

Zbigniew Bubleński\*, Mirosław Jabłoński\*, Zbigniew Mikrut\*

## **Analiza sekwencji filmowych w środowisku VirtualDub w oparciu o platformę FPGA\*\***

### **1. Wprowadzenie**

Etykietowanie (indeksacja) obiektów jest jednym z bardziej istotnych etapów analizy obrazów. Pozwala bowiem na uzyskanie precyzyjnej informacji o liczbie obiektów widocznych na obrazie, jak też – poprzez nadanie wszystkim obiektom unikalnych etykiet – na proste i szybkie wydzielenie dowolnego obiektu do szczegółowych badań. Jednym z obszarów zastosowań, w których etykietowanie obiektów odgrywa istotną rolę, są systemy automatycznej analizy obrazów ruchu drogowego [1–3]. Krytycznym parametrem jest w nich czas reakcji systemu rozumiany jako łączny czas analizy zarejestrowanego obrazu – optymalnym rozwiązaniem byłaby analiza obrazu dokonywana w czasie rzeczywistym. Stawia to wysokie wymagania wobec używanych metod i algorytmów. Szczególnie czas indeksacji obiektów powinien być możliwie krótki, gdyż operacja ta jest jedną z najczęściej używanych. W przypadku gdy realizacja programowa algorytmu pochłania zbyt dużo czasu i jej przyspieszenie jest zbyt trudne i/lub kosztowne, jedynym wyjściem jest wykorzystanie realizacji częściowo lub w pełni sprzętowej.

### **2. Etykietowanie obiektów**

Najczęściej wykorzystywane w praktyce a szczegółowo opisane w [7–9] metody etykietowania obiektów bazują na dwóch podejściach. Pierwsze to indeksacja oparta na rekurencyjnym przeglądaniu punktów obiektu popularnie nazywana metodą „pożaru prerii” (*floodfill*). Ze względu na duże wymagania pamięciowe nie nadaje się do przetwarzania obrazów o znacznych rozmiarach, stwarza też istotne problemy przy próbach stworzenia jej sprzętowej implementacji. Dużo lepsza pod tym względem jest druga z metod – tzw. liniowe przeglądanie obrazu połączone ze sklejaniem uprzednio wstępnie zaetykietowanych obiektów. W metodzie tej wykorzystywane są proste operacje sprowadzające się do przeglądnięcia linii po linii, kolumna po kolumnie wszystkich punktów obrazu i sprawdzenie

---

\* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

\*\* Praca zrealizowana w ramach grantu KBN nr 4T11C01725

dla każdego punktu kilku prostych warunków logicznych, których rezultat warunkuje nadanie danemu punktowi określonej etykiety. Ze względu na to, że w pewnych sytuacjach może okazać się, iż obiekty dotychczas uważane za rozłączne są faktycznie ze sobą połączone, konieczny jest dodatkowy etap przetwarzania, w którym informacje o ewentualnych połączeniach obiektów o różnych etykietach zapamiętane w tzw. tablicy sklejeń wykorzystuje się do dokonania częściowej reindeksacji obrazu tak, aby cały obiekt uzyskał tą samą etykietę. Ze względu na swoją prostotę i nieskomplikowane zależności czasowe w trakcie przetwarzania, metoda ta szczególnie dobrze nadaje się do realizacji sprzętowej – szczególnie opis takiej implementacji zawarto w [5, 6].

Wykorzystanie sprzętowego wspomaganie wiąże się z koniecznością dokonania dodatkowych operacji. Konieczne jest bowiem przesłanie analizowanego obrazu z pamięci komputera do pamięci układu FPGA, uruchomienie przetwarzania w tym układzie, a po jego zakończeniu przesłanie obrazu z powrotem do pamięci komputera. Te dodatkowe czynności oznaczają zmniejszenie efektywnej wydajności wspomaganie sprzętowego – chyba że cały proces analizy będzie realizowany sprzętowo – na razie jest to jednak trudne do osiągnięcia. Aby móc skutecznie ocenić implementację sprzętową, zdecydowano się na równoległe opracowanie realizacji całkowicie programowej. Umożliwiło to porównanie efektywności obu implementacji – tym bardziej, że obie zostały skonstruowane jako filtry działające w środowisku VirtualDub [4]. Takie podejście pozwoliło na zapewnienie identycznych warunków w trakcie badań porównawczych.

### 3. Platforma realizacji

Sprzętowa implementacja algorytmu indeksacji obrazów binarnych została zrealizowana na karcie RC1000-PP, wyposażonej w układ reprogramowalny VIRTEX™ XCV2000E firmy Xilinx [11]. Karta została dostarczona przez firmę Celoxica [10]. Wraz z dołączonymi bibliotekami, zestawem narzędzi programistycznych i uruchomieniowych, stanowi pakiet do sprzętowej realizacji algorytmów w środowisku Windows, działającym na platformie PC. Urządzenie wykonane jest w postaci karty rozszerzającej komputera PC z interfejsem PCI. Elementem centralnym jest układ FPGA, który współdzieli cztery niezależne bloki pamięci statycznej RAM z komputerem nadrzędnym poprzez magistralę PCI. Każdy blok pamięci zorganizowany jest w 512K słów 32-bitowych. Architektura karty przedstawiona na rysunku 1 charakteryzuje się wysoką elastycznością. System arbitrażu umożliwia dynamiczny przydział obszarów pamięci karty zarówno procesorowi PC, jak i układowi reprogramowalnemu FPGA.

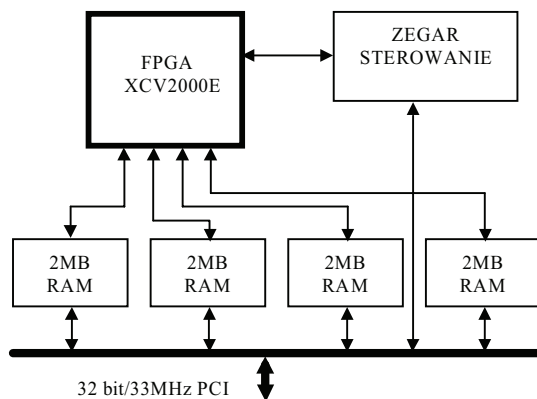
Z poziomu aplikacji komputera PC można również ustawiać częstotliwość pracy karty oraz zaprogramować układ FPGA odpowiednim plikiem konfiguracyjnym BIT. Zarówno zasoby obliczeniowe, jak i dodatkowe elementy pamięciowe skoncentrowane są w układzie FPGA. Wszystkie czynności związane z pracą i konfiguracją karty mogą być wykonane z wykorzystaniem API<sup>1)</sup> oraz pakietu SDK<sup>2)</sup> dołączonego do karty. Prosta architektura

---

<sup>1)</sup> *Application Programming Interface*

<sup>2)</sup> *Software Development Kit*

karty, wysoki stopień konfigurowalności i współbieżności oraz przejrzysty interfejs komunikacyjny sprawiają, iż jest ona szczególnie przydatna na etapie uruchamiania i testowania aplikacji masowego przetwarzania danych, a w szczególności systemu przetwarzania i analizy obrazów.



Rys. 1. Schemat blokowy karty RC1000-PP

Algorytm indeksacji w całości zaimplementowano w języku Handel-C, do utworzenia pliku konfiguracyjnego wykorzystano narzędzia i pakiet technologiczny ISE firmy Xilinx. W tabeli 1 zestawiono charakterystykę projektu algorytmu indeksacji obrazów binarnych, zaimplementowanego dla układu XCV2000E.

Tabela 1  
Wyniki implementacji

Zużycie zasobów obliczeniowych układu FPGA	4%
Zużycie wewnętrznych zasobów pamięciowych	16%
Zużycie zewnętrznych zasobów pamięciowych	100%
Maksymalna częstotliwość pracy	60 MHz

#### 4. Wyniki testów

Pokazane dalej rezultaty uzyskano przy założeniu że w przypadku indeksacji sprzętowej mierzone będą sumaryczne czasy przetwarzania i transferów do i z pamięci komputera. Aby jednak stwierdzić, jakie są relacje pomiędzy czasami transferów a czasem samej indeksacji sprzętowej, obok łącznego czasu dokonywano też pomiaru poszczególnych etapów (transfer PC → FPGA, indeksacja, transfer FPGA → PC).

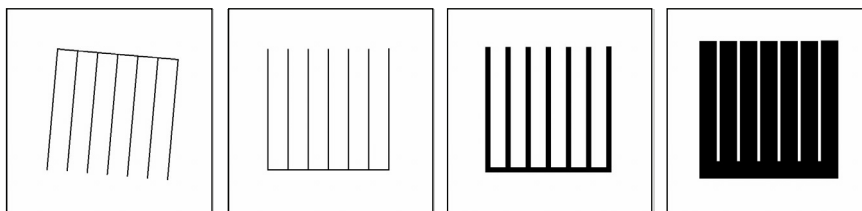
Do pomiaru czasu wykonania wykorzystano tzw. High Performance Timer dostępny w środowisku Windows. Timer ten poprzez wykorzystanie zmiennych 64-bitowych zapewnia rozdzielczość na poziomie pojedynczych mikrosekund i trochę mniejszą dokładność.

Nie stanowi to jednak problemu, gdyż czasy indeksacji w zależności od metody i zawartości samego obrazu zawierają się w przedziale od kilku do kilkudziesięciu milisekund. Aby zwiększyć dokładność pomiaru czasu, uzyskane wyniki skorygowano, uwzględniając czas trwania samego pomiaru czasu. Należy jednak pamiętać, że w większości środowisk wielozadaniowych (a do takich należy MS Windows) pomiar czasu dowolnej operacji nie może być zawsze precyzyjny ze względu na równoległe działanie innych procesów. Zawsze jednak można te błędy minimalizować poprzez powtórzenie operacji i uśrednienie wyników.

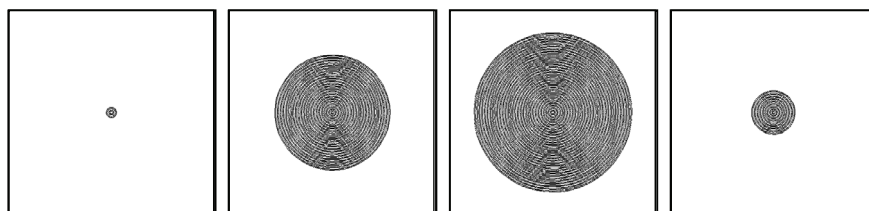
W przypadku realizacji sprzętowej czasy indeksacji powinny być stałe, gdyż zależą wyłącznie od częstotliwości zegara karty FPGA. Same transfery mogą wykazywać pewne odchylenia – transfer mógł być na przykład przerwany przez operacje dyskowe. Najbardziej skomplikowana sytuacja ma miejsce w przypadku realizacji programowej – czas indeksacji będzie silnie zależał od zawartości obrazu, przy czym trudno z góry przewidzieć, jaki obraz będzie „łatwy” do indeksacji, a jaki „trudny”. Oznacza to, że czasy etykietowania obrazów podobnych do siebie (np. jeden jest odwróconą o 180° wersją drugiego) mogą się znacząco różnić. Można to zauważyć, analizując wyniki uzyskane dla filmu „grzebień”.

Zamieszczone dalej wyniki uzyskano dla dwóch filmów – we wszystkich przypadkach rozmiary klatki obrazu wynosiły 512×512 pikseli. Tablica sklejeń miała rozmiar 4096, co pozwala na zaetykietowanie 4094 obiektów (wartość 0 jest zarezerwowana dla tła, a wartość 4095 dla obiektów, którym jeszcze nie nadano żadnej etykiety). Taka wielkość tablicy sklejeń pozwoliła z jednej strony na prawidłowe etykietowanie w zasadzie dowolnego obrazu o przyjętych rozmiarach, z drugiej zaś pozwoliła na jej umieszczenie w obszarze szybszej pamięci dostępnej na karcie FPGA. W pierwszym etapie badań etykietowanie było prowadzone na obrazach 32-bitowych (każdy piksel to liczba 32-bitowa). Oznaczało to konieczność transferowania pomiędzy pamięcią komputera PC a kartą FPGA bloków danych o wielkości 1 MB (512\*512\*4 bajty). Ponieważ w trakcie badań okazało się, że czasy transferów są porównywalne z czasem samej indeksacji, zrealizowano drugi etap badań, w którym zmniejszono wielkość piksela do 16 bitów. Pozwoliło to dwukrotnie zmniejszyć ilość transferowanych danych, co powinno skutkować dwukrotnym skróceniem czasów transferów.

Do testów wygenerowano dwa rodzaje filmów, które umownie nazwano: „Grzebień” oraz „Okręgi”. Przykłady klatek z ww. filmów przedstawiono na rysunkach 2 i 3.



Rys. 2. Przykłady kadrów (negatywy) z filmu „Grzebień”

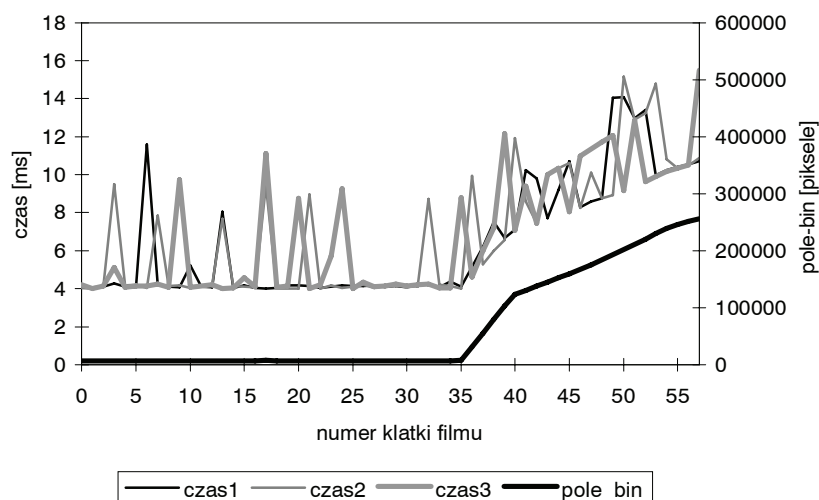


Rys. 3. Przykłady kadrów (negatywy) z filmu „Okregi”

Eksperymenty zostały przeprowadzone dla obu filmów oraz dla sześciu konfiguracji sprzętowo-programowych:

- 1) Athlon 2.4 GHz + Win XP, piksel 32-bitowy, indeksacja programowa i FPGA;
- 2) Athlon 2.4 GHz + Win XP, piksel 16-bitowy, indeksacja programowa i FPGA;
- 3) Athlon 2.4 GHz + Win XP, piksel 32-bitowy, indeksacja FPGA, trzy różne częstotliwości zegara;
- 4) Pentium III 800 MHz + W2K, piksel 32-bitowy, indeksacja programowa i FPGA;
- 5) Pentium III 800 MHz + W2K, piksel 16-bitowy, indeksacja programowa i FPGA;
- 6) Pentium III 800 MHz + W2K, piksel 32-bitowy, indeksacja FPGA, trzy różne częstotliwości zegara.

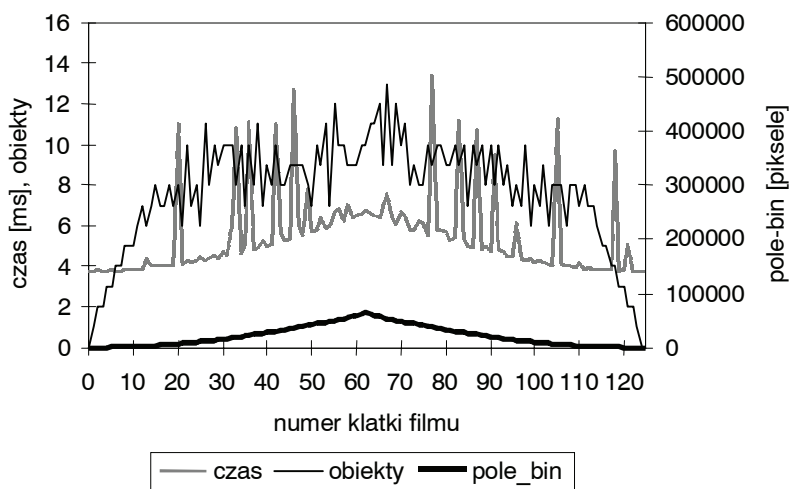
Wykres zmian czasów wykonania algorytmu realizowanego programowo (1) dla kolejnych klatek filmu „Grzebień” przedstawiono na rysunku 4.



Rys. 4. Film „Grzebień”: czasy wykonania indeksacji programowej (trzy powtórzenia) oraz rozmiar indeksowanego obiektu (krzywa *pole\_bin*). Eksperyment (1)

W pierwszym teście zauważono występowanie dużych wahań czasów wykonania algorytmu, dlatego też ten sam eksperyment powtórzono trzykrotnie. Jak widać, zmiany są chwilowe i występują nieregularnie. Trudno je skorelować z zawartością poszczególnych klatek obrazu, dlatego najprawdopodobniej przyczyną tych nieregularności jest system Windows, który w tym czasie wykonuje własne procesy o różnych priorytetach. Przyjmując za rzeczywiste czasy wykonania procedury indeksacji, minimalne czasy, wyznaczone w oknach o długości rzędu pięciu klatek, można stwierdzić istnienie trzech zakresów zmian tych czasów, skorelowanych jedynie z wielkością indeksowanego obiektu (por. rys. 4 krzywa *pole-bin*), a nie z położeniem obiektu. W przypadku analizowanego filmu „Grzebień” w czasie, gdy obiekt jest obracany (klatki o numerach 0–35) rzeczywisty czas indeksacji wynosi około 4 ms. Czas ten rośnie do około 10 ms proporcjonalnie do wzrostu pola obiektu.

Na rysunku 5 przedstawiono wyniki pojedynczego eksperymentu (1) zrealizowanego na filmie „Okregi”. Wyniki są bardzo podobne do poprzednich. Rzeczywiste czasy wykonania programowej indeksacji zmieniają się od 4 do 6 ms, w zależności od rozmiaru pola analizowanych obiektów. Na tym filmie liczba obiektów zmienia się (od 0 do 12), co ilustruje krzywa **obiekty** na rysunku 5. Wydaje się, że zmiany te nie mają wpływu na czas obliczeń.



Rys. 5. Film „Okregi”: czasy wykonania indeksacji programowej (trzy powtórzenia) oraz rozmiar indeksowanego obiektu (krzywa *pole-bin*). Eksperyment (1)

Czasy wykonania algorytmu indeksacji na karcie FPGA powinny być takie same, niezależne od liczby i stopnia komplikacji obiektów, ani od ich powierzchni. Pomiar wykazują jednak pewne wahania tych czasów. Dzieje się tak pomimo dokonania podziału całkowitego czasu na trzy części: transmisji do i z układu FPGA oraz czasu realizacji przetwarzania na samej karcie. Zależności te ilustruje tabela 2.

**Tabela 2**  
Eksperyment (1). Średnie czasy transmisji i przetwarzania na karcie FPGA [ms]

Film	Czas całkowity	Odchyłki %	Transmisja do karty	Odchyłki %	Obliczenia w FPGA	Odchyłki %	Transmisja z karty	Odchyłki %
„Okregi”	58,03	+27,90	13,16	+98,36	35,69	+36,17	9,18	+55,04
		-7,16		-25,44		-1,84		-2,52
„Grzebień”	57,07	+20,47	12,37	+64,11	35,44	+16,52	9,26	+43,11
		-5,62		-20,50		-1,17		-3,33

Procentowe odchyłki kolejnych czasów wyznaczono, obliczając wartości maksymalne i minimalne, a następnie odnosząc je do czasów średnich. W przypadku czasów transmisji można przyjąć, że opóźnienia spowodowane są przez system operacyjny. Interpretacja odchylek zaobserwowanych dla obliczeń wykonywanych na karcie FPGA wymaga dalszych badań.

W eksperymencie (2) zmiana rozmiaru pikselowej reprezentacji obrazu z 32 na 16 bitów nie wpłynęła na kształt wykresów prędkości przetwarzania filmu ani na szybkość analizy kolejnych ramek. Jedyne zmiany dotyczą przetwarzania sprzętowego, a konkretnie czasów transmisji ramek do i z karty FPGA. W tabeli 3, która jest odpowiednikiem tabeli 2 przedstawiono uzyskane wyniki. W związku z dwukrotnym zmniejszeniem wielkości pikseli obrazu odpowiednie czasy transmisji także zmniejszyły się w podobnej proporcji. Karta FPGA pracowała z częstotliwością 50 MHz<sup>3)</sup>.

**Tabela 3**  
Eksperyment (2). Średnie czasy transmisji i przetwarzania na karcie FPGA [ms]

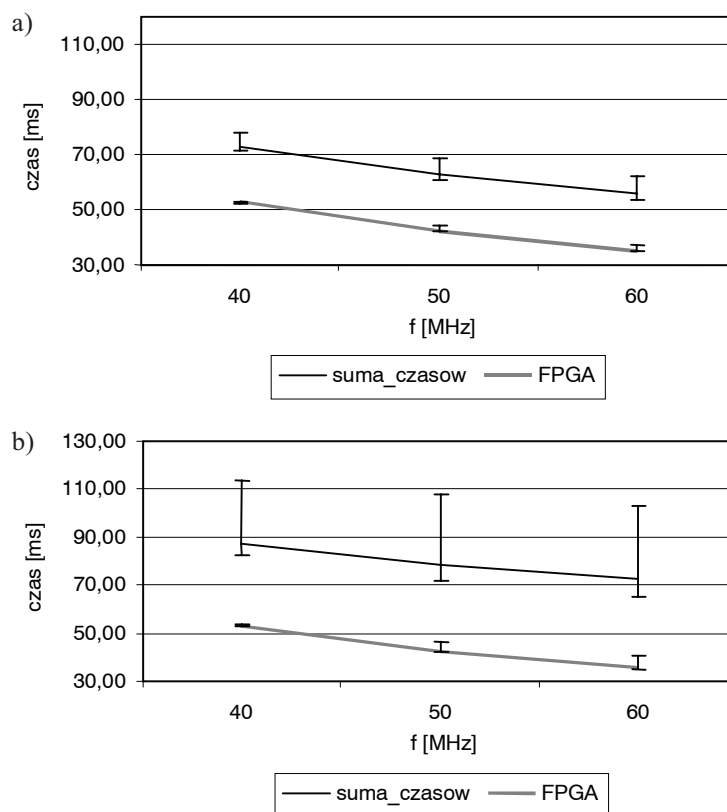
Film	Czas całkowity	Odchyłki %	Transmisja do karty	Odchyłki %	Obliczenia w FPGA	Odchyłki %	Transmisja z karty	Odchyłki %
„Okregi”	53,63	+15,76	6,76	+127,63	42,21	+2,56	4,66	+34,58
		-3,61		-25,15		-0,48		-1,48
„Grzebień”	53,46	+16,12	6,38	+106,39	42,27	+12,72	4,81	+141,91
		-3,32		-20,24		-0,64		-4,59

Eksperymenty (3) oraz (6) miały na celu pokazanie, jak zmieniają się czasy indeksacji obrazów przez kartę FPGA, w zależności od częstotliwości taktowania układu FPGA. Badania przeprowadzono dla 32-bitowej reprezentacji pikseli filmów („Grzebień”) oraz dla

<sup>3)</sup> Częstotliwość została zmniejszona na wniosek autora oprogramowania FPGA.

dwóch środowisk programowo-sprzętowych: komputera z procesorem Athlon 2.4 GHz z systemem operacyjnym Windows XP (3) oraz komputera z procesorem Pentium III 800 MHz z Windows 2000 (6).

Wyniki przedstawiono na rysunku 6. W punktach pomiarowych wykresów zaznaczono wartości minimalne i maksymalne, wyznaczone dla całego filmu.



**Rys. 6.** Wpływ zmian częstotliwości taktowania układu FPGA na czasy indeksacji:  
a) eksperyment (3); b) eksperyment (6)

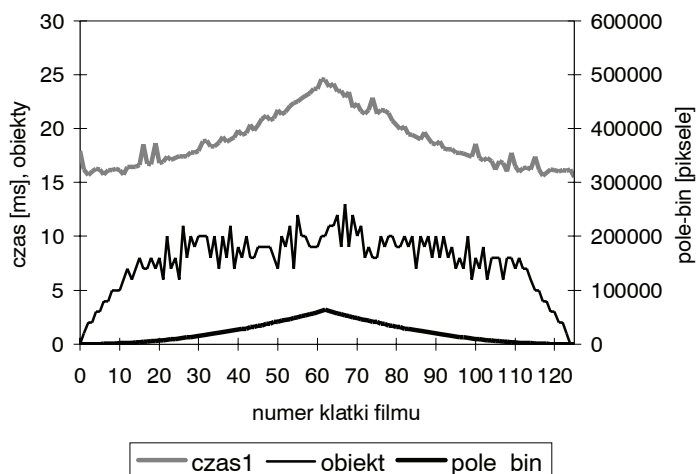
Z postaci wykresów (i otrzymanych wyników liczbowych nasuwa się kilka spostrzeżeń:

- średnie czasy realizacji algorytmu na karcie FPGA są podobne dla obu środowisk, uwagę zwracają jednak zwiększone odchyłki (szczególnie wartości maksymalne);
- sumaryczne czasy (łącznie z transmisjami) są nieco mniejsze w przypadku szybszego komputera, co wydaje się oczywiste; natomiast duże odchyłki czasów transmisji dla środowiska Pentium 800 + W2K są prawdopodobnie związane z działaniem systemu operacyjnego.

Eksperymenty (4) i (5) zostały przeprowadzone w środowisku: komputer z procesorem Pentium 800 MHz + W2K. W związku ze zmniejszeniem szybkości procesora zwią-



szyły się także czasy programowej realizacji operacji segmentacji (blisko cztery razy). Charakter zmian czasów wykonania segmentacji na kolejnych ramkach nie zmienił się, wydatnie poprawiła się natomiast powtarzalność poszczególnych wyników. Przykład wykresu, dotyczący filmu „Okregi” przedstawiono na rysunku 7.



Rys. 7. Film „Okregi”: czasy wykonania indeksacji programowej (3 powtórzenia) oraz rozmiar indeksowanego obiektu (krzywa *pole-bin*). Eksperyment (4)

Rysunek 7 należy porównać z rysunkiem 5. Wtedy wyraźnie widać zmniejszenie się zakłóceń, związanych z czynnościami systemu operacyjnego. Podobne tendencje zaobserwowano na wszystkich wykresach, ilustrujących wyniki eksperymentów (4) i (5). Wykresy są bardziej „gładkie”, natomiast zakłócenia – o ile występują – są bardzo rzadkie, za to duże.

W tabeli 4 (por. tab. 2) pokazano czasy transmisji i przetwarzania na karcie FPGA dla eksperymentu (4). Odchyłki czasów transmisji są na podobnym poziomie procentowym jak w eksperymencie (1). Odchyłki czasów obliczeń na samej karcie są nieco mniejsze. Wyniki eksperymentu (5) dla realizacji sprzętowej są podobne do wyników eksperymentu (2). Podczas obu tych eksperymentów używano 16-bitowych pikseli, a układ FPGA pracował z częstotliwością 50 MHz.

Tabela 4  
Eksperyment (4). Średnie czasy transmisji i przetwarzania na karcie FPGA [ms]

Film	Czas całkowity	Odchyłki %	Transmisja do karty	Odchyłki %	Obliczenia w FPGA	Odchyłki %	Transmisja z karty	Odchyłki %
„Okregi”	72,25	+27,04	22,55	+79,32	35,65	+9,50	14,06	+53,45
		-10,29		-26,31		-1,51		-7,68
„Grzebień”	74,57	+25,13	23,90	+70,88	35,68	+17,08	14,99	+57,99
		-12,72		-28,90		-1,60		-13,48

## 5. Podsumowanie

Głównym celem opisywanych badań było sprawdzenie, czy jest możliwe sprzętowe wspomaganie algorytmu etykietowania (indeksacji) obrazów (filmów) realizowanego w środowisku VirtualDub. W tym celu zaimplementowano dwa filtry: jeden realizujący etykietowanie programowo oraz drugi, który transmituje obrazy do karty FPGA, gdzie indeksacja realizowana jest sprzętowo. Zrealizowano sześć serii eksperymentów w oparciu o dwa rodzaje wygenerowanych filmów i dwie platformy sprzętowo-programowe.

Wyniki najbardziej znaczących eksperymentów w skondensowanej formie przedstawiono w tabelach 5 i 6. Skoncentrowano się na porównaniu czasów przetwarzania dla obu filmów, obu platform oraz 32-bitowej reprezentacji pikseli obrazu. Do sprzętowej indeksacji wykorzystano kartę FPGA pracującą z częstotliwością 60 MHz.

**Tabela 5**  
Porównanie czasów indeksacji (Athlon 2.4 GHz, Windows XP) – czasy w [ms]

Film (Liczba klatek)	Program		Sprzęt			
	Czas średni	Czas maks. Czas min	Czas całkowity	Odchyłki %	Obliczenia w FPGA	Odchyłki %
„Okregi”	11,65	18,06	58,03	+27,90	35,69	+36,17
(128)		10,34		-7,16		-1,84
„Grzebień”	6,81	15,50	57,07	+20,47	35,44	+16,52
(60)		4,01		-5,62		-1,17

**Tabela 6**  
Porównanie czasów indeksacji (Pentium 3 800 MHz, Windows 2000) – czasy w [ms]

Film (Liczba klatek)	Program		Sprzęt			
	Czas średni	Czas maks. Czas min	Czas całkowity	Odchyłki %	Obliczenia w FPGA	Odchyłki %
„Okregi”	35,09	37,67	72,25	+27,04	35,65	+9,50
(128)		33,75		-10,29		-1,51
„Grzebień”	21,48	35,50	74,57	+25,13	35,68	+17,08
(60)		16,18		-12,72		-1,60

Z analizy tabel 5 i 6 wynika, że czasy realizacji algorytmu segmentacji przez kartę FPGA są w przybliżeniu stałe, niezależne od stopnia komplikacji przetwarzanych klatek. Wynik ten nie zaskakuje, niemniej warto przeprowadzić dodatkowe eksperymenty, mające

na celu wyjaśnienie dość znacznych, chwilowych odchyłek. Z uwagi na fakt zastosowania do obliczeń karty FPGA nienajnowszej generacji, eksperymenty zrealizowano na dwóch komputerach. Tylko w jednym przypadku (film „Określi”) realizacja sprzętowa jest realizowana w zbliżonym czasie co programowa (komputer Pentium 3 800 MHz). Warto zaznaczyć, że powyższe zdanie jest prawdziwe, jeśli zaniedba się czasy transferu obrazów do i z karty FPGA, które w przybliżeniu podwajają czas realizacji sprzętowej.

Realizacja programowa algorytmu segmentacji jest – dla komputera Athlon 2.4 GHz – kilkakrotnie szybsza od realizacji sprzętowej. Czasy obliczeń są jednak bardzo różnicowane i zależą głównie od stopnia skomplikowania analizowanych obrazów, a w drugiej kolejności także od wielkości obiektów (liczby pikseli). Na wykresach można także zaobserwować bardzo duże, przekraczające 100% odchyłki czasów obliczeń dla poszczególnych klatek. Są one związane z funkcjonowaniem systemu operacyjnego.

Podsumowując: sprzętowe wspomaganie indeksacji obrazów w środowisku VirtualDub jest jak najbardziej możliwe, chcąc jednak doprowadzić do przetwarzania obrazów w czasie rzeczywistym, konieczne jest zastosowanie najnowszej generacji układów FPGA o znacznie zwiększonej mocy obliczeniowej, oraz magistrali komputera o większej przepływności.

## Literatura

- [1] Adamski A.: *Inteligentne systemy transportowe: sterowanie, nadzór i zarządzanie*. Kraków, UWND AGH, 2003
- [2] Adamski A., Bubleński Z., Mikrut Z., Pawlik P.: *Image Analysis Based Automatic Safety Monitoring System For Traffic Intersections*. IV Konferencja „Telematyka i Bezpieczeństwo Ruchu Drogowego”, 19–20 XI 2004, Katowice–Ustroń, 2004
- [3] Hoose N.: *Computer Image Processing In Traffic Engineering*. RSP Ltd., Taunton, 1991
- [4] Jabłoński M.: *Od obrazu do matrycy znaku – Implementacja sprzętowa*. Seminarium Wyjazdowe nt.: Przetwarzanie i analiza sygnałów w systemach wizji i sterowania, Słok k/Bełchatowa, 2002, 38–43
- [5] Jabłoński M., Gorgoń M.: *Handel-C implementation of classical component labelling algorithm*. Proceedings of the Euromicro Systems on Digital System Design, Rennes, 2004, 387–393
- [6] Lee A.: *VirtualDub project homepage*. <http://www.virtualdub.org>, Internet, 2004
- [7] Parker J.R.: *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing, New York, 1997
- [8] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997
- [9] Umbaugh S.E.: *Computer Vision and Image Processing*. Prentice Hall, Upper Saddle River, 1998
- [10] [www.celoxica.com](http://www.celoxica.com)
- [11] [www.xilinx.com](http://www.xilinx.com)

