Tomasz KRYJAK, Mateusz KOMORKIEWICZ
AGH AKADEMIA GÓRNICZO-HUTNICZA im. STANISŁAWA STASZICA W KRAKOWIE,
WYDZIAŁ EAIiIB, KATEDRA AUTOMATYKI i INŻYNIERII BIOMEDYCZNEJ, Al. Mickiewicza 30, 30-059 Kraków

Implementacja obliczania map dysparycji w czasie rzeczywistym dla strumienia wizyjnego 3D zrealizowana w układzie FPGA

Dr inż. Tomasz KRYJAK

Autor jest absolwentem kierunku Automatyka i Robotyka na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie (2007). Pracuje w Laboratorium Biocybernetyki Katedry Automatyki i Inżynierii Biomedycznej AGH na stanowisku adiunkta. Interesuje się przetwarzaniem i analizą obrazów, ze szczególnym uwzględnieniem zaawansowanych systemów monitoringu wizyjnego oraz sprzętową akceleracją algorytmów wizyjnych z wykorzystaniem układów FPGA. Autor ponad 20 publikacji.

e-mail: kryjak@agh.edu.pl

Mgr inż. Mateusz KOMORKIEWICZ

Autor jest absolwentem kierunku Automatyka i Robotyka na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie (2010). Obecnie jest słuchaczem studiów doktoranckich na wydziale EAIiIB tej samej uczelni. Interesuje się przetwarzaniem i analizą obrazów oraz sprzętową akceleracją algorytmów wizyjnych z wykorzystaniem układów FPGA.

e-mail: komorkie@agh.edu.pl

Streszczenie

W artykule opisano system umożliwiający odbieranie i przetwarzanie strumienia wideo w technologii 3D transmitowanego w standardzie HDMI (tryb *side by side*), co pozwala na współpracę z dostępnymi na rynku kamerami 3D. Zaproponowana architektura umożliwia implementację popularnych metod obliczania map dysparycji: m. in. SAD oraz opartych o transformatę Censusa, realizację sprawdzenia symetryczności mapy oraz filtrację medianową poprawiającą jakość wyników. W pracy omówiono budowę każdego z modułów, użycie zasobów FPGA, zużycie mocy, a także przykładowe rezultaty działania na płycie ewaluacyjnej VC707 z układem Virtex 7.

Słowa kluczowe: dysparycja, SAD, ZSAD, układy FPGA, 3D, systemy stereowizyjne, przetwarzanie obrazów.

Real-time FPGA implementation of disparity map calculation for a 3D video stream

Abstract

In the paper a system for acquisition and processing of a 3D video stream is presented. It can work with 3D HDMI cameras available on the market. In Section 2 the basic concepts of stereovision systems are described [1]. In Section 3 three distance metrics, SAD [4], ZSAD and Census [5], used for correspondence matching are discussed. Evaluation of the matching process on the Middlebury dataset [2] is also presented. The best results were obtained for the SAD and ZSAD methods and greyscale images. In Table 1 there are shown three best configurations. Figure 1 illustrates the obtained disparity maps. A description of the hardware implementation is given in Section 4. The block diagram of the system is presented in Figure 2. The proposed solution is able to process images transmitted in side by side mode, to compute two disparity maps (left to right and right to left, method from [4]), to use SAD or ZSAD cost function, to check maps consistency and execute median filtering for final image processing. The described module is highly parameterizable: different cost functions, window sizes and disparity range can be used, image size and median filtering size can be adjusted. FPGA resource utilization is presented in Table 2. A picture of the working system is shown in Figure 3 (1280 x 720 @60 fps, real-time video-stream processing). The proposed module can be used for video surveillance, pedestrian collision avoidance systems or in autonomous vehicles.

Keywords: FPGA devices, 3D, disparity, stereovision, real-time image processing, SAD, ZSAD.

1. Wprowadzenie

Technologia obrazów 3D staje się coraz bardziej popularna. Jest ona często wykorzystywana przy tworzeniu nowych filmów, podczas projekcji kinowych, a nawet w transmisjach telewizyjnych. Również w zastosowaniach przemysłowych użycie stereowizji przynosi niezaprzeczalne korzyści. Dzieje się tak, ponieważ obraz 3D niesie o wiele więcej informacji niż obraz dwuwymiarowy. Szczególnie duże znaczenie ma to w takich zagadnieniach jak: segmentacja i rozpoznawanie osób w systemach monitoringu wizyjnego i systemach wspomagających kierowcę, wykrywanie i rozpoznawanie obiektów na scenie, pozycjonowanie i nawigacja robotów oraz budowa autonomicznych pojazdów. We wszystkich wymienionych przypadkach informacja o głębi rozważanej sceny pozwala poprawić skuteczność i dokładność algorytmów, a w niektórych jest wręcz niezbędna do prawidłowego działania.

Przetwarzanie strumienia wideo w czasie rzeczywistym z pary kamer jest zadaniem złożonym obliczeniowo. Dwukrotnie wzrasta ilość danych, które muszą być przesłane oraz liczba operacji, które muszą zostać na nich wykonane, przykładowo w ramach przetwarzania wstępnego lub niwelacji zniekształceń obiektów i rektyfikacji. Osobnym, bardzo złożonym obliczeniowo zagadnieniem, jest wyznaczenie map głębi (tzw. map dysparycji).

Układy reprogramowalne FPGA, które są sprawdzoną platformą do implementacji operacji przetwarzania i analizy obrazów, bardzo dobrze nadają się do realizacji powyżej opisanych algorytmów, gdyż oferują duże możliwości zrównoleglenia obliczeń występujących przy obliczaniu map dysparycji.

2. Stereowizja

Zadaniem systemu stereowizyjnego jest odtworzenie głębi sceny (geometrii 3D sceny) na podstawie dwóch obrazów: lewego (L) i prawego (R) zarejestrowanego przez dwie kamery lub dwa czujniki wizyjne wchodzące w skład kamery 3D (np. używanej w eksperymentach kamery Sony HDR 20 VE).

Pierwszym etapem przetwarzania jest zwykle korekcja zniekształceń wprowadzanych przez obiektyw kamery [1]. Mogą one wynikać z niedokładnie zestawionego układu kamer stereowizyjnych lub niedoskonałości obiektów (wprowadzających zniekształcenia tangensoidalne i radialne). Celem jest doprowadzenie do sytuacji, w której odpowiadające sobie piksele znajdują się na wspólnej, poziomej linii (tzw. epipolarnej). Proces ten określany jest mianem rektyfikacji obrazu.

W drugim etapie oblicza się stopień (koszt) dopasowania pomiędzy pikselami lub obszarami na obrazie lewym i prawym. Dzięki przeprowadzaniu rektyfikacji, przeszukiwanie ograniczone jest do jednej poziomej linii. Następnie koszty podlegają agregacji i wyznaczana jest mapa dysparycji, czyli odległości w pikselach pomiędzy takimi samymi obszarami (punktami) na dwóch obrazach. W literaturze opisano szereg metod obliczania dysparycji, które dzielą się na lokalne (agregacja kosztu na poziomie lokalnego kontekstu) i globalne (agregacja kosztu i optymalizacja dla całego obrazu). Przegląd algorytmów można odnaleźć w pracach [2] i [3].

W ostatnim etapie uzyskana mapa dysparycji jest ulepszana. Stosuje się estymację pod-pikselową, a także filtrację medianową. Ponadto często sprawdza się spójność mapy oryginalnej (korespondencja obrazu lewego do prawego: L->R) i obliczonej dla zamienionych obrazów (korespondencja obrazu prawego do lewego: R->L). Uzyskuje się w ten sposób informacje o obszarach, które są niewidoczne dla jednej z kamer.

3. Metody obliczania dysparycji

W ramach badań wstępnych przeanalizowano szereg, opisanych w literaturze, metod obliczania dysparycji. Skoncentrowano się na podejściach lokalnych z uwagi na możliwość ich bezpośredniej realizacji w sposób równoległy i potokowy w układzie reprogramowalnym FPGA.

Jedną z najbardziej popularnych funkcji kosztu dopasowania jest SAD (ang. *Sum of Absolute Differences*) [4]:

$$SAD = \sum_{c \in C} \sum_{i=-n}^{n} \sum_{j=-n}^{n} \left| I_L^c(x+i, y+j) - I_R^c(x+i+d, y+j) \right| \quad (1)$$

gdzie: I_L^c i I_R^c to odpowiednio lewy i prawy obraz. Parametr c oznacza wybraną składową barwną (RGB lub poziomy szarości), n - rozmiar analizowanego otoczenia, d - dysparycja (w przedziale od 0 do maksymalnej odległości przeszukiwania).

Odmianą jest algorytm ZSAD (ang. *Zero Mean Sum of Absolute Differences*), w którym przed obliczaniem SAD, od każdego piksela z bieżącego kontekstu odejmowana jest wartość średnia dla tego kontekstu.

Drugim, często wykorzystywanym podejściem do obliczania kosztu jest transformata Censusa [5]. W najprostszej wersji polega ona na dokonaniu binaryzacji lokalnego kontekstu z progiem w postaci wartość elementu centralnego. Odległość pomiędzy dwoma kontekstami określona jest jako suma występujących różnic (operacja XOR na dwóch ciągach binarnych - odległość Hamminga). W bardziej złożonych wariantach próg binaryzacji wyznaczany może być jako: średnia lub mediana z otoczenia. Transformata Censusa (rezultat binarny) może zostać rozszerzona do LTP (ang. *Local Ternary Pattern*). W tym przypadku, oprócz progu, występuje parametr przesunięcia (ang. *bias*) i wynik może przyjmować 3 wartości (-1 dla pikseli poniżej wartości *próg* - *bias*, 0 dla pikseli w przedziale [*prog-bias;prog+bias*], 1 dla pikseli powyżej wartości *próg* + *bias*. Zaletą tego podejścia jest mniejsza wrażliwość na błędy w obszarach o jednorodnej jasności.

Jako przetwarzanie końcowe wykorzystano filtrację medianową z oknami o różnych rozmiarach (3x3, 5x5, 7x7, 9x9, 11x11).

Do testów użyto obrazów *tsukuba*, *venus*, *teddy* i *cones* ze zbioru Middlebury dostępnego pod adresem `http://vision.middlebury.edu/stereo/` [2]. Na podstawie porównania wyznaczonych map ze wzorcem (ang. *ground truth*) obliczano trzy współczynniki:

- *all* - procent różnic dla całego obrazu (bez brzegów),
- *nonocc* - procent różnic dla obszarów nieprzesłoniętych,
- *disp* - procent różnic dla obszarów, w których następuje zmiana dysparycji (obszary w pobliżu krawędzi obiektów).

Tab. 1. Zestawienie trzech najlepszych wyników uzyskanych dla każdego z analizowanych obrazów testowych
Tab. 1. List of the top three results obtained for each analyzed test image

Obraz	Metoda	Rozmiar otocznia	Rozmiar mediany	Współczynnik		
				nonocc	*all*	*disc*
tsukuba	SAD	5x5	11x11	5,73	7,82	19,91
	ZSAD	5x5	11x11	5,90	7,89	21,82
	SAD	7x7	11x11	6,00	8,05	23,05
venus	ZSAD	7x7	11x11	2,92	4,42	30,08
	ZSAD	7x7	9x9	3,23	4,72	30,19
	ZSAD	5x5	11x11	3,32	4,80	24,79
teddy	ZSAD	5x5	11x11	6,43	15,24	13,90
	ZSAD	5x5	9x9	6,55	15,39	13,86
	ZSAD	5x5	7x7	7,09	15,92	14,29
cones	ZSAD	3x3	11x11	4,19	13,00	9,37
	ZSAD	3x3	9x9	4,11	13,11	9,19
	ZSAD	3x3	7x7	4,32	13,51	9,59

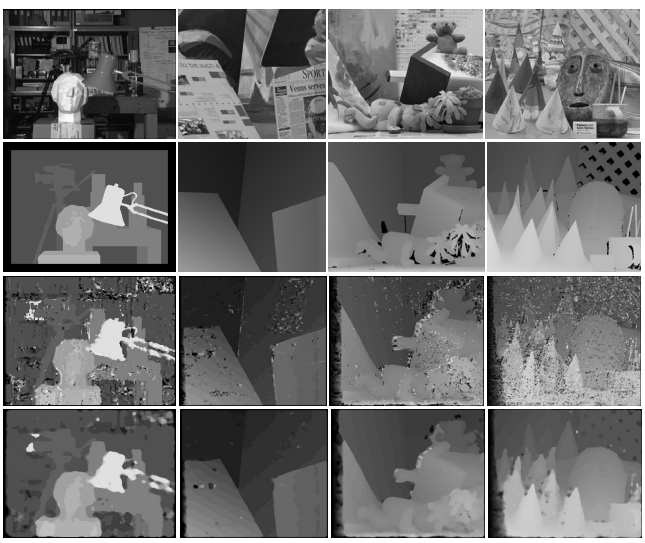
W badaniach wstępnych użyto aplikacji stworzonej w pakiecie MATLAB, która automatyzowała testowanie różnych wariantów obliczania dysparycji. W każdym z przypadków analizowano trzy przestrzenie barw: odcienie szarości, RGB i CIE Lab, różne rozmiary otoczenia (3x3, 5x5, 7x7, 9x9, 11x11) i różne warianty funkcji obliczania kosztu (SAD, ZSAD, Census, LTP). W sumie wyznaczono ponad 2500 map dysparycji dla każdej pary obrazów.

Pierwsze trzy najlepsze wyniki, dla każdego z obrazów testowych zaprezentowano w tabeli 1. Ograniczono się do danych, dla odcieni szarości. Wyznaczone mapy dysparycji zaprezentowano na rysunku 1.

Analiza uzyskanych wyników pozwala sformułować następujące stwierdzenia:

- najlepsze wyniki otrzymuje się dla metod SAD i ZSAD,
- niemal wszystkie metody dają najlepsze wyniki dla przetwarzania w odcieniach szarości,
- dla różnych obrazów testowych, najlepszy wyniki uzyskuje się dla różnych rozmiarów okna,
- kluczowe dla wyznaczenia dobrej mapy dysparycji jest zastosowanie filtracji medianowej z dużym rozmiarem okna (11 x 11).

Na postawie uzyskanych wyników zdecydowano się zaimplementować metody SAD i ZSAD dla odcieni szarości oraz filtrację medianową z możliwością zmiany rozmiaru okna. Wykonano także mechanizm umożliwiający sprawdzenie spójności mapy dysparycji L->R i R-> L.



a) b) c) d)

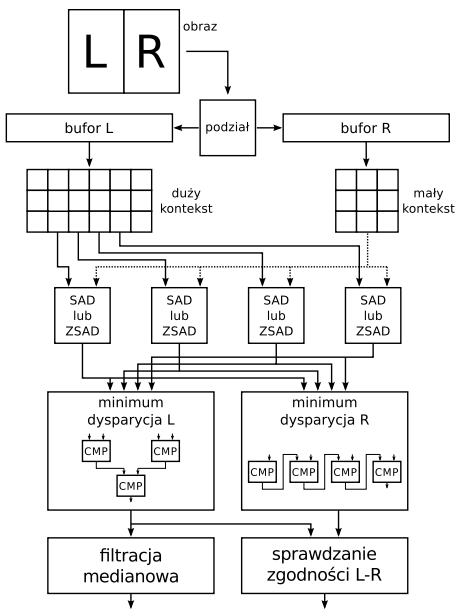
Rys. 1. Obrazy testowe i wyznaczone mapy dysparycji. W wierszach kolejno: obraz oryginalny, mapa referencyjna, mapa uzyskana najlepszą metodą dla danego obrazu, mapa poddana filtracji medianowej z oknem o rozmiarze 11x11. Obrazy a) *tsukuba*, b) *venus*, c) *teddy*, d) *cones*
Fig. 1. Test images and calculated disparity maps. In the rows: the original image, ground truth, map obtained with the best method for a given test image, map with median filtering 11 x 11. Images: a) *tsukuba*, b) *venus*, c) *teddy*, d) *cones*

4. Implementacja sprzętowa

Schemat zaimplementowanego systemu został przedstawiony na rysunku 2. Ponieważ obraz 3D z kamery dostarczany jest w trybie *side-by-side* (obraz lewy i prawy złożone w jedną ramkę), konieczne okazało się rozdzielenie i buforowanie lewej (L) i prawej (R) linii obrazu, tak aby możliwe było jednoczesne dostarczenie odpowiednich pikseli z prawego i lewego obrazu do modułów obliczających dysparycję. W dalszej kolejności konstruowane są dwa konteksty, w celu umożliwienia porównania obszaru (zbioru kontekstów o rozmiarze odpowiadającym maksymalnej wartości dysparycji) z lewego obrazu z aktualnym oknem na prawym obrazie. Dane te są odpowiednio dzielone pomiędzy bloki realizujące wyznaczanie stopnia dopasowania.

Wykonano dwie wersje obliczania kosztu: SAD i ZSAD. Aby wyznaczyć dysparycję obrazu prawego w odniesieniu do lewego (R->L) użyto kaskadowo połączonych komparatorów. Do obliczania dysparycji dla lewego obrazu w odniesieniu do prawego (L->R) wykorzystana została metoda opisana w pracy [4]. Pozwoliło to na uniknięcie konieczności powielania całego modułu generującego konteksty i obliczającego koszty. W tym celu użyto szeregowo połączonego łańcucha komparatorów.

Wyznaczone mapy dysparycji poddawane były filtracji medianowej oraz sprawdzana była ich spójność. Filtrację medianową zrealizowano z wykorzystaniem sieci sortującej i algorytmu parzysto - nieparzystego sortowania przez scalanie Batchara (ang. *Batcher odd-even mergesort*) [6].



Rys. 2. Schemat blokowy systemu
Fig. 2. Block scheme of the proposed system

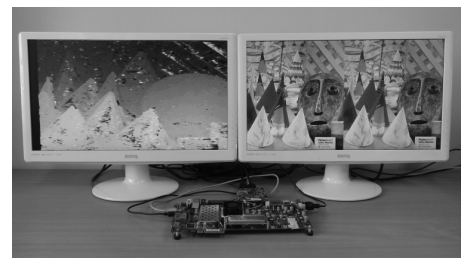
W trakcie prac duży nacisk położono na pełną parametryzowalność modułu. W tym celu wykorzystano dostępne w językach VHDL i Verilog instrukcje *generic* oraz stworzono skrypty w pakiecie MATLAB, które automatycznie generują kod VHDL. Ostatecznie uzyskano możliwość tworzenia modułu o następujących, definiowalnych, parametrach:
- rozdzielczość przetwarzanego obrazu,
- rozmiar okna (kontekstu),
- maksymalna dysparycja,
- metoda liczenia kosztu: (SAD lub ZSAD),
- rozmiar okna dla filtracji medianowej.

Działanie systemu zweryfikowano praktycznie na platformie VC707 z układem FPGA serii Virtex 7 XC7VX485T firmy Xilinx. Użycie zasobów dla dwóch wersji modułu (z liczeniem odległości SAD i ZSAD, przetwarzaniem w odcieniach szarości, oknem 3x3, maksymalną dysparycją 32 i filtracją medianową 7x7) przedstawiono w tabeli. 2.

Tab. 2. Zużycie zasobów FPGA
Tab. 2. FPGA resources utilisation

Zasób	SAD 3x3	ZSAD 3x3	Dostępne
FF	13226 (2%)	44202 (7%)	607200
LUT6	12061 (4%)	35924 (12%)	303600
SLICE	4042 (5%)	14036 (18%)	75900
BRAM_18	1 (0,05%)	1 (0,05%)	2060
BRAM_36	15 (1%)	15 (1%)	1030

Zużycie mocy obliczone przy wykorzystaniu narzędzia XPower Analyzer wynosi 0,815 W (SAD) 1,658 W (ZSAD). Natomiast zmierzone doświadczalnie wynosi 0,91 W (SAD), 1,19 W (ZSAD). Natomiast cała karta ewaluacyjna Xilinx VC707 zużywa 14,1 W (SAD), 14,4 W (ZSAD). Maksymalna częstotliwość pracy raportowana przez narzędzie ISE wynosi 210 MHz (SAD), 200 MHz (ZSAD). Można zauważyć, że chociaż wykorzystanie metody obliczania odległości opartej o funkcję ZSAD prowadzi do uzyskania lepszych wyników, to jest to okupione dużo większym zużyciem zasobów układu reprogramowalnego oraz większym poborem mocy. Działający system przedstawiono na rysunku 3.



Rys. 3. Działający system. Lewy ekran: mapa dysparycji, prawy ekran obraz testowy w trybie *side by side*. Rozdzielczość 1280 x 720 @ 60 fps
Fig. 3. The working system. Left screen: disparity map, right screen test image in *side by side* mode. Resolution 1280 x 720 @ 60 fps

5. Podsumowanie

W artykule przedstawiono rekonfigurowalny system do odbioru i przetwarzania sygnału wizyjnego 3D w czasie rzeczywistym. Umożliwia on współpracę z źródłem sygnału 3D pracującym w standardzie *side by side* (np. kamera Sony HDR 20 VE), wyznaczanie map dysparycji metodami SAD i ZSAD, sprawdzanie spójności mapy oraz filtrację medianową z wykorzystaniem sieci sortującej. Cały moduł opisano w sposób parametryzowalny w językach VHDL i Verilog z częściową generacją kodu za pomocą skryptów w pakiecie MATLAB. Działanie systemu zostało pozytywnie zweryfikowane na platformie sprzętowej VC707 z układem Virtex 7. Uzyskano przetwarzanie w czasie rzeczywistym strumienia wideo o rozdzielczości 1280 x 720 i 60 ramkach na sekundę. Zaproponowany moduł może zostać wykorzystany w monitoringu wizyjnym, systemach poprawy bezpieczeństwa pieszych na drodze lub autonomicznych pojazdach.

Przedstawione w artykule prace były finansowane przez Narodowe Centrum Nauki jako projekt badawczy nr 2011/01/N/ST7/06687.

6. Literatura

[1] Cyganek B., Siebert J.P.: An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley, 2009.
[2] Scharstein D., Szeliski R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.
[3] Tombari, F., Gori, F., Di Stefano, L.: Evaluation of stereo algorithms for 3D object recognition. Computer Vision Workshops (ICCV Workshops), pp.990,997, 6-13 Nov. 2011.
[4] Miyajima, Y., Maruyama, T. A Real-Time Stereo Vision System with FPGA, Lecture Notes in Computer Science, Field Programmable Logic and Application, vol. 2778, pp. 448-457, 2003.
[5] Longfield, S., Chang, M.L. A Parameterized Stereo Vision Core for FPGAs. 17th IEEE Symposium on Field Programmable Custom Computing Machines, pp.263-266, 2009.
[6] Knuth D.E. The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition. Addison-Wesley, 1998. ISBN 0-201-89685-0. Section 5.3.4: Networks for Sorting, pp. 219–247.

otrzymano / received: 20.05.2013
przyjęto do druku / accepted: 03.07.2013 *artykuł recenzowany / revised paper*