

Gdańsk, dn. 18.01.2016 r.

dr inż. Marek Wójcikowski  
Politechnika Gdańska  
Wydział Elektroniki, Telekomunikacji i Informatyki  
Katedra Systemów Mikroelektronicznych  
ul. Narutowicza 11/12  
80-233 Gdańsk  
tel. 58 347 19 74  
faks 58 347 23 78  
e-mail: [wujek@ue.eti.pg.gda.pl](mailto:wujek@ue.eti.pg.gda.pl)

## AUTOREFERAT

### Wykształcenie, uzyskane stopnie i tytuły naukowe

- 1988 - 1993      Studia magisterskie na Wydziale Elektroniki Politechniki Gdańskiej, specjalność Układy Elektroniczne.
- 1993              Obrona pracy magisterskiej pod tytułem "Projektowanie filtrów SI (*Switched Current*)".
- 26 lutego 2002r.      Uzyskanie stopnia doktora w dziedzinie elektroniki po obronie pracy doktorskiej pt. "Obiektowy system do wspomagania projektowania filtrów analogowych CMOS", Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej.

### Doświadczenie zawodowe

- 1994 - 2002      Katedra Układów Elektronicznych, Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska - stanowisko: asystent.
- 2002 – dzisiaj      Katedra Systemów Mikroelektronicznych, Wydział Elektroniki, Telekomunikacji i Informatyki, Politechnika Gdańska - stanowisko: adiunkt.

### Osiągnięcie naukowe habilitanta przedstawione do oceny:

Zbiór publikacji powiązanych tematycznie.

### Tytuł osiągnięcia naukowego:

Projektowanie mikroelektronicznych niskomocowych wizyjnych systemów sensorowych.

### Lista publikacji powiązanych tematycznie

Oświadczenia współautorów dotyczące indywidualnego wkładu w powstanie poszczególnych publikacji znajdują się w odrębnym załączniku.

## Publikacje z listy JCR

Lp.	Autorzy, tytuł publikacji, nazwa wydawnictwa, rok wydania, indywidualny wkład autora	Impact factor
P1	<p><b>M. Wójcikowski</b>, R. Żaglewski, B. Pankiewicz, "FPGA-Based Real-Time Implementation of Detection Algorithm for Automatic Traffic Surveillance Sensor Network", <i>Journal of Signal Processing Systems</i>, vol. 68, iss. 1, 2012, pp. 1-18.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- ogólna koncepcja algorytmu odejmowania tła</li> <li>- realizacja sprzętowa modelu odejmowania tła z wykorzystaniem bieżącej średniej</li> <li>- wykorzystanie transformacji Hough z prostokątnym elementem strukturalnym do finalnego przetworzenia obrazu maski</li> <li>- analiza zużycia zasobów i mocy algorytmu</li> <li>- porównanie realizacji 4- i 8-bitowej algorytmu</li> </ul> <p>Procentowy wkład Autora: 40%</p> <p>Liczba autorów: 3</p>	0.551
P2	<p><b>M. Wójcikowski</b>, R. Żaglewski, B. Pankiewicz, M. Kłosowski, S. Szczepański, "Hardware-Software Implementation of a Sensor Network for City Traffic Monitoring Using the FPGA- and ASIC-Based Sensor Nodes", <i>Journal of Signal Processing Systems</i>, 71(1), 2013, pp. 57-73.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- ogólna koncepcja systemu</li> <li>- realizacja sprzętowa modelu odejmowania tła z wykorzystaniem bieżącej średniej</li> <li>- wykorzystanie transformacji Hough z prostokątnym elementem strukturalnym do finalnego przetworzenia obrazu maski</li> <li>- koncepcja wieloprocesorowego przetwarzania danych obrazu</li> <li>- realizacja śledzenia wysokopoziomowego wykrytych obiektów</li> <li>- koncepcja i realizacja protokołu i transmisji radiowej w sieci sensorowej wraz z testowaniem i uruchomieniem</li> <li>- współudział w realizacji sprzętowej systemu z wykorzystaniem układu FPGA</li> <li>- współudział w projektowaniu układu ASIC</li> <li>- współudział w weryfikacji układu ASIC</li> <li>- współudział w realizacji sprzętowej systemu z wykorzystaniem układu ASIC</li> <li>- koncepcja i realizacja symulatora sieci sensorowej</li> <li>- testy w terenie</li> </ul> <p>Procentowy wkład Autora: 30%</p> <p>Liczba autorów: 5</p>	0.564
P3	<p>S. Szczepański, <b>M. Wójcikowski</b>, B. Pankiewicz, M. Kłosowski, R. Żaglewski, "FPGA and ASIC implementation of the algorithm for traffic monitoring in urban areas", <i>Bull. Pol. Acad. Sci. Tech. Sci.</i>, vol. 59, iss. 2, 2011, pp. 137-140.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- ogólna koncepcja systemu</li> <li>- realizacja sprzętowa modelu odejmowania tła z wykorzystaniem bieżącej średniej</li> <li>- realizacja śledzenia wysokopoziomowego wykrytych obiektów</li> </ul>	0.966

	<ul style="list-style-type: none"> <li>- współdział w realizacji sprzętowej systemu z wykorzystaniem układu FPGA</li> <li>- współdział w projektowaniu układu ASIC</li> <li>- współdział w weryfikacji układu ASIC</li> <li>- współdział w realizacji sprzętowej systemu z wykorzystaniem układu ASIC</li> <li>- testy w terenie</li> </ul> <p><u>Procentowy wkład Autora:</u> 30%</p> <p><u>Liczba autorów:</u> 5</p>	
P4	<p><b>M. Wójcikowski</b>, "Transmission Protocol Simulation Framework for the Resource-Constrained Wireless Sensor Network", Metrology and Measurements Systems, vol. XXII, iss. 2, 2015, pp. 221-228.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- opracowanie emulatora pojedynczego węzła sieci sensorowej</li> <li>- opracowanie symulacji transmisji radiowej pomiędzy osadzonymi emulowanymi węzłami sieci sensorowej</li> <li>- realizacja systemu symulatora sieci sensorowych</li> <li>- wykonanie interfejsu GUI symulatora</li> <li>- wykonanie demonstracji działania autokonfiguracji i wykrywania węzłów sąsiednich (bezpośrednich i niebezpośrednich) w sieci sensorowej</li> </ul> <p><u>Procentowy wkład Autora:</u> 100%</p> <p><u>Liczba autorów:</u> 1</p>	0.925
P5	<p>M. Kłosowski, <b>M. Wójcikowski</b>, A. Czyżewski, "Vision-Based Parking Lot Occupancy Evaluation System Using 2D Separable Discrete Wavelet Transform", Bull. Pol. Acad. Sci. Tech. Sci., vol. 63, iss. 3, 2015, pp. 569-573.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- implementacja algorytmu DWT w sprzęcie</li> <li>- optymalizacja współczynników algorytmu z wykorzystaniem algorytmu genetycznego</li> <li>- przygotowanie obrazów testowych, uczenie i testowanie systemu</li> <li>- testy w terenie</li> </ul> <p><u>Procentowy wkład Autora:</u> 40%</p> <p><u>Liczba autorów:</u> 2</p>	0.914
P6	<p><b>M. Wójcikowski</b>, "Histogram of Oriented Gradients with Cell Average Brightness for Human Detection", accepted for publication in Metrology and Measurements Systems, iss. 1, 2016.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- opracowanie modyfikacji algorytmu HOG (<i>Histogram of Oriented Gradients</i>) do detekcji osób w obrazach statycznych, zwiększającej dokładność działania algorytmu</li> <li>- programowa implementacja zmodyfikowanego algorytmu</li> <li>- testowanie algorytmu, pomiar efektywności działania</li> </ul>	0.925
<b>Sumaryczny Impact Factor:</b>		4.845

Do publikacji najnowszych z roku 2015 i 2016 przypisano współczynniki *Impact Factor* za rok 2014.

## Publikacje konferencyjne

Lp.	Autorzy, tytuł publikacji, rok wydania, nazwa wydawnictwa lub konferencji, indywidualny wkład autora	Indeksowane
K1	<p>M. Musiał, D. Dybek, <b>M. Wójcikowski</b>, "Hardware realization of shadow detection algorithm in FPGA", Zeszyty Naukowe Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej, vol. 18, 2010, pp. 391-396.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- koncepcja systemu detekcji cieni</li> <li>- konsultacje dotyczące realizacji systemu</li> <li>- pomoc w realizacji sprzętowej</li> <li>- przygotowanie i dostarczenie danych testowych</li> </ul> <p><u>Procentowy wkład Autora:</u> 10%</p> <p><u>Liczba autorów:</u> 3</p>	
K2	<p><b>M. Wójcikowski</b>, B. Pankiewicz, " Comparison of software and hardware realization of AES cryptographic algorithm", Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej. Technologie Informacyjne, t. 14, 2007, pp. 589-592.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- sprzętowa realizacja algorytmu szyfrowania</li> <li>- programowa realizacja algorytmu – do porównania</li> <li>- badania wydajności i poboru mocy</li> </ul> <p><u>Procentowy wkład Autora:</u> 50%</p> <p><u>Liczba autorów:</u> 2</p>	
K3	<p><b>M. Wójcikowski</b>, B. Pankiewicz, "ASIC Design Example of Complex SoC with FPGA Prototyping", Przegląd Elektrotechniczny, nr 10, 2013, pp. 156-158.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- współudział w projektowaniu systemu w technologii FPGA</li> <li>- projektowanie układu ASIC – synteza,</li> <li>- rozprowadzanie zasilania w topografii układu ASIC</li> <li>- weryfikacja układu ASIC na etapie projektu topografii</li> </ul> <p><u>Procentowy wkład Autora:</u> 50%</p> <p><u>Liczba autorów:</u> 2</p>	Lista B
K4	<p>M. Kłósowski, B. Pankiewicz, <b>M. Wójcikowski</b>, "Cyfrowy akcelerator wybranych modułów standardu kompresji wideo H.264", Przegląd Elektrotechniczny, nr 9, 2014, pp. 54-57.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- współudział w projektowaniu układu ASIC</li> <li>- synteza układu ASIC</li> <li>- projekt rozprowadzenia zasilania w układzie ASIC</li> <li>- weryfikacja układu ASIC podczas projektowania</li> </ul> <p><u>Procentowy wkład Autora:</u> 15%</p> <p><u>Liczba autorów:</u> 3</p>	Lista B
K5	<p>R. Żaglewski, <b>M. Wójcikowski</b>, "Multi-core processing system for real-time image processing in embedded computer vision applications", Proceedings of the 1st International Conference on Information Technology Gdańsk, Gdansk University of Technology, 19-21 May 2008, pp. 115-118.</p> <p><u>Indywidualny wkład Autora:</u></p>	Web of Science, Core Collection

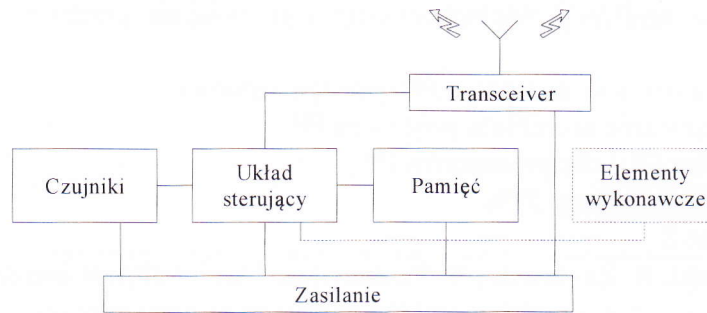
	<ul style="list-style-type: none"> <li>- koncepcja realizacji wieloprocesorowego systemu przetwarzania obrazu</li> <li>- koncepcja budowy procesora PP (<i>pixel processor</i>)</li> <li>- oprogramowanie assemblera procesora PP</li> <li>- środowisko GUI dla procesorów PP</li> </ul> <p><u>Procentowy wkład Autora: 50%</u> Liczba autorów: 2</p>	
K6	<p><b>M. Wójcikowski</b>, R. Żaglewski, B. Pankiewicz, "An intelligent image processing sensor - the algorithm and the hardware implementation", Proceedings of the 1st International Conference on Information Technology Gdańsk, Gdańsk University of Technology, 19-21 May 2008, pp. 377-380.</p> <p><u>Indywidualny wkład Autora:</u></p> <ul style="list-style-type: none"> <li>- koncepcja systemu detekcji obiektów ruchomych</li> <li>- współdziałł w realizacji sprzętowej algorytmu – moduł odejmuwania tła, moduł transformacji Hough</li> </ul> <p><u>Procentowy wkład Autora: 40%</u> Liczba autorów: 3</p>	Web of Science, Core Collection

## 2.1. Wprowadzenie do tematyki badań habilitanta

Habilitant jest adiunktem w Katedrze Systemów Mikroelektronicznych Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej. Niniejszy autoreferat jest podsumowaniem oryginalnego dorobku autora (habilitanta) opracowanego w głównym nurcie jego działań badawczych dotyczących mikroelektronicznych systemów sensorowych. Autor, po obronie w 2002 roku rozprawy doktorskiej (w dyscyplinie elektronika) pt. „Obiektowy system do wspomagania projektowania filtrów analogowych CMOS”, brał udział w projektach dotyczących projektowania układów i systemów programowalnych [G3][G5][G6], scalonych [G2][G4][G5][G6] oraz realizowanych sprzętowo i programowo algorytmów przetwarzania obrazów [G5][G6][G7].

Udział w wymienionych projektach pozwala na wyodrębnienie znaczącego, autorskiego, indywidualnego wkładu w rozwój dyscypliny elektronika w zakresie badań nad mikroelektronicznymi systemami sensorowymi wyposażonymi w sensory obrazu. Niniejszy autoreferat opisuje tematycznie powiązany cykl artykułów opublikowanych na przestrzeni ostatnich lat, które świadczą o znacznym wkładzie naukowym autora w reprezentowaną dyscyplinę nauk technicznych – elektronikę. Publikacje te opisują projekty i badania dotyczące układów i systemów mikroelektronicznych sieci sensorowych i algorytmów przetwarzania obrazów dedykowanych dla sieci sensorowych.

Miniaturyzacja układów i systemów elektronicznych, ich malejący koszt, coraz mniejsze zapotrzebowanie na moc zasilania oraz postęp technologiczny w dziedzinie baterii i pozyskiwania energii z otoczenia sprawiły, iż w ostatnich latach stała się możliwą realizacja tzw. sieci sensorowych. Sieć sensorową definiuje się jako zbiór miniaturowych, autonomicznych urządzeń, wyposażonych w sensory, podukłady łączności bezprzewodowej oraz własne źródło zasilania [1][2]. Urządzenia wchodzące w skład sieci sensorowej nazywane są węzłami tej sieci. Schemat blokowy węzła sieci sensorowej przedstawiono na Rys. 1.



Rys. 1. Schemat blokowy węzła sieci sensorowej

Charakterystyczną cechą sieci sensorowej jest brak odrębnej infrastruktury umożliwiającej przesyłanie danych – infrastruktura jest tworzona wyłącznie przez węzły sieci. Węzły sieci sensorowej instaluje się w środowisku, z którego sieć ma zbierać dane. Instalacja węzłów w środowisku może przebiegać w różny sposób, np. może polegać na dystrybucji węzłów nad dużym obszarem bezpośrednio z samolotu (np. w celu monitorowania pożaru lasu), lub może się również odbywać np. w fabryce, podczas produkcji lub budowy turbiny, rakiety czy dużej konstrukcji budowlanej (budynek, most), gdzie stosowanie czujników przewodowych jest kłopotliwe lub niemożliwe.

W zależności od obszaru zastosowań, węzły sieci sensorowe wyposaża się w odpowiednie czujniki - mogą to być proste czujniki np. światła, temperatury, drgań, lub czujniki o wyższym stopniu skomplikowania, np. detektory promieniowania czy detektory obrazu.

Węzły sieci sensorowej odbierają dane z czujników oraz wstępnie je przetwarzają. Zgodnie z filozofią sieci sensorowej, dla użytkownika sieci sensorowej, najważniejsze są dane pochodzące z sieci, a nie dokładny adres węzła, który je zarejestrował; w niektórych przypadkach węzły sieci sensorowej mogą nie mieć swoich unikalnych adresów. Dane są przetwarzane i przesyłane przez sieć, a następnie dostarczane odbiorcy danych w formie gotowych wyników lub wniosków, np. w postaci mapy.

Do komunikacji między węzłami najczęściej służą transceivery radiowe o małej mocy nadawania, posiadające niewielki zasięg ze względu na ograniczenia wynikające z zasilania baterijnego. W literaturze można również spotkać rozwiązania bazujące na komunikacji z wykorzystaniem promieni świetlnych i luster MEMS np. [3]. Krótki zasięg transmisji radiowej, w połączeniu z rozległym obszarem instalacji sieci, wymusza przeprowadzanie lokalnych transmisji pomiędzy nieodległymi węzłami, w związku z tym transmisja danych w ramach całej sieci sensorowej może się odbywać poprzez wiele węzłów pośrednich, które jednocześnie mogą dodawać do transmisji własne dane. Podział transmisji na wiele skoków (ang. *multi-hop*) ma uzasadnienie, gdyż pod pewnymi założeniami transmisja wieloskokowa jest bardziej oszczędna energetycznie w porównaniu do pojedynczej transmisji na dużą odległość. Zakładając konieczność przesłania danych na odległość  $d$  i korzystając ze wzoru na propagację fal w wolnej przestrzeni, energia  $E_1$  potrzebna do takiej transmisji wynosi [1]:

$$E_1 = Cd^\alpha \quad (1)$$

gdzie  $C$  oznacza pewną stałą, a dla stałej  $\alpha$  przyjmuje się wartość  $\alpha > 2$ .

Energia  $E_2$  potrzebna do transmisji na odległość  $d$ , ale podzielonej na dwa skoki o długości  $d/2$  wynosi:

$$E_2 = 2C\left(\frac{d}{2}\right)^\alpha \quad (2)$$

Porównując ze sobą energie  $E_1$  i  $E_2$  ze wzorów (1) i (2) otrzymujemy:

$$\frac{E_1}{E_2} = \frac{Cd^\alpha}{2C\left(\frac{d}{2}\right)^\alpha} = 2^{\alpha-1} \quad (3)$$

Ze wzoru (3) oraz założenia  $\alpha > 2$  wynika, że transmisja 2-skokowa jest ponad 2-krotnie korzystniejsza energetycznie, przy założeniu, że nie uwzględniamy energii zużytej przez odbiornik węzła pośredniczącego.

Sieć sensorowa zazwyczaj składa się z węzłów o identycznej budowie, chociaż możliwa jest realizacja sieci z dwoma rodzajami węzłów: węzły w pełni wyposażone oraz węzły pozbawione czujników i elementów wykonawczych, służące jedynie do zapewnienia komunikacji pomiędzy innymi węzłami. W sieci możemy wyróżnić węzły będącymi źródłami dostarczającymi dane do sieci oraz węzły odbierające dane (tzw. *data sink*). Czasami występują jeszcze węzły wyposażone w elementy wykonawcze. Zazwyczaj źródła danych, czyli węzły dostarczające dane do sieci, stanowią większość węzłów sieci.

Połączenie sieci sensorowej z istniejącymi sieciami telekomunikacyjnymi i zapewnienie dostępu do danych z komputerów użytkowników, odbywa się za pomocą dodatkowych urządzeń, tzw. bram (*gateway*), które pośredniczą w przepływie danych, zapewniając również odpowiednią translację adresów. Bramy mogą również służyć do łączenia ze sobą kilku odległych sieci sensorowych za pośrednictwem istniejących sieci telekomunikacyjnych.

Sieć sensorowa służy do zbierania danych ze środowiska, w szczególności sieć może wykrywać zdarzenia występujące na monitorowanym obszarze (np. wykrywanie intruzów), może także przeprowadzać regularne pomiary (np. temperatury, wilgotności). Bardziej zaawansowane aplikacje mogą obliczać aproksymacje rozkładu mierzonych wartości w obserwowanym obszarze lub przestrzeni, np. prezentując użytkownikowi mapę temperatury, wykrywać gradienty i krawędzie, a także śledzić obiekty. Sieć sensorowa może być wyposażona w bazę danych, zarówno lokalną dla węzła sieci lub wspólną dla całej sieci. Przykładem rozproszonej bazy danych może być rozwiązanie stosowane w sieciach typu *peer-to-peer*, takich jak BitTorrent, działające zgodnie ze schematem *Distributed Hash Table* [4]. Baza danych może być również zorganizowana poza siecią sensorową, np. jako baza danych wykorzystująca SQL (*Structured Query Language*), działająca na stacji roboczej dołączonej do węzła odbierającego dane.

Projektowanie sieci sensorowej jest obciążone ciągłymi zmaganiem z wszelkiego rodzaju kompromisami. Te kompromisy są wymuszone przez następujące czynniki: ograniczona moc zasilania bateryjnego, wymagania na niewielkie wymiary i koszt węzła sieci, brak infrastruktury sieciowej i centralnego koordynatora, krótki zasięg komunikacji radiowej, możliwość przemieszczania się uczestników systemu. Przemieszczać się mogą źródła danych (np. węzły sieci sensorowej umieszczone na pojazdach lub zwierzętach), odbiorcy danych oraz źródła rejestrowanych zdarzeń.

Niewielka moc zasilania bateryjnego, połączona z wymaganiem na długi czas działania urządzenia, wymaga od projektanta zastosowania rozwiązań minimalizujących moc pobieraną z zasilania przez węzeł. Zmniejszenie poboru mocy możemy osiągać na wielu płaszczyznach – np. poprzez wybór niskomocowych technologii realizacji układów scalonych, zmniejszenie napięć zasilających, zmniejszenie częstotliwości zegarów taktujących system, czy wykorzystanie możliwości przechodzenia w różne tryby uśpienia systemu. Obok stosowania niskomocowych technologii realizacji układów scalonych z wykorzystaniem bramek i przerzutników typu *low-power*, stosuje się także dodatkowe techniki obniżenia poboru mocy z zasilania, takie jak *clock-gating* (CG), *power shutoff* (PSO), *multi-supply voltage* (MSV),

multiple  $V_t$ , *adaptive voltage scaling* (AVS), czy *dynamic voltage frequency scaling* (DVFS) [5][6].

Bardzo ważna jest także minimalizacja poboru mocy z zasilania na poziomie protokołu transmisyjnego. Duża część mocy zasilania jest przeznaczana w sieci sensorowej na transmisje radiowe. Transceivery posiadają nadajniki małej mocy i często wartość prądu pobieranego z zasilania w czasie nadawania jest porównywalna lub nawet mniejsza od prądu pobieranego w czasie odbierania danych. Utrzymywanie włączonego odbiornika przez cały czas, nawet, gdy nie ma żadnych danych do odebrania (tzw. *idle listening*), jest nieefektywne energetycznie, chociaż mogłoby być wygodne ze względu na konieczność zapewnienia ciągłej komunikacji między węzłami. Idealnym byłby odbiornik, który nie pobiera mocy z zasilania, a włącza się tylko w momencie, gdy odbywa się transmisja radiowa przeznaczona dla niego. Niestety jest tu sprzeczność, gdyż, aby wykryć transmisję radiową, odbiornik musi być włączony. W literaturze można spotkać rozwiązania wykorzystujące dwa odbiorniki radiowe, gdzie prosty, energooszczędny odbiornik tzw. *wake-up receiver*, po wykryciu transmisji włącza główny odbiornik [7][8]. Istnienie dwóch odbiorników jest komplikacją i zwiększeniem kosztów, dlatego dąży się do stosowania jednego odbiornika, dostosowując odpowiednio protokół transmisji radiowej.

Rozwiązania problemu straty mocy podczas transmisji radiowej poszukuje się poprzez rozwój energooszczędnych protokołów transmisji danych. Stosowanie protokołów transmisyjnych wymaga nadawania i odbierania dodatkowych danych kontrolnych i sterujących, więc ważne jest, aby tzw. *protocol overhead* był jak najmniejszy. Protokół ma za zadanie zorganizować energooszczędną transmisję danych pomiędzy węzłami, jednocześnie zarządzając topologią sieci i zapewniając trasowanie pakietów oraz adresowanie węzłów i synchronizację czasu. Nasłuchiwanie przez węzeł sieci transmisji, która nie jest przeznaczona dla niego (tzw. *overhearing*) jest stratą energii i protokół powinien ograniczać takie sytuacje. Straty energii mogą również wynikać z kolizji pakietów danych nadawanych jednocześnie w tym samym kanale lub z takim samym kodem CSMA (*Carrier Sense Multiple Access*). Dodatkowym problemem w rozwoju protokołów jest skalowalność protokołu: liczba węzłów w sieci sensorowej  $n$ , gdzie  $n \gg 1$ , może zawierać się w zakresie zmienianym o kilka rzędów wielkości.

Szacuje się, że stosunek energii potrzebnej do wysłania 1 bitu danych do energii potrzebnej do wykonania jednej instrukcji w węzle sieci ma wartość 220-2900 [1][9], dlatego dąży się do przetwarzania jak największej ilości danych wewnątrz węzła sieci. Ze względu na koszty i ograniczenia zasilania, zazwyczaj węzły są wyposażane w jednostki obliczeniowe o małej mocy przetwarzania danych, więc w przypadku dużej ilości danych (np. strumień wideo), można zastosować dedykowane akceleratory sprzętowe [10][K2][K4].

Sieć sensorowa powinna działać przez czas odpowiedni dla aplikacji, w niektórych przypadkach wymiana baterii nie jest możliwa lub opłacalna (np. przypadku czujników instalowanych na stałe np. w konstrukcji budynków, gdzie nie ma możliwości wymiany baterii, wymaga się czasów działania nawet dochodzących do 50 lat). Czas życia sieci sensorowej może być różnie zdefiniowany: może to być czas, po jakim pierwszy z węzłów przestanie funkcjonować, czas po jakim 50% węzłów przestanie działać lub czas który upłynie do momentu, gdy sieć podzieli się na rozłączne części w wyniku wyłączenia węzłów. Odzyskiwanie energii z otoczenia jest często stosowanym rozwiązaniem w celu przedłużenia czasu działania sieci – w tym celu stosuje się baterie fotowoltaiczne, turbiny wiatrowe i gazowe, ogniwa paliwowe, wykorzystuje się także wibracje, zmiany nacisku (przetworniki piezo-elektryczne) lub gradienty temperatury [11].

Wszystkie aspekty działania węzła sieci sensorowej kontrolowane są przez oprogramowanie. Ze względu na energooszczędność, miniaturyzację, minimalizację kosztów oraz małą moc obliczeniową jednostki centralnej węzła, unika się stosowania typowych systemów

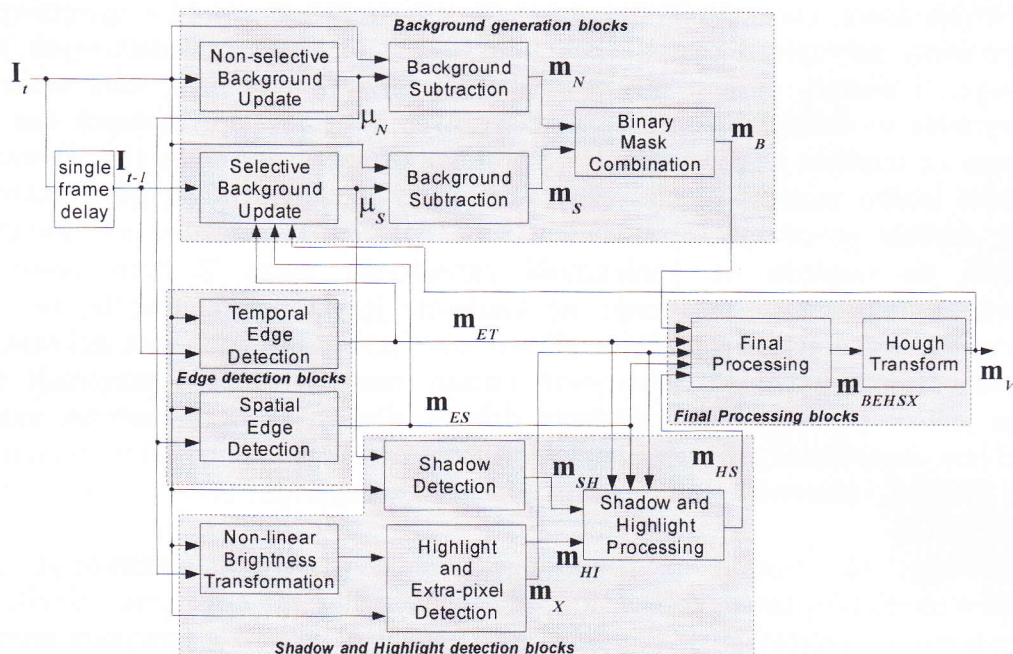


operacyjnych, które, chociaż oferują abstrakcję i wirtualizację zasobów sprzętowych oraz różne poziomy zabezpieczeń, to jednak odbywa się to kosztem dodatkowych zasobów sprzętowych i energii, co jest nie do zaakceptowania w systemach sieci sensorowych. Wykorzystanie w sieciach sensorowych klasycznych systemów operacyjnych jest również utrudnione ze względu na niestandardowe protokoły transmisji danych [1]. W węźle sieci sensorowej trudno zastosować klasyczny warstwowy model ISO/OSI, gdyż poszczególne warstwy modelu przenikają się i istnieją silne połączenia pomiędzy nie sąsiadującymi warstwami ze względu na konieczność zarządzania mocą. Z tych powodów do oprogramowania sieci często stosuje się klasyczne języki programowania, bez systemu operacyjnego, wykorzystując program główny oraz przerwania. Schemat działania węzła, który jednocześnie powinien odczytywać sensory oraz nadzorować transmisje radiowe, wymaga jednak pewnego zrównoleglenia działań, dlatego oprócz przerwań, stosuje się techniki tzw. *event-based programming* i *split-phase programming*, na bazie których powstał system TinyOS, stanowiący przykład systemu operacyjnego przeznaczonego do sieci sensorowej [12].

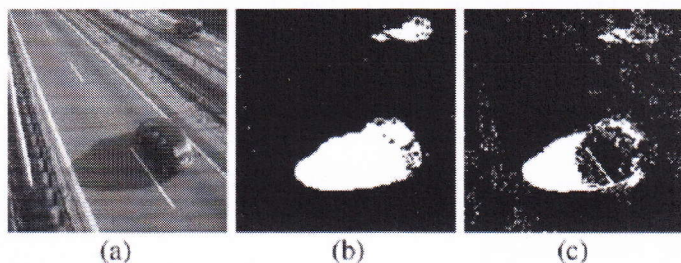
Podsumowując, ze względu na liczne ograniczenia (koszty, miniaturyzacja, zasilanie bateryjne o niewielkiej mocy, konieczność zapewnienia długiego czasu pracy sieci), systemy sieci sensorowych projektuje się pod konkretną aplikację, wszystkie powyższe problemy są rozwiązywane w wyniku licznych kompromisów i optymalizacji. Dotychczas nie powstał jeden dominujący standard projektowania i wykonania sieci sensorowej, nawet kryteria jakości działania sieci (*Quality of Service*) znacznie się różnią od metryk stosowanych w klasycznych sieciach komputerowych i telekomunikacyjnych, dlatego doświadczenie projektanta i szeroka wiedza z zakresu budowy sprzętu, oprogramowania i protokołów transmisyjnych są bardzo ważne.

## **2.2. Prace badawcze nad algorytmami przetwarzania obrazów i ich sprzętową realizacją**

Sieć sensorowa wyposażona w kamerę wymaga zastosowania lokalnego przetwarzania obrazów i przesyłania kanałem radiowym wyłącznie przetworzonych danych, gdyż transmisja obrazu wideo (nawet po kompresji) przez całą sieć sensorową jest nieefektywna energetycznie i potrzebuje bardzo dużego pasma radiowego oraz mocy z zasilania. Z tego powodu istotne jest, aby węzeł sieci sensorowej potrafił przeanalizować obraz, a transmisji radiowej podlegałyby tylko wybrane informacje wywnioskowane z tego obrazu. W ramach opracowanej przez habilitanta koncepcji, w pracy [P1] zainicjowanej przez habilitanta, powstała sprzętowa realizacja algorytmu odejmowania tła i detekcji obiektów pierwszoplanowych z monochromatycznego strumienia wideo pochodzącego ze stacjonarnej kamery. Wykorzystanie strumienia monochromatycznego zostało spowodowane koniecznością minimalizacji zużycia zasobów sprzętowych oraz poboru mocy, co jak pokazano w rozdziale 2.1 niniejszego autoreferatu, jest bardzo istotne w sieciach sensorowych. Do detekcji obiektów habilitant zaproponował wykorzystanie parametrycznego modelu tła o kształcie pojedynczego rozkładu normalnego, którego parametry są otrzymywane dzięki wykorzystaniu metody biegnącej średniej, przy czym w ten sposób zostały obliczone zarówno wartość średnia  $\mu$  jak i wariancja  $\sigma^2$  [P1]. Metoda odejmowania tła wymaga ciągłej analizy strumienia wideo. W wyniku badań symulacyjnych otrzymano schemat blokowy algorytmu, w którym zastosowano szereg operacji mających na celu zwiększenie dokładności działania algorytmu (rys. 2), takich jak np. wykrywanie cieni oraz podświetleń, które stanowią istotne źródło błędów w detekcji (rys. 3). Alternatywna realizacja wykrywania cieni z wykorzystaniem przestrzeni barw  $YC_bC_r$  została dodatkowo przedstawiona w [K1].



Rys. 2. Schemat blokowy algorytmu odejmowania tła [P1]



Rys. 3. Przykład detekcji cieni; (a) – obraz nieprzetworzony, (b) – wynik detekcji obiektów pierwszoplanowych za pomocą odejmowania tła bez detekcji cieni, (c) – cienie wykryte przez moduł detekcji cieni [P1]

Ocenę działania algorytmu przeprowadzono na podstawie testowej sekwencji wideo z ręcznie wyodrębnionymi obiektami pierwszoplanowymi (rys. 4), wykorzystując parametry krzywej ROC (*receiver operating curve*), takie jak *PR* (*precision*) oraz *RE* (*recall*):

$$PR = \frac{TP}{TP + FP} \quad (4)$$

$$RE = \frac{TP}{TP + FN} \quad (5)$$

gdzie:

*TP* – (*true positive*) liczba poprawnie rozpoznanych pikseli w obiekcie pierwszoplanowym;  
*FN* – (*false negative*) liczba pikseli, dla których błędnie nie wykryto obiektu pierwszoplanowego;

*FP* – (*false positive*) liczba pikseli, dla których błędnie wykryto obiekt.

Do określania jakości detekcji można także stosować współczynnik *miss rate* określony wzorem:

$$miss\ rate = \frac{FN}{TP + FN} = 1 - RE \quad (6)$$

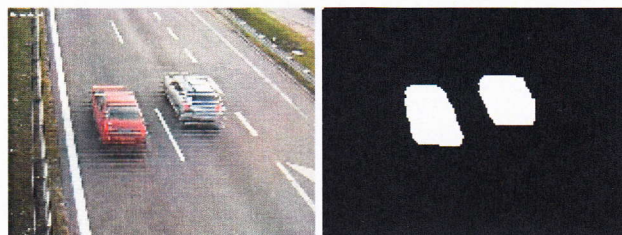
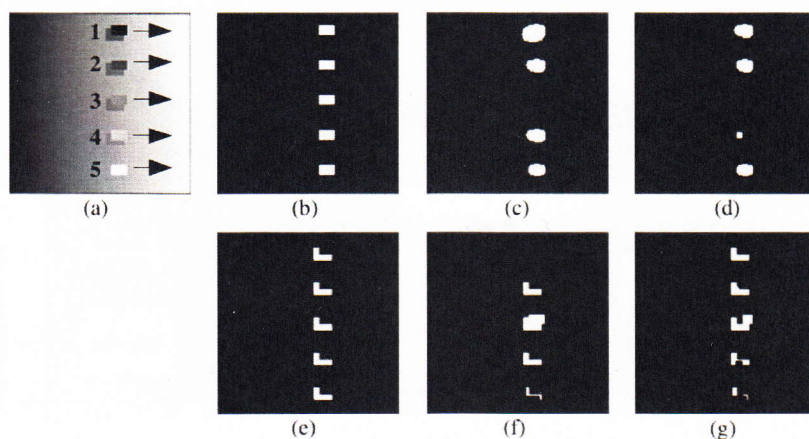


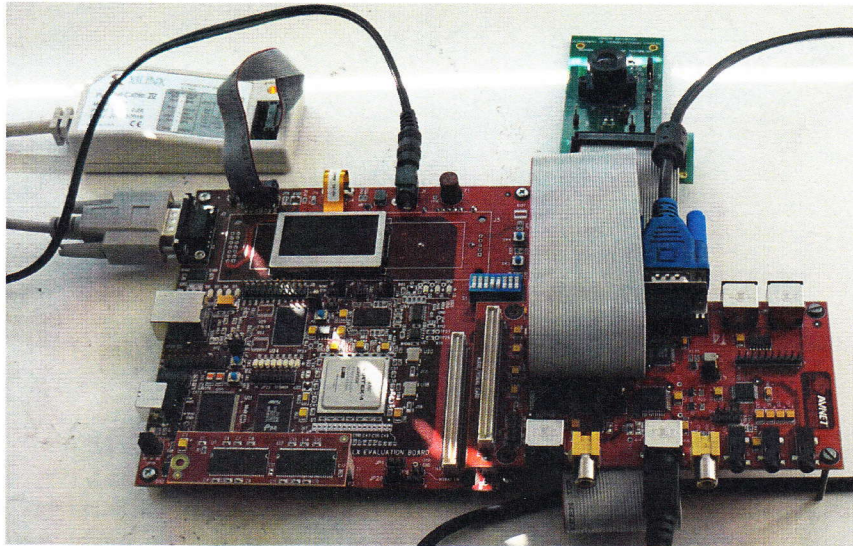
Fig. 4. Przykładowa ramka obrazu oraz ręcznie uzyskany obraz *ground truth*

W pracy [P1] habilitant przeprowadził także analizę działania algorytmu wykorzystując sztucznie wygenerowane obrazy zawierające różne kombinacje jasności poruszających się obiektów (prostokąty na rys. 5) oraz tła wraz z modelowaniem cieni.

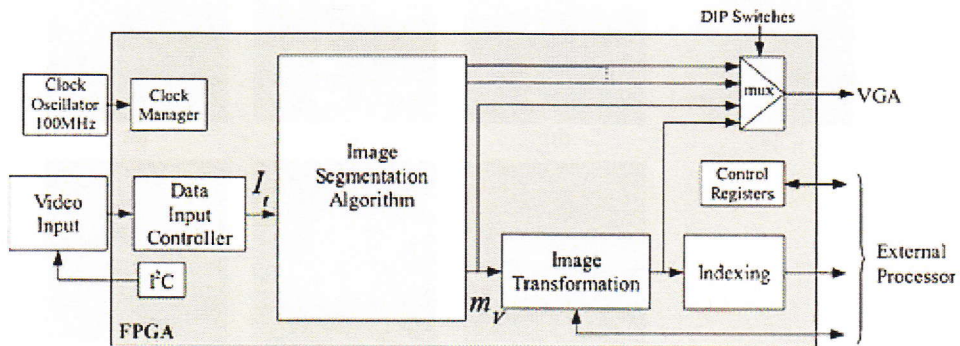


Rys. 5. Porównanie działania algorytmu dla 4- i 8-bitowej reprezentacji danych obrazu; (a) – sztucznie wygenerowany obraz zawierający poruszające się obiekty i ich cienie; (b) – *ground truth* dla obiektów; (c) – wynik detekcji obiektów dla 8-bitowej reprezentacji danych; (d) – wynik detekcji obiektów dla 4-bitowej reprezentacji danych; (e) – *ground truth* dla cieni; (f) – wynik detekcji cieni przy 8-bitowej reprezentacji danych; (g) – wynik detekcji cieni przy 4-bitowej reprezentacji danych [P1]

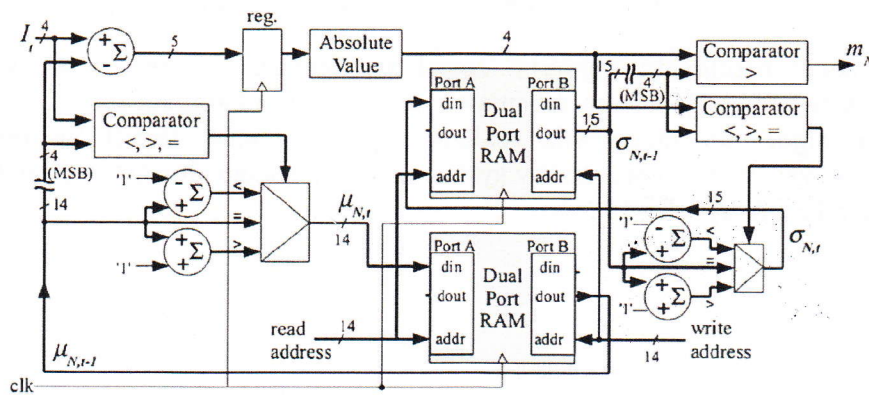
Pod kierunkiem habilitanta, algorytm został zrealizowany sprzętowo w układzie FPGA Virtex-4 SX (rys. 6), zgodnie ze schematem blokowym pokazanym na rys. 7, odpowiednia konstrukcja algorytmu umożliwiła realizację przetwarzania danych w postaci potoku (*pipeline*). Do zapamiętywania stanów systemu, wykorzystano wewnętrzne rejestry i pamięci RAM układu FPGA (rys. 8 i 9).



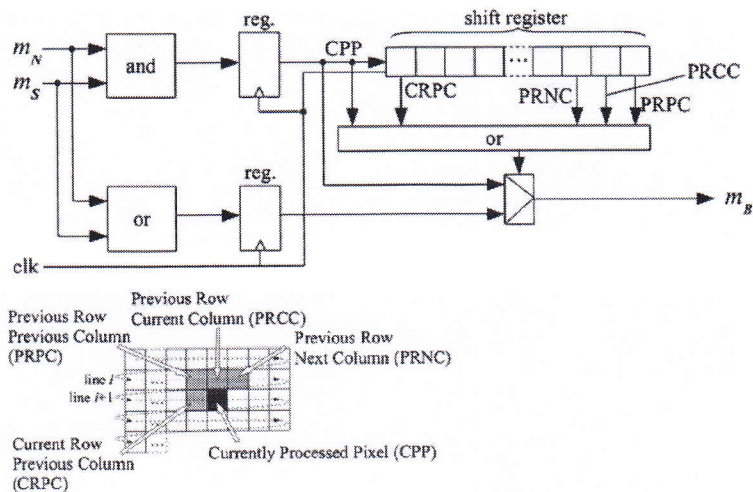
Rys. 6. Sprzęt służący do realizacji algorytmu detekcji obiektów ruchomych – płytka prototypowa Virtex-4 Evaluation Kit ADS-XLX-V4-SX-EVL35-12-G z rozszerzeniem ADS-AV-DAU oraz modulem kamery wykorzystującej przetwornik wideo MT9V111



Rys. 7. Ogólny schemat blokowy realizacji algorytmu w układzie FPGA [P1]

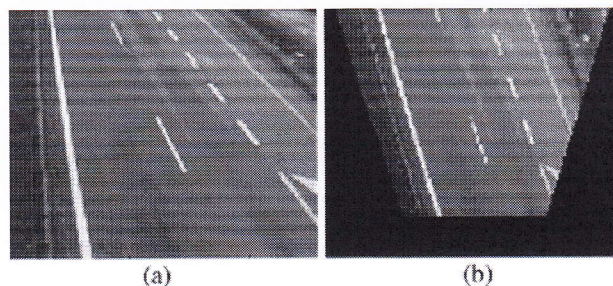


Rys. 8. Schemat blokowy sprzętowej realizacji bloku odejmowania tła [P1]



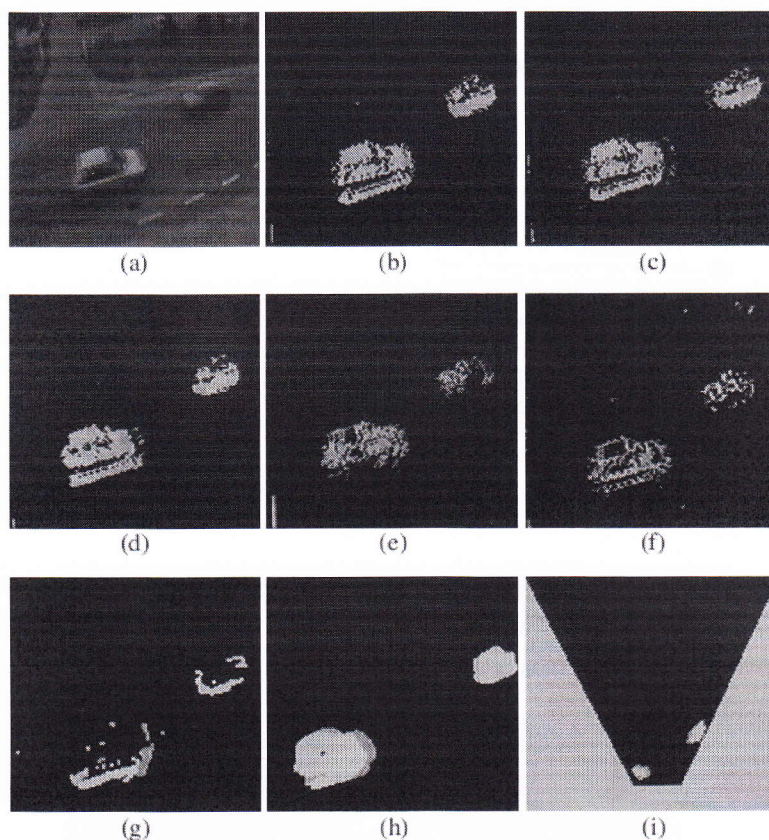
Rys. 9. Schemat blokowy potokowego obliczania maski obrazu [P1]

Dla zapewnienia liniowych zależności pomiędzy współzrędnymi obrazu a rzeczywistymi obiektami, w celu korekty perspektywy (rys. 10), zastosowano odpowiednią transformację geometryczną, również realizowaną sprzętowo.



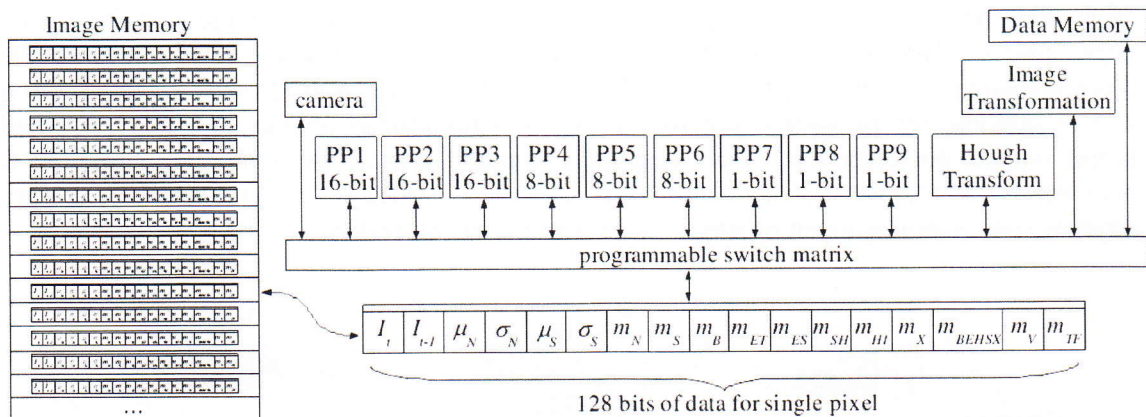
Rys. 10. Przykład działania korekcji perspektywy; (a) – obraz przed korekcją; (b) – obraz po korekcji [P1]

Obraz maski obiektów pierwszoplanowych otrzymany w wyniku działania algorytmu odejmowania tła jest trudny w analizie, ponieważ posiada wiele postrzępionych plam. Zastosowanie operacji morfologicznych na obrazie nie daje w tym przypadku zadawalających efektów. Dlatego też jedną z najważniejszych innowacji zaproponowanych przez habilitanta w bloku ostatecznego przetwarzania wyników algorytmu jest wykorzystanie transformacji Hough [13] z prostokątnym elementem strukturalnym. Ostateczne wyniki działania algorytmu pokazano na rys. 11.



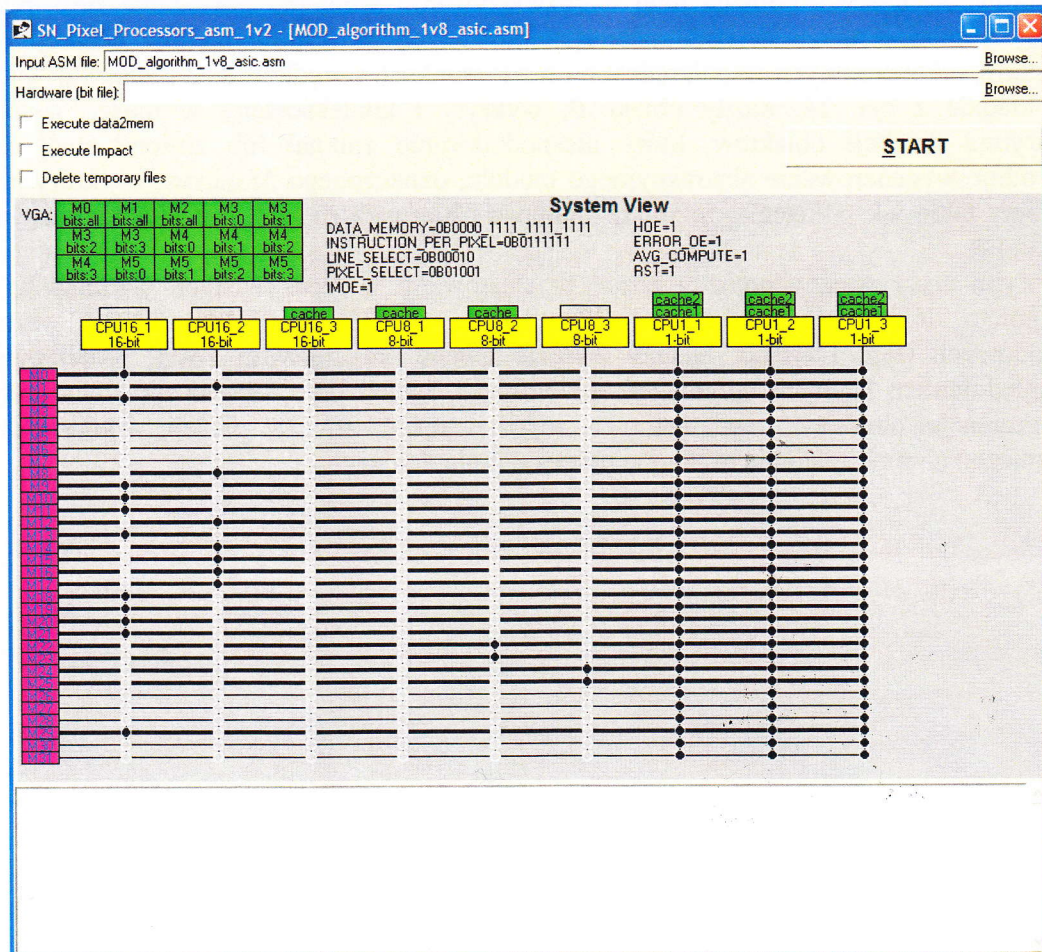
Rys. 11. Fotografia pokazująca działanie zrealizowanego sprzętowego algorytmu detekcji obiektów; (a) – obraz wejściowy z kamery; (b), (c) – wyniki nieselektywnego oraz selektywnego odejmowania tła; (d) – maska łączna pochodząca z bloków odejmowania tła; (e), (f) – maski wykrywania krawędzi w dziedzinie czasu i przestrzeni; (g) – maska pokazująca wykryte cienie i podświetlenia; (h) – maska po ostatecznym przetworzeniu i transformacji Hough; (i) – maska po przeprowadzeniu geometrycznej korekty perspektywy [P1]

Pod kierunkiem habilitanta, w pracy [P2] i [P3] w ramach projektu [G5], w którym habilitant był głównym wykonawcą, przedstawiono zrealizowaną sieć sensorową, której węzły sieci sensorowej zostały wyposażone w kamery. Strumień wideo z kamery był przetwarzany za pomocą opisanego powyżej algorytmu wyodrębniania obiektów pierwszoplanowych, przy czym realizacja sprzętowa algorytmu została wykonana za pomocą wieloprocesorowego podsystemu obróbki obrazu. Habilitant zaproponował rozwiązanie polegające na zbudowaniu zespołu heterogenicznych procesorów do przetwarzania danych obrazu, gdzie pojedyncze procesory, zwane *pixel processor* (PP), w sposób równoległy realizowały wybrane operacje dotyczące obróbki aktualnie przetwarzanego piksela obrazu. W wyniku prac badawczo-rozwojowych opracowano sparametryzowaną architekturę pojedynczego PP, gdzie parametrem była liczba bitów przetwarzanych danych przez procesor. Następnie zrealizowano zestaw procesorów PP o różnej liczbie bitów i przypisano im odpowiednie bloki algorytmu, np. procesory 16-bitowe były odpowiedzialne za realizację odejmowania tła, procesory 8-bitowe realizowały wykrywanie cieni i podświetleń oraz indeksowanie wykrytych obiektów, natomiast procesory 1-bitowe zostały wykorzystane do realizacji operacji morfologicznych na jednobitowych maskach obrazu. Strukturę blokową tak zrealizowanego sprzętu przeznaczonego do wykonywania algorytmu detekcji obiektów przedstawiono na rys. 12.



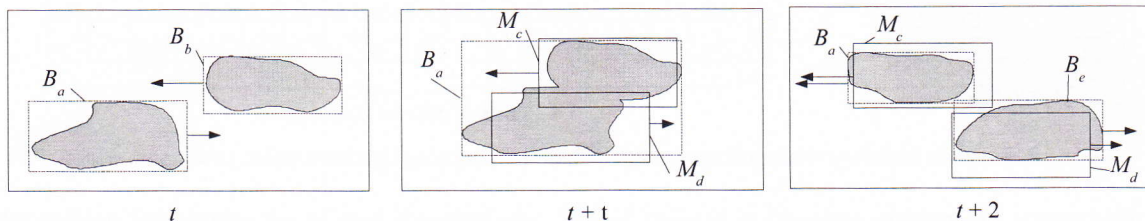
Rys. 12. Schemat blokowy wieloprocessorowego układu realizującego przetwarzanie strumienia wideo [P2]

Procesory PP zostały opisane w języku VHDL, dodatkowo habilitant opracował środowisko projektowe wraz z assemblerem dedykowanym procesorom PP, którego graficzny interfejs użytkownika GUI (*graphical user interface*) pokazano na rys. 13. Procesory zostały połączone poprzez przełączalną maskowalną dedykowaną magistralę danych (*programmable switch matrix*), dzięki temu mogą jednocześnie przetwarzać dane obrazu oraz stan modelu tła zapamiętany dla każdego piksela.



Rys. 13. Widok GUI opracowanego środowiska projektowego dedykowanego dla procesorów PP

W wyniku działania algorytmu otrzymuje się maskę binarną, zawierającą jednorodne obszary (tzw. plamy, *blobs*) odpowiadające wykrytym obiektom. Maskę tę zostaje poddana indeksowaniu przez jeden z procesorów PP. W wyniku indeksowania, każdy piksel otrzymuje własną etykietę informującą o przynależności do konkretnego obszaru, a na koniec każdej ramki obrazu dostępna jest tabela mapująca etykiety pikseli na numer obiektu oraz ich współrzędne i rozmiary prostokątów opisujących te obszary. Dzięki temu można łatwo uzyskać listę współrzędnych prostokątów reprezentujących wykryte obiekty.




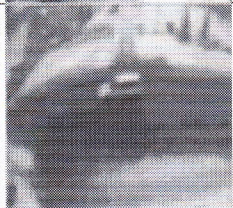

Rys. 14. Przykład wysokopoziomowego śledzenia obiektów dla trzech kolejnych ramek obrazu [P2]

Dane z tej listy zostają następnie poddane przetwarzaniu z użyciem typowego procesora 32-bitowego (jest to, obok 9 procesorów PP, dziesiąty, największy, procesor w jednokładowym systemie węzła sieci sensorowej), który za pomocą zaproponowanych przez habilitanta heurystycznych reguł śledzenia obiektów [P2], określa kierunki ruchu i prędkości wykrytych obiektów. Heurystyka pomaga tutaj w rozwiązywaniu problemu nakładania się i sklejania plam ze sobą, a także łączenia zdefragmentowanych w wyniku błędów detekcji obrazu obiektów o koherentnych współrzędnych oraz kierunkach i prędkościach ruchu. Jak widać na przykładzie z rys. 14, każdy obiekt  $B$ , wykryty i zindeksowany w masce wynikowej algorytmu detekcji obiektów, który niespodziewanie zniknął lub zmienił swój kształt, powoduje wygenerowanie abstrakcyjnego modelu, oznaczonego  $M$  na rys. 14, który posiada atrybuty takie jak położenie, rozmiar, prędkość i kierunek ruchu i jest niejako kontynuacją obiektu  $B$ .

Algorytm wykrywający pojazdy został przetestowany w rzeczywistych warunkach, gdzie osiągnięto dokładność detekcji pojazdów rzędu 63-93%, w zależności od warunków pogodowych (tab. 1, [P2]). Należy zwrócić uwagę, że algorytm został projektowany z uwzględnieniem licznych ograniczeń sprzętowych, takich jak 4-bitowa monochromatyczna reprezentacja obrazów, czy niewielka rozdzielczość 128x128 pikseli, wynikających z ograniczeń na moc zasilania oraz moc przetwarzania danych.



Tabela 1. Wyniki pomiarów detekcji pojazdów przez zrealizowany system w warunkach rzeczywistych [P2]

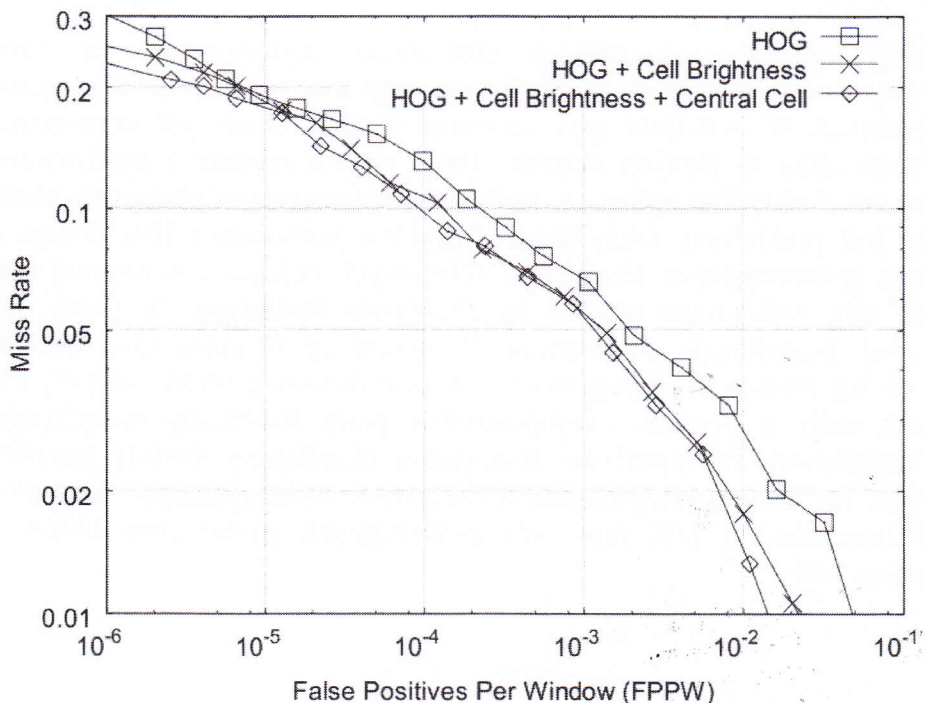
Scene	Frame sample	% of vehicles detected correctly by the sensor network node	% of vehicles falsely overdetected by the sensor network node
Day, strong sun		93	8
Cloudy day		83	9
Night		63	1

Do detekcji obiektów w obrazie statycznym wykorzystuje się również cechy charakterystyczne wykrywanych obiektów, stosując dwa różne podejścia: *scanning window* oraz *parts-based*. W podejściu typu *scanning window* obraz jest skanowany za pomocą przesuwanego okna w różnych skalach. Dane obrazu zebrane z analizowanego okna są zamieniane na deskryptor opisujący cechy charakterystyczne obrazu w oknie. Następnie deskryptor jest poddawany klasyfikacji. Podejście *parts-based* [14] polega na osobnym wykrywaniu poszczególnych elementów składowych obiektu i zazwyczaj działa lepiej w przypadku, gdy wykrywane obiekty są częściowo zasłaniane. W pracy [P6] habilitant zaproponował modyfikację deskryptora *Histogram of Oriented Gradients* (HOG) [15], pracującego wg metody *scanning window* z klasyfikatorem SVM [16][17] i służącego do wykrywania osób w obrazie. Zaproponowana przez habilitanta modyfikacja polega na dodaniu dodatkowej informacji do deskryptora określającej średnią jasność komórki w deskrytorze. Do uczenia klasyfikatora SVM zostało wykorzystanych 2416 obrazów z bazy INRIA *Person dataset* [18] (rys. 15) zawierających osoby oraz 24360 obrazów nie zawierających osób.



Rys. 15. Przykładowe obrazy testowe zawierające osoby (górnny rząd) oraz nie zawierające osób (dolny rząd). Zdjęcia pochodzą z bazy INRIA *Person dataset* [18]

W wyniku analizy otrzymano krzywą *False Positive Per Window* FPPW, zawierającą te same informacje, co krzywe ROC i *recall-precision* (RP) opisane równaniami (4) i (5), ale jej kształt pozwala na dokładniejszą obserwację jakości działania detektora. Na rys. 16 przedstawiono wyniki pomiarów działania deskryptora (z klasyfikatorem SVM) dla jego różnych wersji: krzywa opisana jako "HOG" przedstawia wynik działania klasycznego deskryptora HOG opisanego w [15], natomiast krzywa "HOG + *Cell Brightness*" przedstawia wyniki działania deskryptora zawierającego modyfikację zaproponowaną przez habilitanta w [P6].

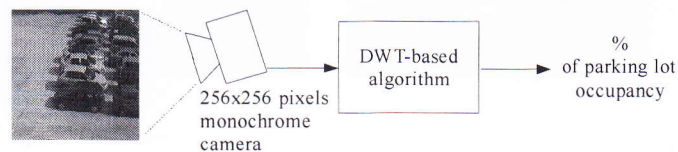


Rys. 16. Krzywa *Detection Error Tradeoff* przedstawiająca wyniki klasyfikacji obrazów z bazy INRIA *Person dataset* [18] dla trzech deskryptorów opisanych w tekście za pomocą klasyfikatora SVM [P6]

Do obliczenia deskryptora HOG wykorzystuje się histogram gradientów jasności sąsiednich pikseli w analizowanym oknie, podzielonym na niewielkie, nachodzące na siebie prostokątne komórki o wymiarach 8x8 pikseli. Obliczanie histogramu dla każdej komórki jest poprzedzone nałożeniem na obraz dwuwymiarowej maski o rozkładzie gaussowskim, której środek pokrywa się ze środkiem komórki. W ten sposób piksele występujące w środku

komórki otrzymują większą wagę niż piksele brzegowe. Z powodu nakładania się komórek, taka maska musi być wielokrotnie nakładana na piksele każdej komórki, co sprawia, że potokowa realizacja sprzętowa takiej operacji jest skomplikowana. Z tego powodu habilitant zastosował zaproponowaną przez siebie modyfikację dla metody HOG wykorzystującą dodatkową komórkę środkową zamiast maski gaussowskiej [19]. Dzięki temu możliwa jest łatwiejsza sprzętowa realizacja procesu obliczania deskryptora. Jak widać z rys. 16, taki deskryptor, oznaczony jako "HOG + Cell Brightness + Central Cell", charakteryzuje się jeszcze lepszymi wynikami detekcji. Dzięki zaproponowanej modyfikacji deskryptora zaobserwowano 25% poprawę współczynnika *miss rate* opisanego równaniem (6) dla wartości  $10^{-4}$  FPPW, co jest równoważne trzykrotnie lepszym współczynnikiem FPPW dla tych samych wartości *miss rate*.

Inne podejście do rozpoznawania obrazu w sieci sensorowej zostało pokazane w pracy [P5], gdzie obraz ze stacjonarnej kamery obserwującej parking dla pojazdów jest przetwarzany przez pojedynczy procesor, a w wyniku analizy obrazu uzyskano procentową zajętość parkingu przez stojące pojazdy. W rozwiązaniu tym wykorzystano dwuwymiarową separowalną dyskretną transformację falkową (2-D DWT) z wykorzystaniem falki Daubechies [20] o 4 współczynnikach.



Rys. 17. Węzeł sieci sensorowej do obserwacji zajętości parkingu

Po transformacji, 16 obrazów reprezentujących komponenty w dziedzinie częstotliwości jest poddawanych liniowej kombinacji wartości bezwzględnych punktów o tych samych współrzędnych zgodnie ze wzorem:

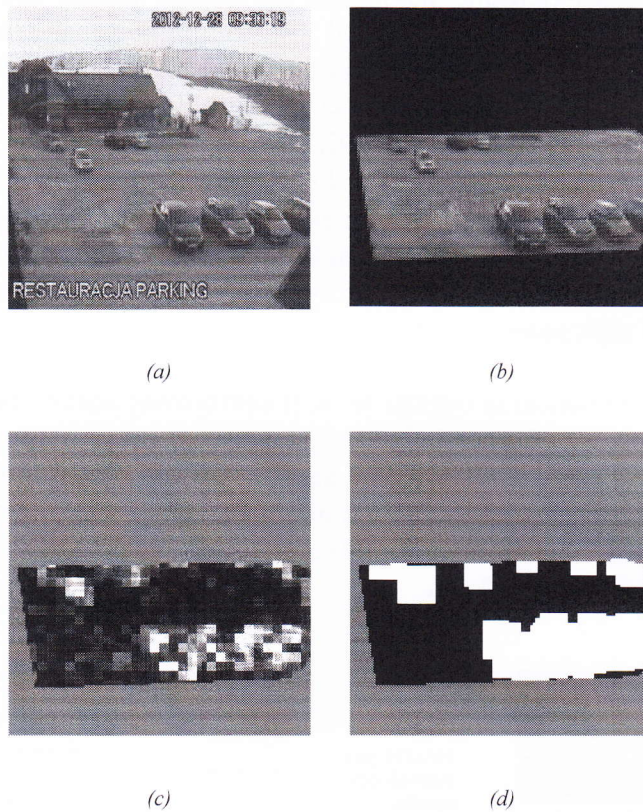
$$r_{xy} = \sum_{i=1}^4 \sum_{j=1}^4 c_{ij} |p_{ijxy} - 128| \quad (7)$$

gdzie:

$r_{xy}$  – jasność piksela o współrzędnych  $(x, y)$  obrazu wynikowego po kombinacji liniowej;  
 $c_{ij}$  – współczynnik  $(i, j)$  macierzy  $\mathbf{C}$  określającej wagi kombinacji liniowej (sposób obliczenia macierzy  $\mathbf{C}$  przedstawiono poniżej);

$p_{ijxy}$  – jasność piksela o współrzędnych  $(x, y)$  z komponentu  $(i, j)$  pochodzącego z obrazu wynikowego transformacji 2-D DWT.

Tak otrzymana macierz punktów  $r_{xy}$  podlega progowaniu za pomocą stałego współczynnika  $\tau$ , a następnie operacjom morfologicznym (zamknięcie i dylatacja morfologiczna z elementem strukturalnym  $3 \times 3$ ). Ostatecznie otrzymujemy maskę zajętości parkingu, z której można łatwo obliczyć procentową zajętość powierzchni parkingu przez pojazdy.



Rys. 18. Przykład działania algorytmu wykorzystującego transformację 2D-DWT do analizy zajętości parkingu; (a) – obraz wejściowy, (b) – obraz wejściowy z zamaskowanym obszarem poza parkingiem, (c) – wartości współczynników  $r_{xy}$ , gdzie kolor czarny to wartość 0, a biały to wartość 255, (d) – wynik progowania oraz operacji morfologicznych na obrazie  $r_{xy}$  [P5]

Macierz współczynników  $\mathbf{C}$  została obliczona przez habilitanta za pomocą procesu uczenia nadzorowanego i algorytmu genetycznego poszukującego najmniejszego błędu detekcji  $\varepsilon$  opisanego równaniem (8), na podstawie przykładowych obrazów wejściowych oraz masek *ground-truth*, uzyskanych poprzez ręczną indeksację:

$$\varepsilon(\mathbf{C}, n_c, n_d, \tau) = \sum_{i=1}^N (S_i(\mathbf{C}, n_c, n_d, \tau) - G_i)^2 \quad (8)$$

gdzie:

- $S_i()$  – liczba pikseli wykrytych w obrazie  $i$  przez algorytm;
- $G_i$  – liczba pikseli w obrazie  $i$  pochodzącym z bazy *ground-truth*;
- $N$  – liczba obrazów uczących.

Jak wynika ze wzoru (8), w chromosomie algorytmu genetycznego zakodowano następujące informacje:

- $n_c$  – liczba operacji zamknięcia morfologicznego;
- $n_d$  – liczba operacji dylatacji morfologicznej;
- $\tau$  – wartość progu przy operacji progowania współczynników  $r_{xy}$ ;
- współczynniki macierzy  $\mathbf{C}$ .

Zaproponowana metoda wykorzystująca transformację falkową nie wymaga ciągłej analizy strumienia wideo, przez co algorytm bardzo dobrze nadaje się dla sieci sensorowych. Węzeł sieci sensorowej może w sposób okresowy włączać kamerę na krótki czas w celu zarejestrowania ramki obrazu, a następnie procesor może wykonać opisany powyżej algorytm

w dowolnym czasie. Z powodu braku nieprzekraczalnej granicy czasowej dla realizacji obliczeń, nie ma tutaj potrzeby sprzętowej akceleracji obliczeń, a do uruchomienia i wykonania algorytmu wystarczy pojedynczy procesor, realizujący także wszystkie inne funkcje węzła sieci.

### **2.3. Prace badawcze nad wybranymi aspektami transmisji radiowej w sieciach sensorowych**

W rozdziale 2.1 niniejszego autoreferatu przedstawiono wpływ transmisji radiowej na zużycie energii w sieci sensorowej. Wspomniano także, że sieci sensorowe, ze względu na liczne kompromisy wynikające z ograniczenia na rozmiary węzła sieci i koszt sprzętu (w tym transceivera radiowego) oraz bateryjne zasilanie, są zazwyczaj projektowane pod konkretną aplikację, aby osiągnąć jak najwięcej efektów wynikających z optymalnego wykorzystania sprzętu i baterii. Z tego powodu wykorzystanie typowych protokołów sieciowych [21] do transmisji radiowej jest nieefektywne ze względu na narzuty protokołu powodujące straty cennej mocy z bateryjnego źródła zasilania. Fakt czasowego usypiania transceivera całkowicie eliminuje tu standardy sieci przewodowych, także standardy z grupy 802.11 nie mają zastosowania w przypadku niskomocowych sieci sensorowych z zasilaniem baterijnym. Standard 802.15.4 w podstawowym wariantcie wymaga koordynatora (*full-function device*, FFD) z dostępem do sieci zasilającej, natomiast wersja typu *peer-to-peer* tego standardu nie definiuje warstwy sieciowej oraz wymaga transmisji nadmiarowych danych sterujących. Z tych powodów bardzo często projektując sieć, implementuje się własny protokół, w maksymalnym stopniu dopasowany do danej aplikacji. W pracy [P2] habilitant zaproponował autorski protokół transmisyjny. Założono, że sieć sensorowa będzie wyposażona w podukład monitoringu wizyjnego działający w trybie ciągłym, który, pomimo sprzętowej realizacji, posiadał znaczne zapotrzebowanie na moc zasilania. W celu osiągnięcia maksymalnie długiego czasu działania węzłów sieci, założono możliwość pozyskiwania energii ze środowiska za pomocą baterii słonecznych. Tym niemniej, aby gabaryty paneli fotowoltaicznych posiadały praktyczne rozmiary, należało dążyć do minimalizacji poboru mocy z zasilania na każdym etapie projektowania węzła sieci.

Do transmisji radiowej pomiędzy węzłami sieci, autor zaproponował wykorzystanie jednokanałowego transceivera korzystającego z bezlicencyjnego pasma ISM (*Industrial, Scientific, Medical*) o częstotliwości środkowej 868MHz i przepustowości 38.4 kbps z kodowaniem Manchester. Ze względu na dostępność tylko jednego kanału radiowego, opracowany przez habilitanta protokół wykorzystuje metodę TDMA (*Time Division Multiple Access*) dostępu do medium transmisyjnego MAC (*Medium Access Control*) w warstwie łącza danych. Do każdego węzła przypisano unikalny adres składający się z adresu podsieci oraz numeru węzła w podsieci, dzięki czemu możliwa jest praca wielu sieci sensorowych w bezpośrednim sąsiedztwie.

Jak przedstawiono w [P2], zgodnie z koncepcją habilitanta, każdy węzeł sieci działa z takim samym okresem  $T_i$  planu transmisji, wykorzystując do tego celu lokalny generator zegarowy. Na podstawie analizy nasłuchu transmisji z bezpośredniego sąsiedztwa, węzeł wybiera swój czas rozpoczęcia transmisji  $t_{tx}$  ulokowany w planie transmisji o okresie  $T_i$  tak, aby zminimalizować możliwość kolizji z innymi transmisjami. Jak wiadomo, każda kolizja powoduje konieczność retransmisji danych, co pociąga za sobą straty energii zasilania. Wybór czasu  $t_{tx}$  odbywa się zgodnie z zależnością:

$$\forall_i |t_{tx,i} - t_{tx}| > T_{MIN} \quad (9)$$

gdzie  $t_{tx,i}$  to początek transmisji sąsiedniego węzła  $i$ , a  $T_{MIN}$  to zdefiniowana globalnie dla całej sieci stała zapewniająca margines czasowy pomiędzy transmisjami. Czasy  $t_{tx,i}$  oraz  $t_{tx}$  są określane względem lokalnego źródła czasu każdego węzła.

Habilitant zaproponował, aby każdy węzeł był wyposażony w lokalną bazę danych zawierającą informację o sąsiednich węzłach na podstawie okresowego nasłuchu radiowego. Baza ta zawiera informację o sąsiednich węzłach, z którymi dany węzeł posiada dwukierunkową transmisję, potwierdzoną za pomocą pakietów potwierdzających ACK (*acknowledge*). Dodatkowo w lokalnej bazie danych węzła znajdują się dane węzłów, które są poza zasięgiem transmisji, ale objęte są zasięgiem węzłów sąsiednich. Takie węzły zostały określone przez habilitanta jako niebezpośrednie węzły sąsiednie (*indirect neighbour*). Śledzenie obecności niebezpośrednich węzłów sąsiednich jest możliwe na podstawie, nasłuchu transmisji węzłów sąsiednich, gdyż w tych transmisjach zawarte są odpowiednie informacje. Rejestrowanie obecności niebezpośrednich węzłów sąsiednich jest istotne w celu uniknięcia problemu ukrytego terminalu (*hidden terminal problem*) oraz problemu oddalonego terminalu (*exposed terminal problem*) [22]. Możliwość nakładania się transmisji w czasie jest stale monitorowana; w opracowanym przez habilitanta protokole zapewniono również możliwość zasugerowania węzłom sąsiednim, powodującym te problemy, zmiany czasu transmisji za pomocą odpowiednich pakietów kontrolnych.

Ze względu na brak infrastruktury sieciowej oraz brak koordynatorów sieciowych (wg terminologii zgodnej ze standardem 802.15.4), organizacja sieci transmisyjnej odbywa się za pomocą regularnie wysyłanych sygnałów radiowych, tzw. *beacon*. Sygnały *beacon* są nadawane przez każdy węzeł w czasie  $t_{tx}$  z okresem  $N_B T_t$ , gdzie  $N_B$  ma wartość typowo od 3 do 10.

Po uruchomieniu zasilania, węzeł najpierw działa w trybie nasłuchu, w którym rejestruje informacje o bezpośrednich oraz niebezpośrednich sąsiadach. Tryb nasłuchu uruchamiany jest również regularnie w trakcie pracy sieci, w celu wykrycia zmian w topologii sieci oraz zarejestrowania ewentualnych nowych węzłów sąsiednich.

Do celów transmisji habilitant opracował dedykowaną strukturę pakietów pokazaną w tab. 2 oraz tab. 3.

Tabela 2. Struktura ramki transmisyjnej

Pole:	HEADER (64-bitowy)	PAYLOAD (0...1920 bitów)	CRC (32 bity)
Dane:	Rodzaj ramki, adres węzła nadającego i odbierającego, długość ramki, suma kontrolna CRC nagłówka.	Dane sygnału <i>beacon</i> , 128-bitowe pakiety z danymi (PKT).	Suma kontrolna CRC pakietów z danymi.

Tabela 3. Struktura 128-bitowego pakietu z danymi (PKT)

Pole:	PKT_ID (4-bitowy) (typ pakietu)	PKT_ADDR (12 bitów)	PKT_REAL TIME (32 bity)	PKT_DATA (64-bitowy) (zawartość zależy od wartości PKT_ID)	PKT_CRC (16 bitów)
Dane:	PKTID_DATA1 (podstawowe statystyki ruchu pojazdów)	Adres węzła nadającego dane	Znacznik czasu ( <i>time stamp</i> )	Liczba i średnia prędkość pojazdów wykrytych w ciągu ostatniej minuty dla każdego z czterech kierunków (N, E, W, S).	Suma kontrolna CRC pakietu z danymi.
	PKTID_DATA2 (dane o środowisku)			Temperatura otoczenia, nasłonecznienie, itp.	
	PKTID_DATA3 (parametry robocze węzła)			Stan baterii węzła, liczba wykrytych węzłów sąsiednich, adresy pierwszych 8 węzłów sąsiednich.	
	PKTID_DATA4 (dane obrazu z kamery)			64 bity z obrazu kamery.	

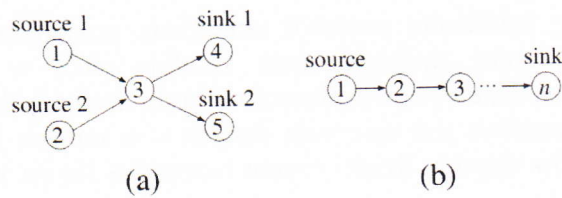
Zaprojektowany przez habilitanta protokół umożliwia autoorganizację sieci – węzły po zainstalowaniu i włączeniu zasilania same zbudują sieć; w przypadku uszkodzenia poszczególnych węzłów, dane będą wysyłane do innych, dostępnych i działających węzłów w okolicy. Dodatkowo możliwa jest agregacja danych – w ramach jednego nagłówka można nadawać wiele pakietów danych, dzięki czemu oszczędza się na wysyłaniu niepotrzebnych nagłówków.

Węzły regularnie nadają informacje o zebranych statystykach ruchu pojazdów z obserwowanych obszarów do sąsiednich węzłów. W rezultacie zaproponowanego przez habilitanta procesu samoorganizacji sieci, każdy węzeł posiada informacje, do którego następnego węzła dane mają być przekazywane. W ten sposób dane, po wielu retransmisjach (*hops*) pomiędzy kolejnymi parami węzłów, trafiają do węzłów oznaczonych w sieci jako odbiorcy danych (*sink*). Protokół również umożliwia węzłom przesyłanie zarejestrowanej klatki obrazu z kamery. Ze względu na niewielką przepustowość sieci, odbywa się to na żądanie zarządzającego siecią (rys. 19).

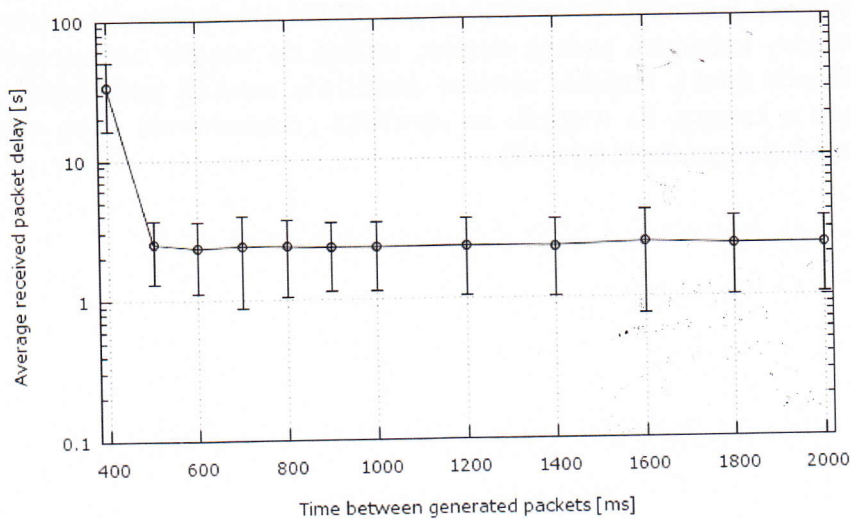


Rys. 19. Widok okna programu nadzorującego sieć sensorową, pokazujący obraz z kamery węzła, przesłany przez sieć

Opóźnienia transferu danych w sieci zostały przedstawione na rys. 21 i 22 dla dwóch przypadków pokazanych na rys. 20.

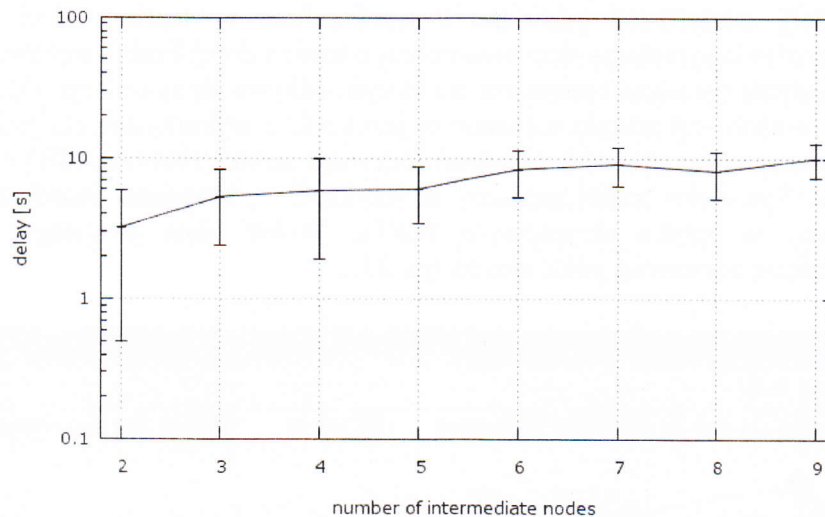


Rys. 20. Konfiguracje testowe rozmieszczenia węzłów sieci sensorowej wykorzystane do pomiarów transmisji danych



Rys. 21. Średnie opóźnienie pakietów danych dla konfiguracji sieci z rys. 20a





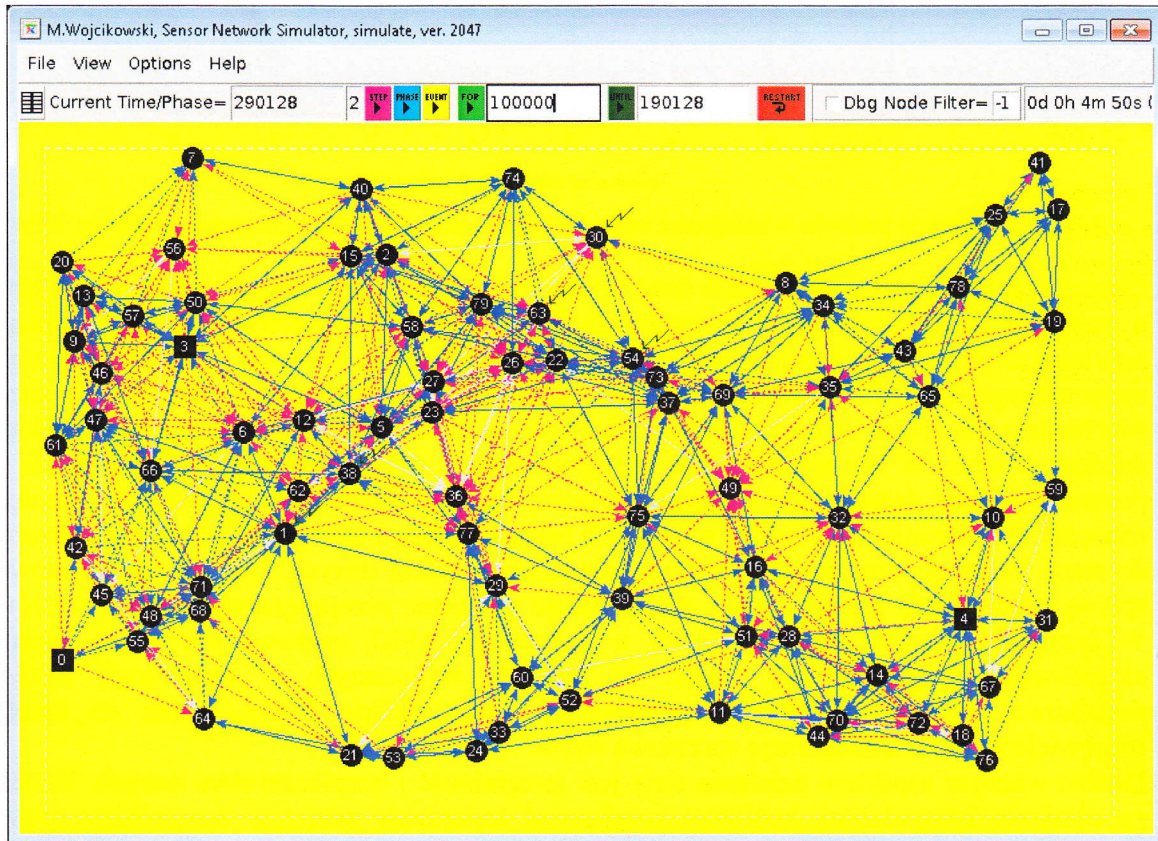
Rys. 22. Opóźnienie pakietów w zależności od liczby węzłów pośrednich dla konfiguracji sieci z rys. 20b

Oprócz regularnej pracy sieci, gdzie dane zebrane z czujników spływają do węzłów typu *sink*, a następnie są rejestrowane w komputerach bezpośrednio podłączonych do tych węzłów, habilitant zaproponował i zrealizował transmisję w przeciwnym kierunku – dane z komputera stacjonarnego typu PC, poprzez *sink*, są dostarczane do wszystkich węzłów w sieci. Mechanizm ten został wykorzystany do zdalnego uaktualniania oprogramowania wszystkich węzłów w sieci. W ten sposób nowe oprogramowanie, transportowane we fragmentach w postaci pakietów danych, jest zapisywane przez każdy z węzłów w wewnętrznej pamięci roboczej. Po zebraniu przez węzeł wszystkich pakietów, węzeł ten wysyła informację zwrotną do stacji bazowej potwierdzającą otrzymanie całego pakietu danych. Gdy do stacji bazowej spłyną odpowiednie potwierdzenia, stacja nadaje polecenie przeprogramowania się węzłów. W przypadku węzłów zrealizowanych w technologii ASIC, możliwa jest w ten sposób aktualizacja oprogramowania węzła. W węzłach wykonanych w technologii FPGA, dodatkowo możliwe jest zdalne przeprogramowanie układu FPGA, przez co można całkowicie zmienić konfigurację sprzętową węzła.

Bardzo ważnym aspektem działania sieci jest integralność i bezpieczeństwo danych. W celu zapewnienia integralności, habilitant zrealizował oznaczanie pakietów znacznikiem czasu, zaproponowany protokół sieciowy umożliwia globalną synchronizację czasową wszystkich węzłów, inicjowaną w regularnych odstępach czasowych przez węzły bazowe. Dane nadawane przez sieć sensorową są szyfrowane za pomocą sprzętowo realizowanego algorytmu AES (*Advanced Encryption Standard*) [23], każdy węzeł posiada zestaw unikalnych wielopoziomowych haseł, których podanie jest niezbędne do uzyskania określonych funkcjonalności. Szyfrowanie danych przez węzły zapobiega także wstrzykiwaniu obcych pakietów danych. Dostęp osoby postronnej do pojedynczego węzła nie wystarczy do przejścia całej sieci, gdyż hasła są unikalne dla każdego węzła.

Projektowanie protokołu sieci sensorowej jest złożonym procesem, dlatego bardzo cenna jest możliwość symulacji nowo projektowanego protokołu w celu wykrycia błędów. W pracy [P4] habilitant zaproponował i zrealizował unikalne środowisko do symulacji i emulacji transmisji danych w radiowej sieci sensorowej. Wykorzystanie typowych symulatorów sieciowych, np. ns-2 [24], ns-3 [25], SSFNet [26] czy GloMoSim [27] i wiele innych, obarczone jest szeregiem niedogodności, które opisano w [P4]. Z tego powodu, w opracowanym przez habilitanta środowisku, zastosowano emulację rzeczywistych węzłów sieci oraz symulację transmisji danych pomiędzy wirtualnymi transceiverami. Dzięki takiemu rozwiązaniu, oprogramowanie węzłów, które zostało poddane analizie w środowisku symulacyjnym, może

być bez żadnej modyfikacji przesłane i uruchomione w węzłach sieci sensorowych. Symulator umożliwia symulację sieci sensorowej o bardzo dużej liczbie węzłów, liczba ta jest ograniczona jedynie pamięcią komputera, na którym odbywa się symulacja. Oprogramowanie węzłów sieci sensorowej zostało napisane w języku C z wykorzystaniem jednowątkowego programu głównego oraz procedur przerwań (*interrupt service routines, ISR*) oraz częściowo w asemblerze. Symulator został napisany w języku C++, natomiast interfejs użytkownika został napisany w języku skryptowym Tcl/Tk. Widok okna głównego symulatora z przykładową siecią sensorową pokazano na rys. 23.



Rys. 23. Widok interfejsu GUI środowiska do symulacji sieci sensorowych

Symulator obrazuje w sposób graficzny topologię sieci wraz z zaznaczeniem osiągniętych transmisji. Symulator może generować stochastyczne błędy transmisji przy założeniu stałej stopy błędów. W prosty sposób możliwe jest dodanie nowych modeli, np. dwustanowego modelu kanału Gilbert-Elliot'a.

W każdej chwili można zatrzymać symulację i przeprowadzić analizę wewnętrznych stanów poszczególnych węzłów sieci. Na rys. 24 pokazano widok bazy danych zawierającej informacje o węzłach sąsiednich dla przykładowego węzła. Rys. 25 przedstawia zbiorczą tabelę stanów wewnętrznych grupy węzłów.

Node 15 hops=1, TTb=(4385)DISC=(1210/49) Global Tx Time=(4385)											
ADR	TYP	TX	TTB	DIS	HOP	LEN	MIS	LTX	LRX	STA	GLD
5	Hear	8511	10	-1	255	-1	12	0	107	-1	(9260)
17	Hear	6706	0	-1	255	-1	12	1	26	-1	(7455)
13	Hear	6602	10	-1	255	-1	12	0	107	-1	(7351)
18	Norm	4841	9	96	0	-1	116	9	299	-1	(5590)
6	Norm	3373	3	10	255	-1	12	9	300	-1	(4122)
2	Norm	1692	3	20	255	-1	12	9	300	-1	(2441)
1	Norm	1547	3	0	255	-1	12	9	300	-1	(2296)

Rys. 24. Przykładowy widok bazy danych węzła w środowisku do symulacji sieci sensorowych

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Battery	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ClgBuffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ClgBufferLen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ClgBufferSeq	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ClgBufferTTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DataBuffer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DataBufferLen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DataBufferSeq	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DataBufferTTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
DataCollectTime	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	
DiscoverEndTime	2400004	1529998	1729998	1769998	1849998	1389998	1629998	1409998	1939998	1739998	1489998	1789998	1749998	1389998	1649998	1499998	1849998	1489998	2489998	1739998	
DiscoverEndTimeABS	(2400005)	(1530703)	(1730703)	(1770106)	(1850167)	(1390970)	(1630547)	(1410649)	(1940633)	(1740860)	(1490703)	(1740927)	(1750188)	(1390627)	(1640741)	(1410747)	(1650570)	(1490117)	(2490325)	(1740934)	
DiscoverStartTime	2200006	1330000	1530000	1570000	1650000	1190000	1430000	1210000	1740000	1540000	1250000	1540000	1550000	1190000	1640000	1210000	1650000	1290000	2230000	1540000	
DiscoverStartTimeABS	(2200007)	(1330752)	(1530711)	(1570108)	(1650169)	(1190972)	(1430549)	(1210551)	(1740635)	(1540862)	(1250710)	(1540929)	(1550190)	(1190629)	(1640743)	(1210749)	(1650572)	(1290119)	(2230327)	(1540936)	
LocalTime	1325993	1324842	1324883	1325486	1325425	1324622	1325045	1325043	1324959	1324732	1324884	1324655	1325404	1324965	1324851	1324845	1325022	1325475	1325267	1324998	
Mode	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ModeNext	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MyAddr	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
MyHopsToSink	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
MyMaxLen	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
MyTxTime	5	1544	1730	5993	414	8288	3573	2497	7179	4825	4902	5099	100	6722	5894	3636	1032	7338	5263	4952	
MyTxTimeABS	(6)	(2396)	(2441)	(16101)	(583)	(9280)	(4122)	(3048)	(7814)	(5687)	(5612)	(6028)	(290)	(7351)	(9037)	(4395)	(1604)	(7455)	(5550)	(5546)	
Nbr	51 92591	15 4 3633	15 4 3674	14 4 475 1 3	2 4 272 0	0 1 9034 8	15 4 3835	10 2 9455 1	4 4 9948 1	12 4 9428	10 1 9296 8	12 4 9361	16 4 3932 0	10 1 9377 8	12 4 9547	10 2 9257 1	16 4 3950 0	0 1 9867 8	0 1 9679 8	12 4 9694 1	
NbrLen	6	1	1	1	1	1	7	1	7	1	1	1	1	1	1	1	1	1	7	7	
nbrK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
nbrY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
nr_of_rows	2	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	
PacketsToSnd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
RescheduleList	5611 10	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1	-1 -1 -1 -1
RescheduleListLen	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
RxData	64 0 137 0	128 15 27 0	128 15 27 0	128 4 27 0	128 4 27 0	64 0 137 0	128 15 27 0	64 0 137 0	128 4 27 0	128 12 27 0	64 0 137 0	128 12 27 0	128 6 27 0	64 0 137 0	128 12 27 0	64 0 137 0	128 6 27 0	64 0 137 0	128 0 55 0	128 12 27 0	
RxDatalen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
RxDatalSrc	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
RxDatalTime	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
SIMInitalizeionTime	1	752	711	109	972	549	951	635	862	710	929	190	629	743	749	572	119	327	596		
SIMNodeLocalTime	1325993	1324842	1324883	1325486	1325425	1324622	1325045	1325043	1324959	1324732	1324884	1324655	1325404	1324965	1324851	1324845	1325022	1325475	1325267	1324998	
SIMposX	20	349	396	951	493	152	368	172	527	660	64	496	537	124	610	274	480	113	145	699	
SIMposY	183	140	296	403	370	217	200	244	450	284	223	36	175	274	157	164	176	230	295	197	
SIMRx	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
SIMRxDatStatus	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
SIMRxDatLen	0	26364	26313	29512	0	26191	0	39	569	0	325	26099	0	449	0	26437	0	10	0		
SIMRxDatStart	-1	23364	23367	57048	45289	-1	33345	-1	56961	-1	56371	45346	-1	56950	-1	46956	-1	1325315	-1		
SIMTx	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
SIMTxDataStatus	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	
SIMTxRange	163	173	179	165	166	177	173	184	180	181	181	181	185	186	170	162	189	189	185	174	
SIMTxStart	1325994	1261544	1261570	1265383	1270414	1119298	1259579	1132492	1307170	1264825	1214902	1265030	1270100	1119722	1265834	1133638	1271033	1217338	1326262	1264992	
TimeToBeacon	7	2	2	7	4	9	2	0	4	4	9	4	4	9	4	9	4	9	8	3	
TimeToBeaconABS	(70006)	(22296)	(22441)	(76101)	(40983)	(95260)	(24122)	(3048)	(47614)	(49687)	(95612)	(46028)	(40290)	(97351)	(46837)	(4395)	(41604)	(97455)	(85590)	(35646)	
TxData	192 0 9 0 4	128 1 31 0	128 2 31 0	128 3 27 0	128 5 31 0	128 6 31 0	128 7 31 0	128 6 27 0	128 9 27 0	128 10 31 0	128 11 27 0	128 12 27 0	128 13 31 0	128 14 27 0	128 15 31 0	128 16 27 0	128 17 31 0	128 18 31 0	128 19 27 0		
TxDatalen	9	31	31	27	31	31	31	27	27	31	27	27	31	27	31	27	31	31	137		

Rys. 25. Widok ekranu pokazującego szczegółowe dane grupy węzłów sieci

Ze względu na bardzo dużą zbieżność symulacji z rzeczywistym działaniem sieci sensorowej, osiągniętej poprzez emulację węzłów sieci w symulatorze, zaproponowane środowisko do symulacji pozwoliło na opracowanie nowego protokołu sieci sensorowej i eliminację nawet trudnych do zlokalizowania błędów programowych czy koncepcyjnych.

## 2.4. Prace badawcze na realizacją sprzętową i metodami projektowania węzłów sieci sensorowych

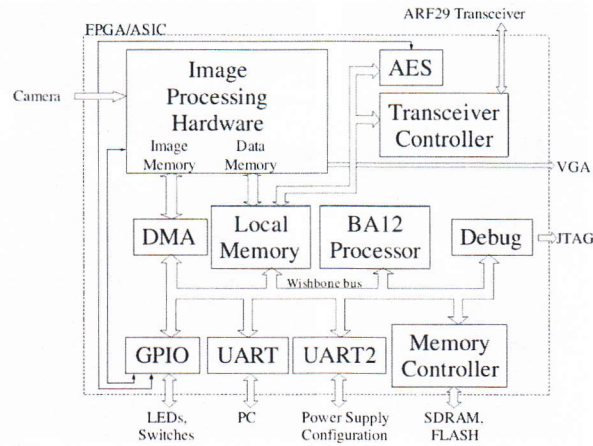
Energooszczędna realizacja węzła sieci sensorowej jest podstawowym celem wyznaczanym przy projektowaniu sieci sensorowej. Uruchamianie algorytmów na procesorze jest bardzo wygodne, gdyż uzyskujemy dużą elastyczność, krótki czas przygotowania oprogramowania, a także możliwość łatwej modyfikacji i aktualizacji danej aplikacji. W przypadku, gdy algorytm musi pracować w sposób ciągły i przetwarzać dużą ilość danych, takie rozwiązanie jest nieefektywne energetycznie. W pracy [K2] pokazano, że przykładowa realizacja programowa

algorytmu AES z kluczem 128-bitowym realizowana przez procesor, wymaga do przetworzenia pakietu danych ok. 3200 cykli zegara. Realizacja sprzętowa potrzebuje na taką samą operację tylko 17 cykli, przy czym obydwie realizacje – procesorowa i sprzętowa zostały zrealizowane w tej samej technologii (układ FPGA) i dla tej samej częstotliwości zegara taktującego. Stosunek energii zużywanej przez realizację procesorową do energii wymaganej przez realizację sprzętową wynosi ok.  $22000\text{pJ}/300\text{pJ} \approx 73$ . Obliczenia dotyczą energii dynamicznej, bez uwzględnienia prądów upływu. Z tego wynika, że dla obliczeń wykonywanych w sposób ciągły, realizacja sprzętowa, z punktu widzenia zużywanej energii, jest uzasadniona.

Innym powodem sprzętowej realizacji algorytmów w sieciach sensorowych jest potrzeba osiągnięcia określonej wydajności obliczeniowej dla danej częstotliwości zegara. Ciągła obróbka strumienia wideo wymaga dużych mocy obliczeniowych, które mogą być nieosiągalne dla pojedynczego procesora. W celu osiągnięcia wymaganej mocy obliczeniowej można wykorzystać dodatkowe procesory lub zrealizować akceleratory sprzętowe jako niezależne bloki lub akceleratory.

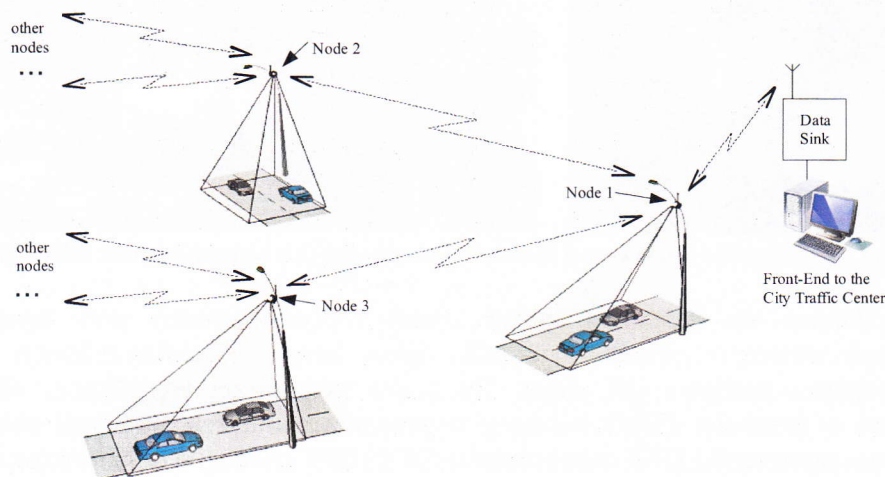
W pracy [P1] oraz [K6] zrealizowano sprzętowo algorytm przetwarzający strumień wideo o prędkości 25fps w postaci potoku w układzie FPGA klasy Xilinx Virtex-4 SX XC4VSX35, z głównym zegarem pracującym z częstotliwością 28.5MHz, przy czym większość bloków algorytmu została taktowana częstotliwością 1.79MHz. Dzięki niskim częstotliwościom zegara osiągnięto pobór mocy około 1200mW, przy mocy dynamicznej rzędu 600mW oraz mocy statycznej wynoszącej również 600mW. Implementacja ta pozwalała na zwiększenie częstotliwości do 135MHz, wtedy możliwe jest przetwarzanie obrazów z prędkością 117fps. Realizacja programowa tego samego algorytmu w języku C, uruchomiona na komputerze PC z procesorem Intel Dual Core o częstotliwości zegara 2.13 GHz, przetwarza dane z prędkością 160fps, ale wymaga 65W mocy zasilania.

Ten sam algorytm zrealizowano również w technologii FPGA z wykorzystaniem dziewięciu dedykowanych procesorów [K5], opracowanych od zera pod kierunkiem habilitanta, wyniki implementacji przedstawiono w skrócie w rozdziale 2.2 niniejszego autoreferatu, szczegółowe rezultaty przedstawiono w [P2]. Zastosowanie zespołu procesorów umożliwiło późniejsze zmiany działania algorytmu przetwarzającego strumień wideo, ze względu na to, że ten sam system został również zrealizowany w technologii ASIC. Porównanie implementacji algorytmu w technologiach FPGA i ASIC opisano w [P3]. Implementacja FPGA wykorzystywała układ Xilinx Virtex-4 LX, zegar taktujący miał częstotliwość 50MHz. Oprócz algorytmu przetwarzania obrazu, wewnątrz tego układu zrealizowano wszystkie inne elementy węzła sieci sensorowej, w tym sprzętowy blok szyfrowania AES. Bloki systemu były kontrolowane przez oprogramowanie napisane w języku C z wstawkami assemblerowymi, działającym na dodatkowym 32-bitowym procesorze BA12 z magistralą Wishbone, o wydajności równoważnej procesorom z rodziny ARM9™. W węzle sieci zastosowano zewnętrzny moduł transceivera o mocy nadawania 500mW i częstotliwości 868MHz oraz kamerę i zewnętrzne pamięci SDRAM i FLASH. Schemat blokowy pokazujący elementy systemu umieszczone wewnątrz układu FPGA przedstawiono na rys. 26.



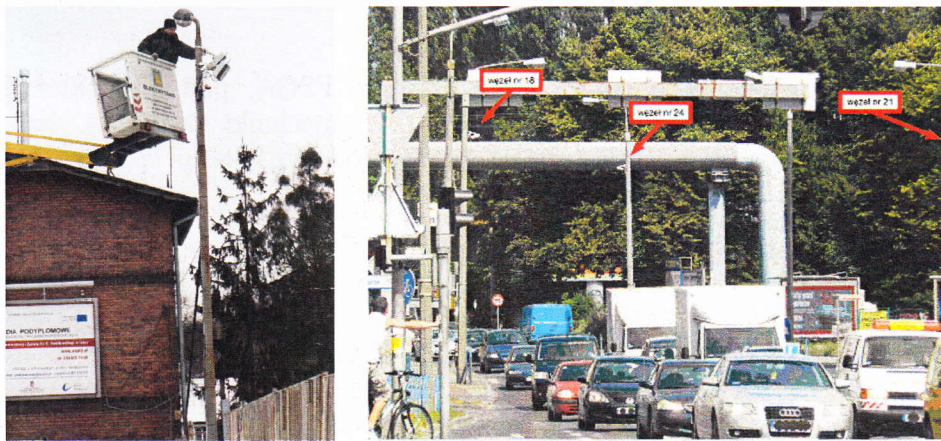
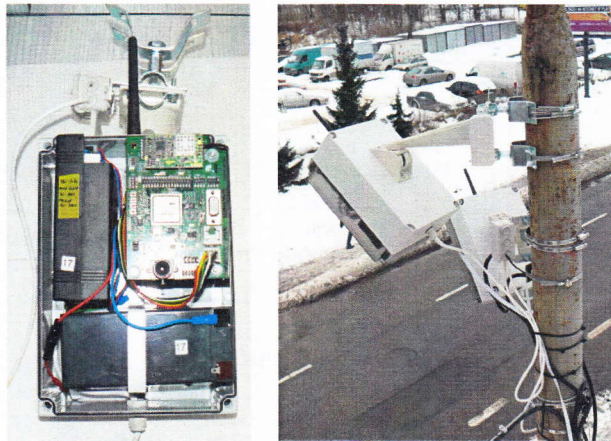
Rys. 26. Schemat blokowy systemu mikroelektronicznego realizującego funkcjonalność węzła sieci sensorowej [P2]

Wykorzystując implementację sprzętową z układem FPGA, pod kierunkiem habilitanta zbudowano i zainstalowano na ulicach w okolicy Politechniki Gdańskiej sieć sensorową składającą się z 22 węzłów do obserwacji pojazdów zgodnie z koncepcją z rys. 27.



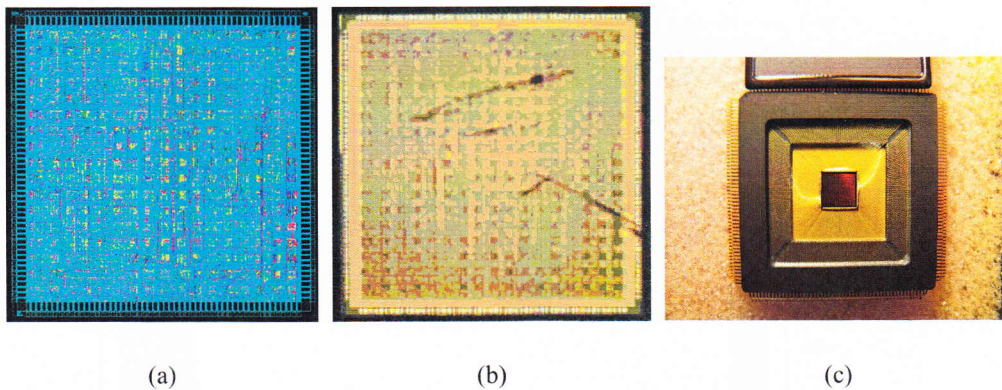
Rys. 27. Widok koncepcyjny sieci sensorowej do pomiarów natężenia ruchu pojazdów w obszarach miejskich

Zdjęcia węzłów sieci sensorowej zbudowanych z wykorzystaniem układu FPGA przedstawiono na rys. 28. Realizacja węzła z układem FPGA pobierała ok. 1500mW mocy z zasilania. W celu redukcji poboru mocy zrealizowano system z rys. 26 w technologii ASIC. Całość zmieściła się na płytce krzemu o wymiarach 5x5mm, układ został zrealizowany w technologii UMC 130nm z wykorzystaniem cyfrowych bibliotek typu *standard cell* oraz pamięci *on-chip* wygenerowanych za pomocą generatora pamięci firmy Faraday. Szczegóły procesu projektowania układu ASIC przedstawiono w [K3]. Przy projektowaniu układu ASIC wykorzystano zarówno wersje LP (*low power*) jak i HS (*high speed*) komórek bibliotecznych, projektowanie odbywało się za pomocą pakietu Cadence SoC Encounter GXL 6.2. W celu obniżenia poboru mocy zastosowano technikę *clock gating*.

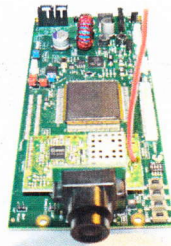


Rys. 28. Fotografie węzłów sieci sensorowej zbudowanych z wykorzystaniem układu FPGA

Przed wysłaniem do produkcji, układ został zoptymalizowany pod kątem zakłóceń sygnałowych wewnątrz układu (*crosstalk*, *signal integrity*), elektromigracji i lokalnych obniżen napięcia zasilania (*IR drop*). Do celów późniejszej weryfikacji, układ scalony wyposażono w kontroler JTAG sterujący wyprowadzeniami zewnętrznymi układu, a także zastosowano przerzutniki DFT oraz kołnierze DFT (*DFT collars*) wokół bloków wbudowanej pamięci *on-chip* do wykrywania błędów wewnątrz układu. Habilitant przeprowadził również weryfikację kodu HDL przed i po syntezie logicznej, wykorzystując do tego celu oprogramowanie *Cadence Conformal Equivalence Checker*. Z 14154 punktów kontrolnych udało się zweryfikować 13972 punkty w sposób automatyczny, pozostałe różnice przeanalizowano i potwierdzono ręcznie. Podobnie przeprowadzona została przez habilitanta weryfikacja układu przed i po implementacji, gdzie udało się zweryfikować automatycznie 14505 punktów kontrolnych z wszystkich 14688. Pozostałe rozbieżności również zostały wyjaśnione ręcznie. Kolejnym krokiem służącym weryfikacji była pełna symulacja układu po implementacji za pomocą symulatora Cadence *ncsim* i odpowiednio przygotowanego modułu *testbench*. Rysunek topografii układu scalonego oraz fotografie zrealizowanego układu przedstawiono na rys. 29.

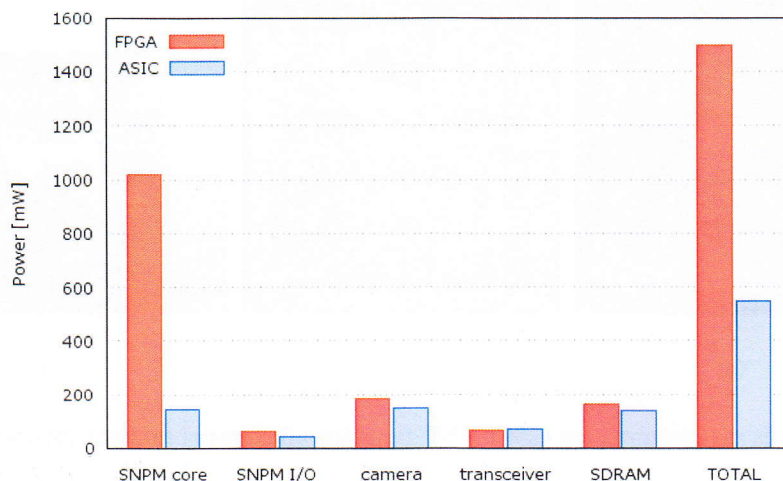


Rys. 29. Topografia zaprojektowanego układu ASIC (a), fotografia gotowej struktury (b) oraz całego układu (c)

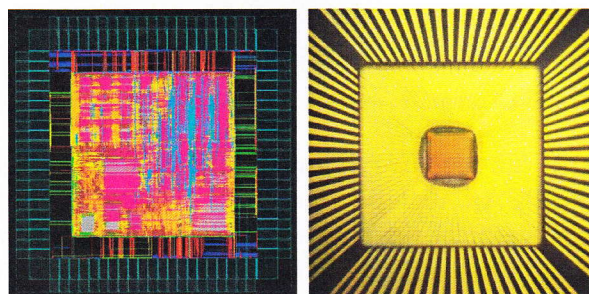


Rys. 30. Fotografia węzła sieci sensorowej z układem ASIC

W układzie ASIC nie ma już możliwości zmiany konfiguracji sprzętowej, tak, jak to miało miejsce w układzie FPGA, dlatego realizacja algorytmu przetwarzania obrazu z wykorzystaniem wielu procesorów umożliwiła modyfikacje algorytmu, gdyż do jego zmiany wystarczyła modyfikacja zawartości pamięci zewnętrznej typu FLASH. Prototypy węzłów sieci sensorowej wykorzystujące układ ASIC (rys. 30) pobierały ok. 550mW, dzięki czemu możliwa stała się bezproblemowa praca węzłów z niewielkim panelem fotowoltaicznym. Do tego celu został zaprojektowany inteligentny zasilacz sterujący ładowaniem akumulatora oraz uzyskiwaniem maksymalnej mocy z baterii słonecznej za pomocą techniki MPPT (*Maximum Power Point Tracking*). Na wykresie z rys. 31 przedstawiono różnice w poborze mocy poszczególnych bloków węzłów sieci zrealizowanych z wykorzystaniem technologii FPGA i ASIC.



Rys. 31. Pobory mocy zasilania przez bloki węzła sieci sensorowej w realizacji FPGA i ASIC. Układ FPGA pracował z napięciami 1.2V i 2.5V dla rdzenia, 3.3V dla bloków I/O, kamery i pamięci SDRAM oraz 3V dla transceivera. Układ ASIC zasilany był napięciem 1.2V dla rdzenia, oraz 3.0V dla bloków I/O i pozostałych układów



Rys. 32. Topografia oraz fotografia zrealizowanego układu akceleratora kompresora H.264. Układy wykonane zostały w technologii UMC 90nm

Innym przykładem sprzętowej realizacji algorytmu do przetwarzania obrazów może być rozwiązanie przedstawione w [K4]. Habilitant brał udział w projektowaniu cyfrowego układu scalonego ASIC w technologii UMC 90nm w wersji L90N Mixed-Mode/RF – 1P9M2T1F z bibliotekami *standard cell* i generatorem pamięci *on-chip* Faraday. Projekt układu ASIC został poprzedzony prototypem z wykorzystaniem układu FPGA Virtex-6 VLX365T. Układ realizował akcelerację sprzętową kompresji obrazu H.264, w tym także transformację DCT (wprost i odwrotna). W projekcie układu ASIC została również zastosowana technika *clock-gating*, taktowanie układu odbywało się zegarem o częstotliwości 100MHz. Przy kompresji z prędkością 45fps dla obrazu HD 1024x768, rdzeń układu pobierał 15mW ze źródła zasilania 1V. Niskomocowe rozwiązanie kompresji obrazu umożliwia implementację takiego algorytmu w sieciach sensorowych.

## 2.5. Podsumowanie autorskiego wkładu i perspektywy dalszego rozwoju

Rozwój mikroelektronicznych sieci sensorowych stał się możliwy dzięki osiągnięciom w wielu dziedzinach nauki. Miniaturyzacja układów i systemów elektronicznych, postępy w budowie źródeł zasilania, rozwój metod i systemów pozyskiwania energii ze środowiska, a także badania dotyczące algorytmów i protokołów sprawiają, że możliwa stała się realizacja sieci sensorowych przeznaczonych do wielu różnych zastosowań. Habilitant po doktoracie



aktywnie włączył się w te badania, działając w wielu kierunkach, począwszy od badań nad algorytmami przetwarzania obrazów i protokołami transmisyjnymi, a kończąc na realizacji miniaturowych systemów mikroelektronicznych zbudowanych z układów FPGA, a także z opracowanych i zaprojektowanych układów typu ASIC.

Sieci sensorowe są zazwyczaj kojarzone z miniaturowymi urządzeniami wyposażonymi w proste czujniki, np. temperatury lub światła. Badania habilitanta w kierunku rozwoju niskomocowych algorytmów przetwarzania obrazów pokazały, że możliwa jest realizacja wizyjnych sieci sensorowych, składających się z autonomicznych węzłów sieci wyposażonych w podsystemy przetwarzające obraz oraz wyciągające wnioski z zaobserwowanych zdarzeń.

Metody i algorytmy przetwarzania obrazów, obecnie intensywnie rozwijane na świecie oraz szeroko opisywane w literaturze, są przystosowane dla typowych systemów komputerowych o dużej mocy obliczeniowej i bez specjalnych ograniczeń, jeśli chodzi o zasilanie. Specyfika sieci sensorowych wymaga innego podejścia, aby realizacja danych algorytmów była w ogóle możliwa w węzle sieci sensorowej, który posiada liczne ograniczenia dotyczące mocy zasilania, wymiarów czy kosztów wytworzenia sprzętu – w takich przypadkach wymuszone jest stosowanie analizy obrazu o znacznie gorszych parametrach, np. czarno-białego, o niewielkiej rozdzielczości (np. 128x128 pikseli), ze zmniejszoną dynamiką sygnałów (np. 4-bitowej reprezentacji jasności pikseli) oraz metody nie wymagające dużych mocy obliczeniowych. Taka analiza jest dużo trudniejsza, wymaga odrębnego podejścia i opisujących je publikacji w literaturze praktycznie nie ma – badania habilitanta mają na celu wypełnienie tej niszy.

Pod kierunkiem habilitanta opracowane zostały unikalne algorytmy detekcji obiektów ruchomych możliwe do implementacji w miniaturowych niskomocowych systemach zasilanych bateryjnie. Przebadane zostały różne realizacje sprzętowe tych algorytmów, począwszy od przetwarzania potokowego, skończywszy na realizacji wieloprocessorowej. W wyniku badań, została opracowana przez habilitanta oryginalna rodzina procesorów dedykowanych do przetwarzania obrazów w sieciach sensorowych.

Habilitant opracował również autorski, dedykowany protokół sieci radiowej, służący jako warstwa komunikacyjna sieci sensorowych. Algorytm posiadający niewielkie zapotrzebowanie na moc zasilania, wymagający prostego jednokanałowego transceivera, jest skalowalny do dużej liczby węzłów i umożliwia realizację wielu funkcjonalności, takich jak: transmisja danych z węzłów sieci sensorowej do węzłów zbierających dane, samoorganizacja topologii sieci sensorowej, adaptacja topologii sieci do zmian w środowisku, dystrybucja danych do wszystkich węzłów sieci umożliwiająca aktualizację firmware'u i oprogramowania węzłów, w tym także rekonfigurację układów FPGA (o ile takie występują w węzłach sieci). Protokół umożliwia również zbieranie danych wideo w zakresie, na jaki pozwala przepustowość sieci.

W celu umożliwienia dalszych badań nad protokołami transmisyjnymi w sieciach sensorowych, habilitant zaprojektował oraz zrealizował oryginalne, dedykowane środowisko symulacyjno-emulacyjne, które w bardzo dokładny sposób pozwala symulować badaną sieć sensorową, umożliwiając wgląd w stan wewnętrzny wszystkich węzłów symulowanej sieci.

Habilitant aktywnie brał udział w badaniach, projektowaniu i realizacji mikroelektronicznych złożonych, prototypowych układów i systemów z wykorzystaniem układów FPGA, takich jak koprocessor QoS opracowany i zrealizowany dla firmy Intel [G3][G4]. Jest współautorem projektów 4 nowatorskich układów scalonych, począwszy od programowalnych układów analogowych, a skończywszy na układach cyfrowych składających się z wielu milionów tranzystorów i wyposażonych w technologie JTAG i DFT. Wszystkie układy zostały wyprodukowane oraz uruchomione z sukcesem.

Podsumowanie najważniejszych osiągnięć habilitanta:

- opracował unikatowe algorytmy detekcji obiektów ruchomych możliwe do implementacji w miniaturowych niskomocowych systemach zasilanych bateryjnie;
- opracował oryginalną rodzinę procesorów dedykowanych do przetwarzania obrazów w sieciach sensorowych;
- zrealizował unikatowe sieci sensorowe w technologiach FPGA i ASIC;
- opracował autorski, dedykowany protokół sieci radiowej, służący jako warstwa komunikacyjna sieci sensorowych;
- zrealizował miniaturowe systemy mikroelektroniczne wykorzystujące układy FPGA i ASIC;
- zaprojektował i zrealizował dedykowane środowisko symulacyjne sieci sensorowych;
- jest współautorem projektów 4 układów scalonych, począwszy od programowalnych układów analogowych, a skończywszy na układach cyfrowych składających się z wielu milionów tranzystorów i wyposażonych w technologie JTAG i DFT.

W wyniku przeprowadzonych badań oraz zrealizowanych, habilitant wykazał, że możliwa jest realizacja wizyjnych systemów sensorowych. Pod kierunkiem habilitanta powstały unikatowe rozwiązania wizyjnych sieci sensorowych w technologii FPGA i ASIC, w których habilitant zastosował opisane powyżej rozwiązania dotyczące analizy obrazu i transmisji danych. Przykładowa realizacja takiego systemu została pod kierunkiem habilitanta zaprojektowana, wykonana i z sukcesem uruchomiona – sieć składająca się z 22 węzłów działała przez rok czasu w warunkach rzeczywistych, obserwując ulice w rejonie Politechniki Gdańskiej i zbierając dane o ruchu pojazdów.

Na podstawie posiadanych doświadczeń, habilitant prowadzi obecnie dalsze badania nad algorytmami detekcji i rozpoznawania obrazów dedykowanych dla sieci sensorowych, aby umożliwić niskomocowym węzłom sieci rozpoznanie coraz większej liczby zdarzeń i obiektów w obrazie wideo. Znajomość wielu aspektów projektowania sieci sensorowych, takich jak projektowanie sprzętu, protokołów i algorytmów, umożliwia habilitantowi aktywne działania w tej dziedzinie.

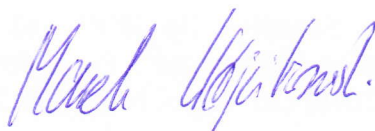
## 2.6. Literatura

- [1] H. Karl, A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley & Sons, 2007.
- [2] I. Stojmenovic, *Handbook of Sensor Networks, Algorithms and Architectures*, Wiley & Sons, 2005.
- [3] L. Zhou, J. M. Kahn, K. S. J. Pister, "Corner-cube retroreflectors based on structure-assisted assembly for free-space optical communication," *Journal of Microelectromechanical Systems*, vol.12, no.3, pp.233-242, June 2003.
- [4] P. Mazières, D. P. Druschel, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric", *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2002.
- [5] A.P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol.27, no.4, pp.473-484, Apr 1992.
- [6] L. Benini, A. Bogliolo, G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.8, no.3, pp.299-316, June 2000.
- [7] C. Guo, L. C. Zhong, J. M. Rabaey, "Low Power Distributed MAC for Ad Hoc Sensor Networks", *Proc. IEEE GlobeCom*, San Antonio, 2001.

- [8] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, S. Roundy, "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking", *IEEE Computer*, 33(7), pp.42-48, 2000.
- [9] J. L. Hill, D. E. Culler, "Mica: a wireless platform for deeply embedded networks", *IEEE Micro*, vol.22, no.6, pp.12-24, Nov/Dec 2002.
- [10] Yen-Kuang Chen, S. Y. Kung, "Trend and Challenge on System-on-a-Chip Designs", *Journal of Signal Processing Systems*, vol.53(1), pp.271-229, 2008.
- [11] S. Roundy, *Energy scavenging for Wireless Sensor Networks*, Kluwer Academic Publishers, 2003.
- [12] P. Levis, *TinyOS Programming*, Cambridge University Press, 2009.
- [13] P.V.C. Hough, "Machine Analysis of Bubble Chamber Pictures", *Proc. Int. Conf. High Energy Accelerators and Instrumentation*, 1959.
- [14] H. Zhang, X. Bai, J. Zhou, J. Cheng, H. Zhao, "Object Detection via Structural Feature Selection and Shape Model", *IEEE Trans. Image Process.*, 22(12), 4984-4995, 2013.
- [15] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit.*, 886-893, 2005.
- [16] B. E. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers", *Proc. Fifth Annual Workshop on Computational Learning Theory*, ACM Press, 144-152, 1992.
- [17] C. Cortes, V. Vapnik, "Support-vector network", *Machine Learning*, 20:273-297, 1995.
- [18] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool, "The PASCAL Visual Object Classes Challenge 2006 (VOC 2006) Results", *Technical Report*, Univ. of Oxford, 2006.
- [19] P. Geismann, A. Knoll, "Speeding Up HOG and LBP Features for Pedestrian Detection by Multiresolution Techniques", *Proc. 6th Int. Symposium Advances in Visual Computing*, ISVC 2010, Las Vegas, NV, USA, 243-252, 2010.
- [20] I. Daubechies, *Ten Lectures on Wavelets*, Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1992.
- [21] *Network Protocols Handbook*, Javin Technologies Inc., 2004.
- [22] F. A. Tobagi, L. Kleinrock, "Packet Switching in Radio Channels: Part II The Hidden Terminal Problem in CSMA and Busy Tone Solutions", *IEEE Transactions on Communications*, 23(12), pp. 1417-1433, 1975.
- [23] Joan Daemen, Vincent Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard*, Springer, 2002.
- [24] *The Network Simulator, ns-2*, <http://www.isi.edu/nsnam/ns>
- [25] *ns-3*, <http://www.nsnam.org>
- [26] *Scalable Simulation Framework (SSF)*, <http://www.ssfnet.org>
- [27] *Global Mobile Information Systems Simulation Library (GloMoSim)*, <http://pcl.cs.ucla.edu/projects/glomosim>

## **2.7. Projekty badawcze, w których brał udział habilitant**

- [G1] Projekt badawczy KBN nr 8T11B04910: "Obiektowe hybrydowe systemy ekspertowe dla projektowania i diagnostyki układów elektronicznych oraz sterowania procesami pomiarów automatycznych", realizacja w 1995r., projekt badawczy - wykonawca.
- [G2] Projekt badawczy KBN nr 8T11B01114: "Metody projektowania i realizacji scalonych układów CMOS i BiCMOS w strukturach programowalnych", okres realizacji 1998-1999r. – wykonawca.
- [G3] Umowa Nr 015459 z Intel Technology Poland, Nr ITP-TUG-2001/2002-2: "Quality of Service (QoS) Hardware Building Blocks: FPGA Implementation", realizacja w latach 2001-2002, projekt badawczy dla firmy Intel – wykonawca.
- [G4] Umowa z Intel Technology Poland, Nr ITP-WETI-ASIC-R-2003-Q3: "CMOS ASIC Coprocessor", realizacja w 2003r., projekt badawczy dla firmy Intel – wykonawca.
- [G5] Projekt rozwojowy MNiSzW Nr R02-01401: "Projekt i realizacja zintegrowanych modułów sieci sensorowej w technologiach FPGA i ASIC do monitorowania środowiska i ruchu pojazdów w obszarach miejskich", realizacja w latach 2006-2009 – główny wykonawca.
- [G6] Projekt rozwojowy MNiSzW Nr O R00 0046 09: "Bezprzewodowy system bezpieczeństwa wykorzystujący inteligentne mikrouządzenia rozpoznawcze do przekazywania obrazu i dźwięku wewnątrz budynków", realizacja w latach 2009-2011, konsorcjum: PG – Lider, RADMOR Gdynia, SiGarden, Gdańsk – wykonawca.
- [G7] Projekt rozwojowy MNiSzW Nr O R00000312: "Zintegrowany system fotografii laserowej do monitoringu otwartych przestrzeni i zapobieganiu zagrożeniom terrorystycznym", realizacja w latach 2010-2012, konsorcjum PG, WAT – Lider, CTM Gdynia, Hardsoft Kraków) – główny wykonawca ze strony PG.



Gdańsk, 18.01.2016 r.

dr inż. Marek Wójcikowski