ szumów. Zastosowany obraz testowy został zwiększony do rozmiaru 128x128 pikseli.

Tab. 1. Wpływ amplitudy prążków na jakość przetwarzania
Tab. 1. Influence of the fringe amplitude on the processing quality

Amplituda prążków	127	32.	8	4	2	1
RMSE [rad ²]	0,0069	0,0068	0,0073	0,0079	0,0136	0,0159
Błąd maks. [rad]	0,4051	0,4051	0,4051	0,4110	0,4299	0,5388

Tab. 2. Wpływ szumu na jakość przetwarzania
Tab. 2. Influence of noise on the processing quality

Poziom szumu	5%	10%	15%	20%	25%	30%
RMSE [rad ²]	0,0074	0,0093	0,0122	0,0172	0,0230	0,0358
Błąd maks. [rad]	0,3959	0,4387	0,4676	0,8370	1,0088	3,0926



Rys. 7. Faza odtworzona przez zaimplementowany algorytm
Fig. 7. Phase extracted by the implemented algorithm

5. Podsumowanie

Zaimplementowany algorytm wykazuje dobrą odporność na szum. Ponadto jego poprawnej pracy nie zakłóca uboga paleta odcieni: poprawne wyniki można uzyskać nawet przy trzech poziomach szarości obrazu wejściowego (amplituda = 1).

Czas przetwarzania ulega nieznacznym wahaniom w zależności od przetwarzanych wartości (układ dzielący oraz blok aproksymacji) i dla obrazu o wymiarach 128x128 pikseli wynosi od 68 do 77 ms przy taktowaniu częstotliwością 100 MHz. Należy zaznaczyć, iż czas przetwarzania jest w głównej mierze zależny od bloku wyznaczającego transformację, który można zwielokrotnić w zależności od ilości dostępnych zasobów. W opisanej implementacji użyto jedynie siedmiu z 48 dostępnych układów mnożących. W ten sposób czas przetwarzania może ulec kilkukrotnemu skróceniu. Głównie ten fakt, jak również możliwość tworzenia dedykowanych bloków funkcyjnych, stanowi w tym wypadku przewagę układów FPGA nad klasycznymi procesorami. Stąd też wybór tego typu architektury.

Prace badawcze, których wyniki przedstawiono w niniejszej publikacji, zostały współfinansowane ze środków Europejskiego Funduszu Rozwoju Regionalnego.

6. Literatura

- [1] Lilley F.: An optical 3D body surface measurement system to improve radiotherapy treatment of cancer. Ph.D. thesis, John Moores University, Liverpool 1999.
- [2] Abid A. Z. A.: Fringe Pattern Analysis using Wavelet Transforms. John Moores University, Liverpool 2008.
- [3] Zhang S.: High-resolution, Real-time 3-D Shape Measurement. Ph.D. thesis, Stony Brook University, 2005.
- [4] Fringe Pattern Analysis Using Wavelet Transforms and Phase Demodulation Using Wavelet Transform. <http://www.ljmu.ac.uk/GERI/79684.htm>.
- [5] Virtex-5 LX FPGA ML501 Evaluation Platform. <http://www.xilinx.com/products/devkits/HW-V5-ML501-UNI-G.htm>.
- [6] Virtex-5 Family Overview. http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf.

otrzymano / received: 15.11.2011

przyjęto do druku / accepted: 02.02.2012

artykuł recenzowany / revised paper