

**POLITECHNIKA WARSZAWSKA**  
**WYDZIAŁ ELEKTRYCZNY**  
**INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ**

**PRACA DYPLOMOWA INŻYNIERSKA**  
**na kierunku ELEKTROTECHNIKA**  
**specjalność: AUTOMATYKA I INŻYNIERIA KOMPUTEROWA**



**Kacper ANTCZAK**

**Nr albumu: 230146**

Rok akad.: 2012/2013  
Warszawa, 25.09.2012

**ZASTOSOWANIE PRZETWARZANIA OBRAZÓW DO**  
**WYKRYWANIA USZKODZEŃ W KASETONACH REKLAMOWYCH**

**Zakres pracy:**

1. *Wprowadzenie i sformułowanie celu pracy.*
2. *Opis wybranych elementów przetwarzania obrazów.*
3. *Projekt i implementacja algorytmów przetwarzania obrazów do detekcji usterek w kasetonach reklamowych.*
4. *Badania eksperymentalne i analiza otrzymanych wyników.*
5. *Podsumowanie i wnioski.*

**Kierujący pracą: dr inż. Witold Czajewski**

**Konsultant:**

**KIEROWNIK ZAKŁADU STEROWANIA**

**Dr hab. inż. Bartłomiej Beliczyński**

**Termin złożenia pracy: 28.01.2013 r.**

Praca wykonana i obroniona pozostaje  
własnością Instytutu, Katedry i nie będzie  
zwrócona wykonawcy.

# **Zastosowanie przetwarzania obrazów do wykrywania uszkodzeń w kasetonach reklamowych**

## **Streszczenie**

Praca inżynierska miała za zadanie rozwiązać problem praktyczny, jakim jest zaprojektowanie wizyjnego systemu technicznego do wykrywania uszkodzeń w kasetonach reklamowych. Rozdział pierwszy zawiera wprowadzenie do zagadnień związanych z przetwarzaniem obrazów, charakterystykę kasetonów oraz problemów związanych z ich użytkowaniem. Opisuje on również powody podjęcia zagadnienia i precyzuje wymagania, których spełnienie zapewni funkcjonalność systemu.

Projekt wymaga zaznajomienia się ze specyfikacją elementów przetwarzania obrazu biblioteki OpenCV, ich składnią w kompilatorze C++ oraz praktycznym zastosowaniem. Dodatkowo do ukończenia projektu wymagane było zaznajomienie się z rynkiem komputerów miniaturowych, kamer internetowych, a także możliwymi sposobami przesyłu danych. Wszystkie zagadnienia teoretyczne związane z systemem zostały zawarte w rozdziale drugim i trzecim.

Stworzony program porównuje obiekt badany z obiektem wzorcowym w poszukiwaniu usterek. Eksperymenty opisane w rozdziale czwartym, sprawdzają poprawność działania systemu podczas zaplamienia, pęknięcia, znacznej eksploatacji oświetlenia, a także częściowej i całkowitej awarii zasilania kasetonu. W pracy zawarte są również badania działania systemu podczas analizy liter przestrzennych.

Eksperymenty wykazały dużą skuteczność systemu w wykrywaniu uszkodzeń, program zadziałał niepoprawnie tylko w jednym przypadku. Porównując postawione w pracy teoretyczne wymagania systemu z jego praktycznym działaniem można stwierdzić, że wizyjny system techniczny spełnia postawione cele. Jest to uniwersalny układ który działa z dużą skutecznością, jego obsługa jest prosta, a dodatkowo, zaproponowane rozwiązania sprzętowe pozwalają wierzyć, że projekt jest uzasadniony biznesowo. Rozdział piąty zawiera wnioski i podsumowanie dotyczące pracy, a także dalsze możliwości rozbudowy systemu.

# **Application of image processing in damage detection of advertising signs**

## **Abstract**

The aim of this BCs thesis was solving a practical problem, which is designing the technical system able to detect damage of advertising signs. Chapter one provides introduction to image processing, characteristics of advertising signs and problems associated with their use. It also describes main reasons for taking this BCs thesis and specifies the requirements to be met to ensure the functionality of the system.

The project requires familiarization with the specification of image processing algorithms included in OpenCV library, their syntax in C++ compiler and practical application. In addition, in order to finish the BCs thesis, it was required to become familiar with market of miniature computers, webcams and possible ways of computer communication. All theoretical issues associated with the system are contained in the second and third chapter.

Created program compares the test object with the object model in search of damage. The experiments described in chapter four, analyze the effectiveness of the system in case of stains, cracks, significant use of lighting and also partial and total power failure in advertising signs. BCs thesis also includes tests of the system behavior regarding spatial letters analysis.

Experiments have shown that implemented project have high efficiency in detecting damage, the program worked incorrectly in only one case. Comparison of posed theoretical project requirements with its practical behavior shows that the technical system meets all of the posed goals. It is a universal system that works with high efficiency, its service is simple and in addition, proposed hardware solutions lead us to believe that the project is profitable. The fifth chapter contains the conclusions and summary of BCs thesis, as well as further possibilities for system development.

Warszawa, dnia 7.02.2013..... roku.

Politechnika Warszawska  
Wydział Elektryczny

### OŚWIADCZENIE

Świadom/a odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa inżynierska/magisterska pt. Zastosowanie przetwarzania obrazów do wykrywania uszkodzeń w kasetonach reklamowych

- została napisana przeze mnie/nas samodzielnie
- nie narusza niczyich praw autorskich
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Imię i Nazwisko dyplomanta: Kacper Antczak

Podpis dyplomanta: *Kacper Antczak*.....

*(Imiona, Nazwiska, podpisy pozostałych dyplomantów w przypadku prac zespołowych)*

1	WPROWADZENIE .....	1
1.1.	Cel i układ pracy .....	3
2	OPIS WYBRANYCH ELEMENTÓW PRZETWARZANIA OBRAZÓW .....	4
2.1.	Modułowość OpenCV .....	5
2.2.	Opis elementów przetwarzania obrazu .....	6
2.2.1.	Akwizycja informacji .....	6
2.2.2.	Filtracja danych .....	6
2.2.3.	Analiza obrazów w celu wykrycia kluczowych cech.....	13
3	PROJEKT I IMPLEMENTACJA ELEMENTÓW PRZETWARZANIA OBRAZÓW DO DETEKCJI USZKODZEŃ NA KASETONACH REKLAMOWYCH.....	17
3.1.	Kluczowe własności systemu technicznego.....	17
3.2.	Specyfikacja sprzętowa systemu.....	18
3.3.	Charakterystyka najczęściej występujących usterek kasetonów reklamowych.....	22
3.4.	Implementacja elementów przetwarzania obrazów do detekcji uszkodzeń w kasetonach reklamowych .....	25
3.4.1.	Niwelowanie wpływu tła na późniejszą analizę .....	26
3.4.2.	„Zszywanie” zdjęć .....	29
3.4.3.	Zapewnienie tożsamości położenia obiektów .....	33
3.4.4.	Analiza danych .....	36
4	BADANIA EKSPERYMENTALNE I ANALIZA OTRZYMANYCH WYNIKÓW .....	40
4.1.	Opis i zasadność badań eksperymentalnych .....	40
4.2.	Badania eksperymentalne i analiza wyników .....	41
4.2.1.	Pęknięcie/przerwanie kasetonu.....	41
4.2.2.	Zaplamienie kasetonu .....	42
4.2.3.	Znaczne wyeksploatowanie kasetonu.....	43
4.2.4.	Częściowa awaria zasilania kasetonu .....	45
4.2.5.	Analiza poprawności działania systemu podczas bezawaryjnej pracy kasetonu	50
5	PODSUMOWANIE I WNIOSKI.....	52
5.1.	Wnioski z przeprowadzonych badań.....	52
5.2.	Możliwości rozbudowy .....	53

## WPROWADZENIE

W ostatnich latach jesteśmy świadkami bardzo szybkiego rozwoju technologii. Oprócz szybszych komputerów i ekranów dotykowych coraz częściej spotykamy się z pojęciem automatyzacji stanowisk pracy. Ludzie są powoli zastępowani systemami technicznymi. Aby to usprawnić, projektowane są układy, które imitują działanie ludzkich narządów zmysłu. Człowiek przetwarza otrzymane bodźce całkowicie naturalnie, bez problemu wyciąga wnioski dotyczące otoczenia, otrzymując informacje z jednego z narządów zmysłu. Komputer jednak nie ma żadnych zdolności podejmowania decyzji na podstawie np. samego obrazu z kamery. Kiedy człowiek widzi na ekranie dwóch ludzi, którzy się kłócą, dla jednostki centralnej to tylko macierz wypełniona liczbami, stąd też głównym celem automatyki jest takie przetworzenie danych, aby były one przydatne i czytelne dla systemów technicznych. Z racji tego, iż wzrok ma największy wpływ na nasze życie, na podejmowane przez nas działania. Stworzenie systemu poprawnie przetwarzającego dane z otoczenia na podstawie obrazu stanowi najistotniejszy problemem w procesie automatyzacji stanowisk pracy. Taką analizę obrazu, pozwalającą na kierowanie działaniami zautomatyzowanych struktur, nazywamy cyfrowym przetwarzaniem obrazu.

Systemy działające na podstawie danych wizyjnych znajdują coraz większe zastosowanie w życiu człowieka. Programy analizujące informacje z kamer można znaleźć na każdym kroku, w nadzorze procesów produkcyjnych, jako imitacja wzroku w robotyce czy też w sterowaniu ruchem drogowym. Oprócz tego, systemy wizyjne pomagają w zapewnieniu bezpieczeństwa obywateli. Wdrażane są programy, które na podstawie danych z monitoringu policyjnego, mogą poinformować o możliwym zamiarze kradzieży, źle zaparkowanym samochodzie czy przejechaniu na czerwonym świetle. Nadal jednak istnieje wiele dziedzin, w których automatyzacja nie została wdrożona, a mogłaby ona znacznie usprawnić niektóre procesy. Jedną z takich dziedzin, jest analiza usterek na kasetonach reklamowych.

Z powodu coraz większej konkurencji na rynku, wszystkie firmy, te nowo powstałe jak również te istniejące od dłuższego czasu, szukają sposobów by rozreklamować swoje usługi. Do tego celu wykorzystują każdą możliwą drogę, Internet, pocztę pantoflową, marketing szeptany czy oznakowanie reklamowe. Niektóre z form dotarcia do potencjalnego klienta, jak np. Internet, są bardzo proste i łatwe w użytkowaniu, po wprowadzeniu, nadzór nad ich poprawnym działaniem nie stanowi dla firmy znacznego obciążenia. Niestety, inne wymagają od firm zdecydowanie więcej jej zasobów. Jedną z bardziej wymagających form reklamy jest

korzystanie z zewnętrznego oznakowania reklamowego. Najbardziej powszechnym rodzajem takiego oznakowania są kasetony. Są to przestrzenne obiekty, zwykle prostopadłościowe, dobrze widoczne z daleka, dzięki intensywnemu wewnętrznemu podświetleniu. Ich budowa jest bardzo prosta: składają się z profilu, najczęściej aluminiowego oraz lica. Lico jest istotą reklamową kasetonu, gdyż to właśnie na nim widnieje charakterystyczny dla danej firmy logotyp. Zwykle wykonuje się je z plexi lub tkaniny winylowej. To właśnie prostota, a co za tym idzie niska cena, spowodowała bardzo duże zainteresowanie tym sposobem promowania produktu. Dodatkowo, nieskomplikowany sposób produkcji pozwala na duże zróżnicowanie rozmiarów i kształtów kasetonów, dzięki czemu każda firma może posiadać unikatowe i wyróżniające z tłumu oznakowanie reklamowe.

Aby kaseton reklamowy spełniał swoją funkcję, jego działanie musi być w pełni sprawne. Zawsze powinien przyciągać wzrok, za dnia mocnymi kolorami, w nocy, intensywnie świecąc. Jednak każdy przedmiot czasami się psuje, dlatego też firmy produkujące kasetony są odpowiedzialne za okresowe oględziny oznakowania, w celu sprawdzenia poprawnego działania obiektu i jego naprawy, w przypadku wykrycia usterki. Niestety, przy dużej liczbie kasetonów (liczonej w tysiącach) łączny koszt dojazdu do każdej placówki używającej tego rodzaju oznakowania reklamowego, może być dla firmy serwisującej rujnąjący. Dzieje się tak szczególnie w sytuacji, gdy większość oględzin jest wykonywana niepotrzebnie, ponieważ z powodu prostoty produktu, jego wadliwość nie jest wysoka. Dodatkowo czas spędzony przez serwisantów na dojazdach nie może być wykorzystany w żaden konstruktywny sposób, co powoduje straty dla firmy. Często zdarza się również, że usterka zostaje wykryta dopiero podczas oględzin, a klient nieświadomie korzysta z niesprawnego oznakowania, które nie spełnia swojej funkcji.

Wprowadzenie automatyzacji wykrywania uszkodzeń na kasetonach, które byłoby w stanie automatycznie i bez ingerencji człowieka powiadomić o zaistniałej usterce, mogłoby znacznie usprawnić czasochłonny i kosztowny proces dbania o prawidłowe funkcjonowanie oznakowania reklamowego. Głównymi korzyściami dla pracodawcy byłoby obniżenie kosztów dojazdu i liczby pracowników wykorzystywanych do serwisu oznakowania. Dla klienta to skrócenie czasu wykrywania uszkodzenia, a co za tym idzie przyspieszenie procedury naprawy usterki. Z kolei dla pracowników korzyścią jest zdecydowanie mniejsza liczba godzin spędzonych w podróży, z dala od domu. Dlatego też, zaprojektowanie wizyjnego systemu technicznego, który zniweluje potrzebę oględzin kasetonów, uważam za bardzo ważne zagadnienie.

## 1.1. Cel i układ pracy

Głównym celem pracy jest stworzenie stabilnego, prostego i uniwersalnego programu, bazującego na algorytmach przetwarzania obrazów, który, na podstawie zanalizowanych danych, wykryje uszkodzenie w kasetonie reklamowym. System ten opierać się będzie na porównywaniu zdjęcia obiektu ze wzorcem zrobionym zaraz po montażu kasetonu. Dodatkowo, w przypadku wystąpienia usterki, system ma za zadanie powiadomić pracownika odpowiadającego za nadzór placówek o zaistniałym uszkodzeniu oraz umożliwić wgląd w dane wizyjne kamery, w celu potwierdzenia pojawienia się defektu. Program ten w przyszłości może być zastosowany do zautomatyzowania procesu informowania o usterekach firmy, która zajmuje się czynnościami gwarancyjnymi i pogwarancyjnymi, serwisem oraz odświeżaniem wyglądu paneli reklamowych. Dodatkowym celem tej pracy jest sprawdzenie, czy taki system jest opłacalny oraz zbadanie, w jakich przypadkach można go zastosować będąc pewnym jego niezawodności.

Praca zakłada podejście praktyczne i szczegółowe do kwestii analizy obrazu. Problemy związane z rozwiązaniami sprzętowymi, sposobem akwizycji danych, metodami przesyłania komunikatów na temat wystąpienia usterek oraz samej komunikacji z urządzeniem zostały przemyślane pod kątem ogólnym i hipotetycznym a zagadnienia te mogą zostać rozwinięte w przyszłych publikacjach.

Oprócz wstępu, praca zawiera kolejne cztery rozdziały.

Pierwszy z nich opisuje bibliotekę OpenCV, historię jej powstania, modułowość, zastosowania oraz przybliża działanie algorytmów przetwarzania obrazów, które zostały wykorzystane podczas projektowania wyżej opisanego systemu analizy wizyjnej.

Drugi rozdział opisuje proces projektowania programu. Obrazuje tok rozumowania Autora i wybiera sposoby na rozwiązanie problemu od pierwszych wersji testowych po wersję finalną systemu.

Trzeci rozdział zawiera wyniki badań eksperymentalnych oraz ich analizę.

Kończący, czwarty rozdział, podsumowuje przebieg projektu oraz precyzuje wnioski

## **ROZDZIAŁ 2**

### **OPIS WYBRANYCH ELEMENTÓW PRZETWARZANIA OBRAZÓW**

Projekt systemu analizującego zdjęcia kasetonów w poszukiwaniu uszkodzeń został wykonany przy użyciu biblioteki OpenCV( „Open Source Computer Vision”). Jest to zbiór funkcji znacznie ułatwiających programistom skomplikowane zadanie, jakim jest szeroko rozumiane przetwarzanie obrazu. Narzędzie to znacznie przyspiesza i ułatwia pracę z obrazami, szczególnie ich filtracje i analizę.

Biblioteka ta została stworzona przez firmę Intel. Twórcy doszli do jednak wniosku, iż jest to urządzenie, którego głównym zadaniem jest rozwój skomplikowanej i złożonej dziedziny, jaką jest cyfrowe przetwarzanie obrazu, a zadanie to wykona najlepiej, jako projekt wolnego dostępu, którym zostaje do dzisiejszego dnia. Nad rozwojem biblioteki w tym momencie sprawuje pieczę Willow Garage. Jest to laboratorium zajmujące się rozwojem robotyki, do której jak wiadomo analiza wizyjna jest niezbędna. Liczba funkcji w OpenCV rośnie z każdą aktualizacją i w obecnej chwili jest ich około pięciuset. Biblioteka ta zawiera funkcje ułatwiające filtrowanie obrazu. Wiele prostych jak i skomplikowanych przekształceń obrazu pozwala na śledzenie obiektów, tworzenie panoramy z wielu ząbających się zdjęć i wiele innych. Dzięki różnorodności funkcji, narzędzie to wykorzystywane jest w wielu dziedzinach - zaczynając od dydaktyki i projektów badawczych, poprzez tworzenie animacji do filmów, kończąc na aplikacjach tworzących panoramy i systemach wykrywających źle zaparkowany samochód na podstawie nagrań z monitoringu policyjnego. W początkowym projekcie biblioteka OpenCV została stworzona dla języka programistycznego C, jednak z biegiem czasu, jej kompatybilność została poszerzona o większość języków programistycznych m.in. C++, Java, Python czy Matlab. Oprócz tego istnieją biblioteki przeznaczone na oprogramowanie mobilne Android firmy Google oraz iOS firmy Apple.

## 2.1. Modułowość OpenCV

Aby ułatwić pracę z biblioteką, została ona podzielona na moduły tematyczne. Każdy segment zawiera tylko funkcje związane z konkretnym zagadnieniem przetwarzania obrazu. Oprócz uporządkowania wiedzy taki podział zmniejsza ilość zużytej pamięci, gdyż użytkownik ma możliwość wykorzystania jedynie tych modułów, które zostaną zastosowane w kompilowanym kodzie. Początkowo istniały jedynie 4 moduły, jednak wersja 2.0 OpenCV przyniosła znaczną przebudowę biblioteki, między innymi większą specjalizację i liczbę segmentów. W wersji 2.4.3 bibliotekę możemy podzielić na osiem głównych części oraz kilka pośrednich, bardzo wyspecjalizowanych modułów zajmujących się konkretnymi zagadnieniami.

Najbardziej istotne dla wykonania projektu segmenty biblioteki OpenCV to:

- **core** – zawiera podstawowe funkcje biblioteki,
- **imgproc** – obejmuje wszystkie algorytmy związane z przetwarzaniem obrazu,
- **highgui** – zawiera interfejsy użytkownika wysokiego poziomu oraz obsługa wejścia/wyjścia,
- **video** – odpowiada za obróbkę wideo,
- **calib3d** – składa się z algorytmów pozwalających na rekonstrukcję 3D oraz kalibracje kamer,
- **features2D** – bardzo istotny segment zawierający funkcje wykrywające kluczowe właściwości obrazów 2D,
- **objdetect** – zawiera algorytmy kluczowe dla wykrywania obiektów,
- **nonfree** – segment składający się z algorytmów komercyjnych,
- **flann** – szybka biblioteka do przybliżania najbliższych sąsiadów,
- **stitching** – moduł automatyzujący „zszywanie” ze sobą dwu lub więcej zdjęć.

Pełną dokumentację biblioteki można znaleźć w: [1] i [3]

## 2.2. Opis elementów przetwarzania obrazu

Działanie systemu wizyjnego wykrywającego uszkodzenia w kasetonach reklamowych można podzielić na cztery etapy:

- akwizycja informacji,
- filtracja danych,
- analiza obrazów w celu wykrycia kluczowych cech,
- wyciągnięcie wniosków na bazie przeanalizowanych informacji i ich przedstawienie.

Po przyjęciu założenia, iż przetwarzany obraz zawierać będzie tylko kaseton reklamowy oraz tło, etapy segmentacji obiektów oraz ich klasyfikacji zostały pominięte.

Finalny projekt systemu został stworzony w języku C++, tak więc funkcje przetwarzania obrazu biblioteki OpenCV, które zostaną opisane poniżej przedstawione będą z punktu widzenia tegoż języka. Każda funkcja zawarta w podrozdziale 2.2 posiada dokładną dokumentację w [3].

### 2.2.1. Akwizycja informacji

Aby możliwa była jakakolwiek analiza, pierwszy etap musi stanowić dostarczenie danych, które można będzie przetworzyć. Funkcja **imread** ma za zadanie wykonać właśnie ten pierwszy krok. Wczytuje ona dane obrazu z dysku komputera do programu, jednak docelową funkcją odpowiadającą za pozyskanie danych będzie **VideoCapture** odpowiadająca za wczytanie jednej klatki z pliku wideo. Taki wybór został podyktowany tym, iż ostatecznie system analizujący kasetony będzie miał za zadanie pobierać dane z kamer internetowych.

Przed przystąpieniem do filtracji obrazu należy pamiętać, iż wiele algorytmów operuje jedynie na obrazach przedstawionych w formie szarościowej. Przekształcenia w zakresie modeli przestrzeni barw umożliwia funkcja **cvtColor**.

### 2.2.2. Filtracja danych

Na tym etapie można zastosować wiele różnych funkcji. Krok ten przeprowadza się głównie w celu przygotowania danych do późniejszej analizy.

Najważniejsze metody przekształcenia obrazu w systemach wizyjnych możemy podzielić na cztery rodzaje[2]:

- przekształcenia geometryczne,
- przekształcenia punktowe,
- przekształcenia kontekstowe liniowe,
- przekształcenia kontekstowe nieliniowe.

Do grupy przekształceń geometrycznych należą wszystkie odbicia, obroty, ogólnie wszystkie działania mające na celu zmianę geometrii obrazu. Istotną w C++ funkcją wykorzystującą przekształcenia geometryczne jest **warpPerspective**. Algorytm ten, na podstawie wcześniej obliczonej macierzy homografii (która zostanie dokładniej opisana w dalszej części pracy), dokonuje przekształceń mających na celu dopasowanie położenia obiektu badanego tak, aby było równe położeniu obiektu wzorcowego.

Kolejną grupą filtrów są przekształcenia punktowe. Jednym z wielu bardzo przydatnych algorytmów tego rodzaju jest wyrównanie histogramu (w C++, odpowiednikiem jest **equalizeHist**). Histogram to sposób przedstawiania częstotliwości występowania danych wartości w obrazie. Dzięki tej metodzie z łatwością możemy zorientować się, np. jaki odcień szarości występuje w obrazie najczęściej. Funkcja histogramu, zlicza liczbę występowania danej wartości w badanej macierzy liczb. Pomimo prostoty narzędzie to jest niezwykle przydatne i wykorzystywane przez wiele funkcji. Jedną z nich to wyżej wspomniana metoda wyrównania histogramu. Algorytm ten *„...polega na takiej zmianie wartości punktów obrazu, aby w równych przedziałach histogramu ilość punktów obrazu była w przybliżeniu taka sama. Operacja ta pozwala na wypuklenie mało kontrastowych szczegółów w obrazie.”*[4]. Jeżeli więc potrzebujemy wyróżnić możliwie największą liczbę szczegółów, metoda wyrównania histogramów to algorytm, który powinniśmy zaimplementować podczas filtracji obrazu.



Zdjęcie 2.1 Zdjęcie wzorcowe. Źródło: materiały własne.



Zdjęcie 2.2 Wynik wykonania funkcji wyrównania histogramu na Zdjęcie 2.1

Powyższe Zdjęcie 2.1 i Zdjęcie 2.2 przedstawiają działanie wyrównania histogramu. Jak widać, tło zdjęcia zostało uwydatnione. Po zastosowaniu jednego prostego algorytmu, można dostrzec znacznie więcej szczegółów. Niestety zdarzają się sytuacje, w których zastosowanie tej metody podczas filtracji utrudni późniejszą analizę. Takimi wyjątkami są obrazy posiadające bardzo jasne elementy. W tych przypadkach, wykonanie wyrównania histogramu może spowodować utratę szczegółów obrazu oraz poprawienie widoczności tła.



Zdjęcie 2.3 Obraz wzorcowy



Zdjęcie 2.4 Obraz po wykonaniu na zdjęciu 2.3 wyrównania histogramu.

Po analizie zdjęć 2.3 i 2.4 można dojść do wniosku, że widoczność tła zdjęcia uległa zwiększeniu. Dodatkowo niektóre szczegóły obiektu zostały utracone. W tym przypadku zastosowanie funkcji `equalizeHist` znacznie pogarsza późniejsze możliwości analityczne systemu wizyjnego. Szczegółowy opis oraz inne rodzaje korekcji histogramów można znaleźć w [4].

Drugą funkcją stosującą przekształcenie punktowe jest porównanie (w C++ funkcja **compare**). W zależności od metody porównania, funkcja ta może sprawdzać czy wartość piksela jednego zdjęcia równa się wartości tego samego piksela w drugim, czy jest większa lub mniejsza itd. Przewidziana jest również możliwość porównania pikseli z określoną, stałą wartością. W przypadku spełnienia warunku, wartość piksela macierzy wynikowej funkcji jest równa zero, gdy założenie nie zostało spełnione, wartość jest równa jeden. Jej zastosowania to głównie progowanie obrazów lub porównywanie identyczności.

Przekształcenia kontekstowe liniowe to następna omawiana grupa filtrów. Główną różnicą odróżniającą przekształcenia kontekstowe od punktowych czy geometrycznych jest wpływ otoczenia analizowanego piksela na jego wartość wyjściową. Dodatkowo, działanie tych algorytmów wykazuje zależność liniową między wartością wyjściową i wejściową piksela. Najczęściej stosowanym filtrem kontekstowym liniowym jest filtr splotowy (konwolucja).

Zastosowanie funkcji splotu na danych obrazu, można ująć, jako, „...*wyznaczenie – dla każdego punktu obrazu - kombinacji liniowej wartości jego sąsiadów z pewnymi współczynnikami wagowymi.*”[2]. Gdzie macierz zawierająca wagi tych współczynników nazywamy maską splotu.

Jednym z takich przekształceń jest dwukrotne pomniejszenie obrazu. W pewnych okolicznościach, gdy dla przykładu, macierz danych jest zbyt obszerna aby przeprowadzić zaplanowane obliczenia w krótkim czasie, a jej szczegóły nie są istotne. W tym wypadku można zastosować funkcję **pyrDown**. Jest to algorytm, którego każda kolejna iteracja powoduje czterokrotne zmniejszenie liczby danych do przetworzenia, co może znacznie zwiększyć szybkość działania.

Kolejnym użytecznym przekształceniem wykorzystującym splot jest dolnoprzepustowy filtr Gaussa (jego odpowiednikiem w OpenCV jest funkcja **blur**). Przekształcenie to rozmywa obraz z intensywnością zależną od parametrów zdefiniowanych podczas wywołania funkcji. Zdjęcie 2.5 przedstawia wynik działania funkcji blur, która została wykonana na zdjęciu 2.1.



Zdjęcie 2.5 Wynik użycia funkcji rozmycia z maską 3x3 na Zdjęcie 2.1

Ostatnim omawianym rodzajem filtracji obrazu są przekształcenia kontekstowe nieliniowe. Ich główną cechą to brak liniowej zależności pomiędzy wartością wejściową i wyjściową piksela. Przekształcenia kontekstowe nieliniowe dzielą się na filtry medianowe i morfologiczne.

W przypadku filtrów medianowych, wartość wyjściowa przekształcanego piksela jest równa medianie wartości pikseli sąsiednich. U podstaw każdego filtru morfologicznego stoją dwa bazowe operatory: erozji (**erode**) oraz dylacji (**dilate**). Operacja erozji polega na znalezieniu lokalnego minimum wśród wartości pikseli objętych zadaną maską i nadpisanie wartości środkowego piksela, tym minimum.

Dylacja jest działaniem odwrotnym, tak więc wartość środkowego piksela przyjmuje maksimum lokalne. Przypadek, gdy erozja występuje bezpośrednio po dylacji, nazywamy otwarciem, natomiast kiedy dylacja występuje zaraz po erozji, zamknięciem.

Otwarcie stosujemy, gdy mamy w zamiarze pozbyć się jaśniejszych elementów na ciemniejszym tle. Metoda ta nadaje się szczególnie do niwelowania wpływu zniekształcenia typu „sól” na późniejszą analizę. Natomiast zamknięcia, które jest działaniem dualnym do otwarcia, używamy, aby usunąć z obrazu ciemne szczegóły na jaśniejszym tle. Filtr ten jest wyjątkowo skuteczny przy usuwaniu szumu typu „pieprz” z obrazu. Dodatkowo powyższy sposób filtracji, oprócz pozbycia się niechcianych elementów zdjęcia, wygładza również krawędzie obiektów nieusunięte podczas działania przekształcenia.



Zdjęcie 2.6 Wynik otwarcia o czterokrotnej iteracji zastosowanego na Zdjęcie 2.1

Na zdjęciu 2.6 można dostrzec zdecydowane zmniejszenie liczby szczegółów przy zachowaniu kształtu obiektu.

Zdarzają się sytuacje, w których aby dokładnie oddać istotę obiektu, potrzebujemy wykonać kilka zdjęć, gdzie każde z nich zawiera w sobie jedynie część interesującej nas rzeczy. W takich przypadkach, aby należycie zanalizować obiekt potrzebne jest uprzednie „zszycie” obrazów w całość, czyli takie przekształcenie geometryczne analizowanego obrazu, aby położenie odpowiadających sobie punktów kluczowych badanego przedmiotu były równe. Jest to możliwe tylko wtedy, gdy analizowane zdjęcia „nakładają się na siebie”, czyli część pierwszego jest widoczna na drugim. Jednak, aby zapewnić poprawnie i dokładnie „zszyty” obraz, w którym nie widać nierównych krawędzi, używane zdjęcia muszą spełniać konkretne dodatkowe wymagania. W przypadku obiektu płaskiego, ruch kamery może być dowolny, z wykluczeniem dużych zmian perspektywicznych. Natomiast przy „zszywaniu” zdjęć z obiektem trójwymiarowym, ruch kamery ogranicza się jedynie do rotacji wokół matrycy światłoczułej. Drugi, dość rygorystyczny warunek, można złagodzić w momencie, gdy odległość pomiędzy obiektywem, a obiektem jest wielokrotnie większa niż długość ogniskowej ( jak np. podczas wykonywania panoramy gór). W takim wypadku, obiekt może być traktowany, jako płaski.

Implementacja tej metody jest zadaniem skomplikowanym, wymagającym od użytkownika użycia kilku następujących po sobie funkcji:

- odnalezienia punktów kluczowych i deskryptorów zszywanych obrazów oraz łączenie odpowiadających sobie punktów w pary,
- obliczenia macierzy homografii przy pomocy funkcji **findHomography**,
- użycia powyższej do przekształcenia perspektywicznego zszywanego obrazu przy pomocy funkcji **warpPerspective**,
- połączenia obu zdjęć w jedno.

Biblioteka OpenCV została zaktualizowana o dodatkowy moduł, **Stitching**, który automatyzuje stworzenie panoramy na podstawie kilku zdjęć. Wykonuje wszystkie wymagane do „zszycia” obrazów operacje a ponadto niweluje różnice jasności obiektu przy pomocy operacji blendowania kolorów oraz wygładzania niedokładnie „zszytych” krawędzi. Podczas „zszywania” może zdarzyć się sytuacja, w której kontrast części obiektu na zdjęciu pierwszym nie odpowiada kontrastowi zdjęcia drugiego. W takim przypadku, na granicy ich połączenia, pojawi się znaczna różnica kontrastu. Blendowanie ma za zadanie „rozetrzeć” tę różnicę tak, aby nie była widoczna.

### 2.2.3. Analiza obrazów w celu wykrycia kluczowych cech

Ekspozowanie cech kluczowych(ang. features) jest bardzo istotnym etapem przetwarzania obrazów. To właśnie dzięki śledzeniu tych własności obiektów możliwe jest poprawne sterowanie robotami i wykrywanie poruszających się obiektów. Cechy te, to zwykle specyficzne, możliwe do rozróżnienia struktury pikseli

- krawędzie,
- linie,
- zakończenia linii,
- grzbiety(bridges), szkielety,
- tekstury,
- skupiska(blobs) -koła/kule.

Aby poprawnie odnaleźć kontury obiektu, należy działanie to poprzedzić wyekspozowaniem krawędzi. Do tego zadania możemy wykorzystać opracowany przez Johna F. Canny’ego w 1986 roku detektor krawędzi (w C++ wywołujemy za pomocą funkcji **Canny**). Jest to funkcja, która odnajduje wszystkie możliwe krawędzie na podstawie gradientu. Jej działanie opiera się na trzech etapach. Pierwszym z nich jest zastosowanie

dolnoprzepustowego filtra Gaussa w celu pozbycia się nieistotnych elementów z obrazu. Jest to podyktowane dużą czułością detektora na nieusunięty szum. Następnym krokiem jest wyznaczenie gradientu intensywności koloru w obrazie, co pozwala na wyeksponowanie krawędzi. Ostatnim etapem jest pocienianie otrzymanych krawędzi w celu otrzymania, jednopikselowej obwiedni obiektów występujących na obrazie wejściowym. Więcej informacji na temat detektora Canny można znaleźć w [5].

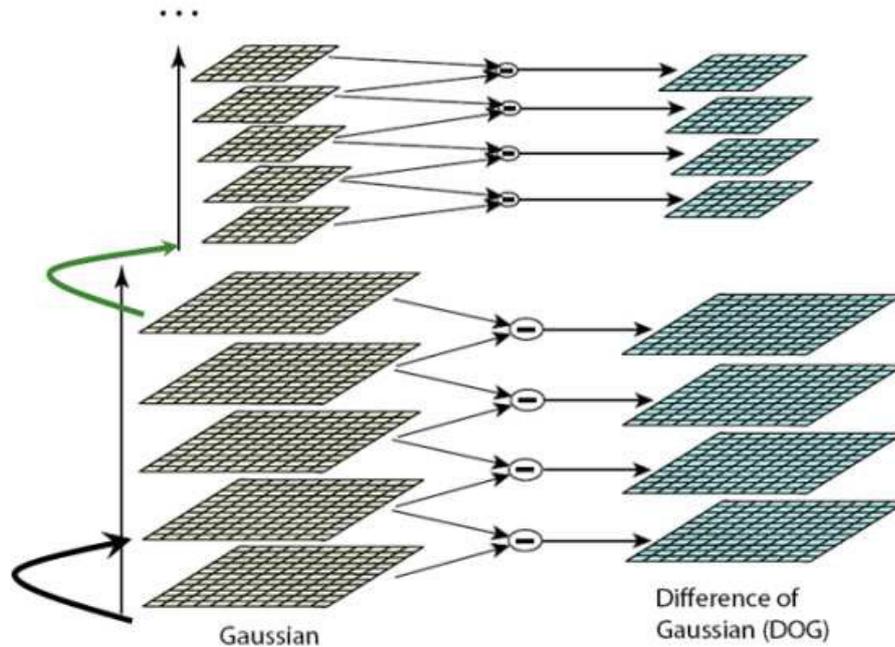


Zdjęcie 2.7 Detektor Canny’ego zastosowany na Zdjęcie 2.1

Algorytm, który zapoczątkował rozwój uniwersalnych i odpornych metod opisu cech kluczowych na zdjęciach, to stworzony przez Dawida Lowe(1999, 2004) deskryptor **SIFT** (Scale Invariant Feature Transform). Jest on bardzo przydatny w wielu dziedzinach wizji komputerowej powiązanej z dopasowywaniem punktów pomiędzy różnymi perspektywami zdjęcia 3-D, „zszywaniu” zdjęć oraz rozpoznawaniu obiektów. Deskryptor SIFT jest odporny na translacje, rotacje oraz skalowanie. Jest także zdolny pracować poprawnie przy niedużej zmianie perspektywy oraz różnicy w oświetleniu obiektu.

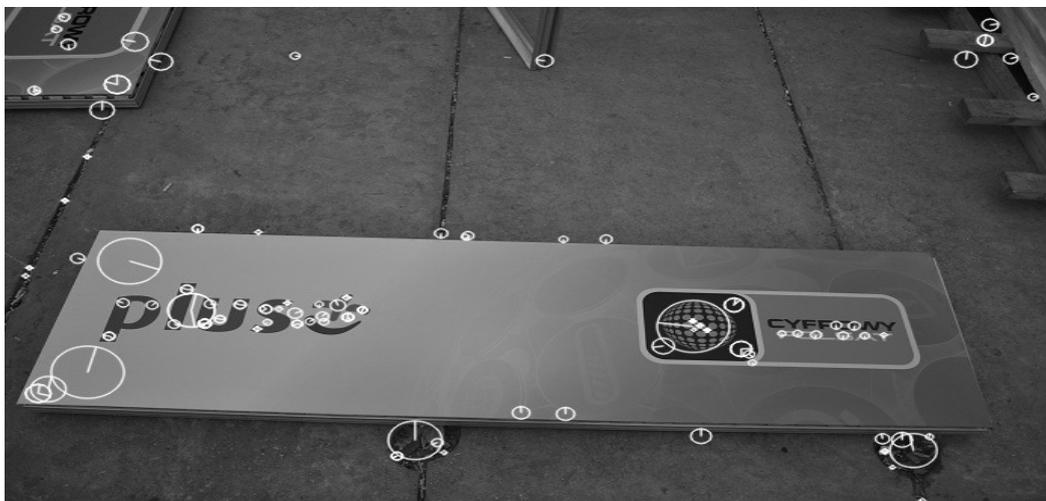
W oryginalnym zamyśle deskryptor SIFT zawierał metodę odnajdywania cech kluczowych z obrazów szarościowych na podstawie liczenia statystyk lokalnych kierunków gradientu intensywności obrazu. Wartości te zostają do siebie dodane tworząc podsumowujący opis lokalnych struktur obrazu w lokalnym sąsiedztwie naokoło punktów kluczowych. Wyliczenie tych danych ma na celu późniejsze użycie ich do łączenia w pary korespondujących ze sobą punktów kluczowych z różnych zdjęć.

Algorytm SIFT odnajduje punkty kluczowe obiektu poddając obraz kolejnym Gaussowskim filtrom dolnoprzepustowym, po czym odejmuje od siebie kolejne wyniki wyjściowe zastosowania filtra (DOG, zdjęcie Zdjęcie 2.8), dając w wyniku zdjęcia przefiltrowane środkowo-przepustowo. Następnie znajdowane jest lokalne maksimum, które przyjmuje się jako punkt kluczowy.



Zdjęcie 2.8 Sposób filtrowania obrazu w celu odnalezienia lokalnego maksimum. Źródło: [8]

Dla każdego punktu kluczowego tworzony jest również jego deskryptor, który został zaproponowany przez Davida Lowe (1999, 2004) jako coś w rodzaju zależnego od pozycji histogramu lokalnego gradientu sąsiadów naokoło punktu. Aby utrzymać niezależność deskryptorów na skalowanie, lokalne sąsiedztwo musi zostać znormalizowane w sposób odporny na skalowanie. W celu utrzymania niezależności na obrót, zostaje wyznaczona dominująca orientacja lokalnego sąsiedztwa powstała z analizy lokalnych kierunków wektorów gradientu. Jest ona następnie używana do orientacji siatki, na podstawie której liczony jest histogram deskryptora. Niezależność na zmiany iluminacji jest zapewniona poprzez normalizację deskryptora do sumy jednostkowej. Dzięki temu histogram będzie odporny na lokalne zmiany jasności wokół punktu kluczowego. Dokładna dokumentacja deskryptora SIFT znajduje się w [7] oraz [8]



Zdjęcie 2.9 Operacja wykrycia punktów kluczowych SIFT wykonana na zdjęciu Zdjęcie 2.1

Deskryptor SIFT jest algorytmem dokładnym, dającym dobre rezultaty podczas skalowania, obrotów, zmiany intensywności otoczenia czy zwiększającego się rozmycia. Jednak posiada jedną ogromną wadę - działa bardzo powoli. Długi czas obliczeń spowodowany jest dokładnym liczeniem każdego z gradientów punktów kluczowych.

Z powodu rozwijającej się technologii oraz rosnącego zapotrzebowania na obróbkę cyfrową zdjęć i filmów, potrzeba rozwoju zajęła również algorytmy wykrywania cech kluczowych obiektów. Niestety, niemożliwym jest stworzenie jednej metody do opisu punktów kluczowych w obrazie, która byłaby uniwersalna i optymalna w każdym możliwym przypadku. Dlatego też, zaprojektowane zostały różne deskryptory posiadające indywidualny zestaw cech, odpowiadający wymaganiom stawianym podczas innych rodzajów przetwarzania cyfrowego. Oprócz deskryptora SIFT, możemy wyróżnić następujące, najczęściej używane, algorytmy wykrywania kluczowych właściwości obrazu[7]:

- SURF (Speeded-Up Robust Feature descriptor) – równie odporny na przekształcenia jak SIFT, jednak dużo szybszy,
- LAZY – około trzy-krotnie szybszy od SURF, jednak nie aż tak odporny na przekształcenia,
- BRIEF – najszybszy, nie jest niezmienniczy na obrót i skalowanie,
- ORB(Oriented-BRIEF) – delikatnie wolniejszy od BRIEF, jednak jest on niezmienniczy na obrót.

Deskryptory zostały mocno wyspecjalizowane. Dlatego też, gdy najważniejszym celem działania ma być szybkość a obraz do przetwarzania jest prosty, bez skalowania oraz obrotów, oczywistym wyborem będzie BRIEF. Do śledzenia obiektów najlepsze wyniki daje deskryptor ORB. Natomiast, w przypadku skomplikowanych analiz obrazu takich jak dla przykładu „zszywanie” zdjęć czy odnajdywanie obiektów w innych obrazach, powinniśmy wybrać algorytm SURF lub LAZY.

## ROZDZIAŁ 3

### PROJEKT I IMPLEMENTACJA ELEMENTÓW PRZETWARZANIA OBRAZÓW DO DETEKcji USZKODZEŃ NA KASETONACH REKLAMOWYCH.

#### 3.1. Kluczowe własności systemu technicznego

W czasach szybkiego rozwoju technologii, implementacja systemów technicznych jest coraz prostsza i szybsza. Dodatkowo zautomatyzowane stanowiska działają z mniejszą zawodnością niż ludzie, a także, po ich wdrożeniu, eksploatacja jest zdecydowanie tańsza. Jednak, aby system wizyjny był funkcjonalny, musi spełniać kilka specyficznych dla swojej funkcji warunków. W przypadku projektu wykrywającego uszkodzenia w kasetonach reklamowych, te warunki to:

- niezawodność i wytrzymałość - od systemu wymaga się, aby pracował bezawaryjnie, co najmniej dwa lata, z małą liczbą fałszywie pozytywnych ostrzeżeń o usterce;
- niski koszt - wdrożenie systemu nie może być droższe niż całkowity dwuletni koszt dojazdu pracowników do placówki w celu jej oględzin (ok. 200zł);
- dokładność - czułość wykrywania defektów projektu w założeniu ma być porównywalna do wrażliwości ludzkiego oka;
- uniwersalność - wymagana jest poprawna praca przy większości rodzajów kształtów i wymiarów kasetonów;
- prostota użytkowania - zaimplementowanie systemu musi być na tyle proste, aby mogło zostać wykonane przez człowieka bez jakiegokolwiek wiedzy na temat bibliotek OpenCV czy programowania.

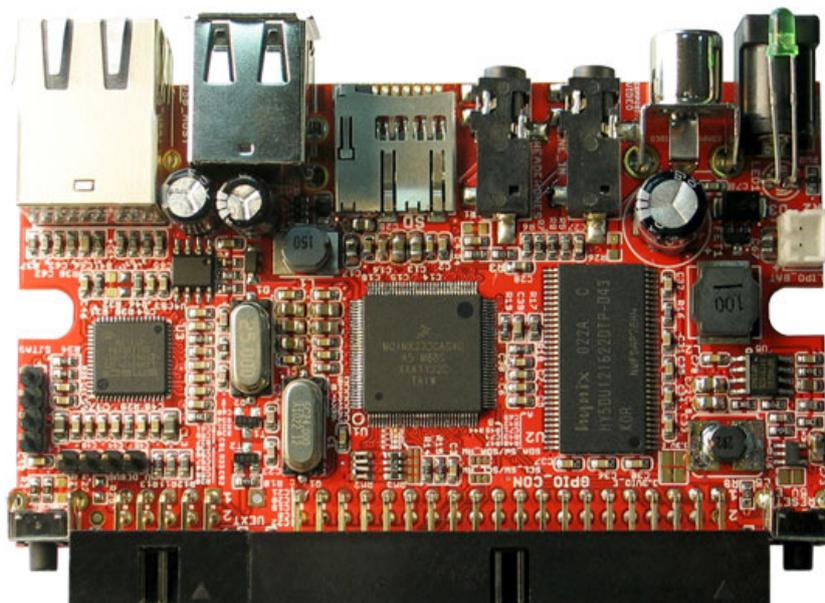
Pierwsze dwa warunki mogą być spełnione jedynie przy pomocy dobrego doboru konfiguracji sprzętowej systemu. To dzięki optymalnemu połączeniu jednostki obliczeniowej, narzędzia zbierającego informacje oraz ich komunikacji możemy zapewnić niezawodne, wytrzymałe i tanie użytkowanie. Dokładność to bardzo istotny warunek, który chcąc spełnić należy dobrać sposób akwizycji obrazu o wystarczająco wysokiej rozdzielczości a także zaprojektować program dostatecznie czuły na pojawiające się uszkodzenia w kasetonie. Za uniwersalność oraz prostotę systemu odpowiada jedynie stworzenie programu precyzyjnie przetwarzającego uzyskane dane.

Jednak, aby jakakolwiek analiza była możliwa, wymagane jest wcześniejsze zapewnienie akwizycji wzorcowego zdjęcia obiektu. Dlatego też, zaraz po montażu kasetonu w placówce, jeden z pracowników będzie zobligowany do wykonania w systemie technicznym pierwszego zdjęcia obiektu, które następnie będzie przechowywane jako wzorzec dla analizy oznakowania pod kątem awaryjności.

### 3.2. Specyfikacja sprzętowa systemu

W celu spełnienia pierwszych trzech warunków, wymagane jest, aby elementy sprzętowe projektu były czułe, tanie i bezawaryjne. Do tego bardzo istotna jest również niezawodność sposobu łączności pomiędzy urządzeniem odpowiedzialnym za dostarczenie informacji wizyjnych z jednostką obliczeniową, a także komunikacji indywidualnego systemu wizyjnego z centralnym punktem nadzoru wszystkich placówek.

Z racji specyfikacji projektu, najważniejszym kryterium przy wyborze komputera do przetwarzania obrazów w systemie technicznym, był jego całkowity koszt. Całkowity czas analizy obrazu nie jest istotny, więc wydajność platformy jest drugorzędną kwestią. Poniżej zostało przedstawione porównanie dwóch komputerów, które spełniały wymagania stawiane przez system. Reszta podobnych platform była zdecydowanie za droga. Pierwszą rozpatrywaną platformą jest wyprodukowana przez firmę Olimex - A13-OLinuXino-MICRO.



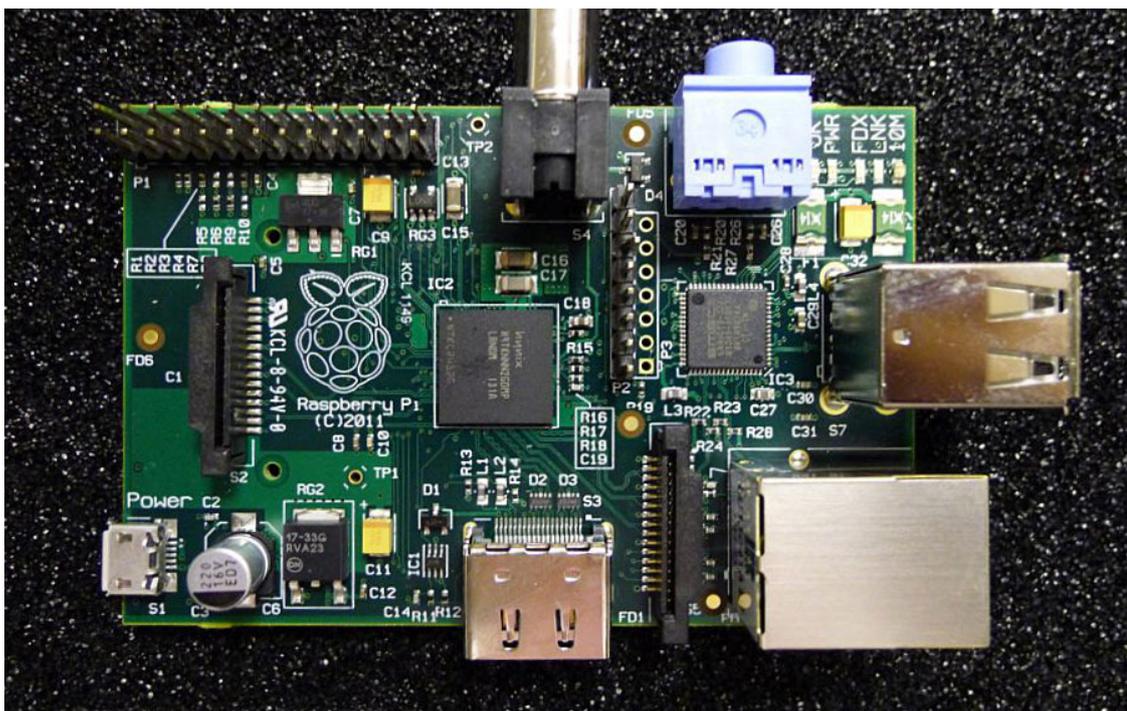
Zdjęcie 3.1 A13-OLinuXino Źródło:[8]

Jej najważniejsze parametry wyglądają następująco:

- CPU: A13 Cortex A8 processor at 1GHz, 3D Mali400 GPU,
- pamięć (SDRAM): 256 MB,
- porty USB 2.0: 1,
- wyjście wideo: VGA,
- nośnik danych: złącze kart SD / MMC / SDIO,
- połączenia sieciowe: brak,
- wymiary: 100 mm × 85 mm,
- obsługiwane systemy operacyjne: Linux.

Jest to platforma o dobrej specyfikacji dla projektu. Jej cena jest zadowalająca (przy dużej liczbie sztuk ok.110zł), jednak posiada jedynie jeden port USB, więc aby obsługiwać kamerę internetową oraz Wi-Fi wymagane będzie użycie zewnętrznego USB Hub'a. Co więcej, OLinuXino nie posiada wbudowanego portu RJ45, co może być znaczącym utrudnieniem. Producent ten oferuje, oprócz zaprezentowanej wersji MICRO, kilka innych platform o różnych specyfikacjach jednak są one zbyt drogie. Dodatkowe informacje w [8].

Drugą jednostką obliczeniową wymagającą uwagi jest stworzony przez Raspberry Pi Foundation, Raspberry Pi wersji B. Jest to platforma komputerowa o wielkości karty kredytowej.



Zdjęcie 3.2 Raspberry Pi Źródło: [9]

### Specyfikacja techniczna Raspberry Pi:

- CPU: 700 MHz ARM1176JZF-S core (ARM11 family),
- pamięć (SDRAM): 256 MB,
- porty USB 2.0: 2,
- wyjścia wideo: Composite RCA, HDMI,
- nośnik danych: złącze kart SD / MMC / SDIO,
- połączenia sieciowe: 10/100 Ethernet (RJ45),
- wymiary: 85,60 × 53,98 mm,
- obsługiwane systemy operacyjne: Debian GNU/Linux, Fedora, Arch Linux.

Po przeanalizowaniu danych technicznych można zauważyć, że jest to urządzenie optymalne do wykonywania zadań określonych w projekcie. Jego bezawaryjność zapewnia architektura, w jakiej został wykonany. Jest to komputer, którego komponenty są scalone w jedną małą płytkę. Pomimo niewielkich rozmiarów może być użyty do skomplikowanych analiz obrazu. Oczywiście prędkość przetwarzania danych będzie bardzo wolna, jednak nie jest to warunek konieczny do spełnienia w celu optymalnego funkcjonowania. Dodatkowo komputer ten nie jest wymagający finansowo. Projekt Raspberry Pi zakładał, iż będzie to bardzo tani, ale w pełni funkcjonalny komputer. Dlatego też, jego cena (po odliczeniu Vat'u, 112 zł) jest całkowicie zadowalająca. Dodatkowe informacje na temat Raspberry Pi można znaleźć w [9].

Porównanie obu platform ze sobą prowadzi do wniosku, że Raspberry Pi to komputer odpowiadający wszystkim wymaganiom projektu. Jest on tani, wytrzymały i w przeciwieństwie do OLinuXino, posiada dwa porty USB, co jest bardzo wygodne w przypadku podłączenia kamery internetowej i modułu Wi-Fi. To Raspberry Pi został wybrany jako komputer, który będzie wykorzystywany przez system techniczny do przetwarzania obrazów.

Kolejnym, wyjątkowo ważnym dla działania układu urządzeniem, jest element systemu odpowiadający za akwizycje danych.

Do tego celu zostanie wykorzystana kamera o następujących, kluczowych, wymaganiach:

- wykonywanie dobrej, jakości zdjęć w nocy z bliskiej odległości( ok. 50cm),
- wyjście USB,
- możliwość użytku zewnętrznego, w tym wodoszczelność i odporność na zmiany temperatury,
- niski koszt.

Niestety na rynku nie istnieje urządzenie w pełni odpowiadające wymaganiom stawianym przez projektowany system. Wyjście USB posiadają jedynie kamery internetowe, które niestety nie są wodoszczelne i nie mogą zostać użyte na zewnątrz. Natomiast kamery przemysłowe można wykorzystywać w warunkach zewnętrznych - niestety posiadają one niekompatybilne z Raspberry Pi wyjście. Jednak w przypadku wprowadzenia tego projektu w życie, liczba kamer potrzebnych do wdrożenia systemów będzie tak duża (docelowo, kilka tysięcy sztuk), że pojawi się możliwość zaprojektowania i zamówienia kamer dokładnie odpowiadającym stawianym wymaganiom bezpośrednio u producenta w odpowiadającej cenie.

Przesył informacji z urządzenia wizyjnego do jednostki obliczeniowej zapewni zwykły przewód USB. Natomiast łączność całego systemu z centralnym punktem nadzoru nie jest już tak prostą czynnością. Jest to proces, który stwarza wiele problemów, a czasem także generuje dość duże koszty. W zależności od sytuacji komunikacja ta zostanie umożliwiona przy pomocy jednej z trzech metod, w której każda prowadzi do zapewnienia podłączenia systemu do Internetu.

W przypadku, gdy placówka wykorzystująca system wizyjny posiada Wi-Fi, do którego możliwy jest wolny dostęp, jak np. na stacjach benzynowych, wykorzystany zostanie dedykowany dla Raspberry Pi miniaturowy moduł Wi-Fi, dołączany do jednego z wyjść USB. Sposób ten jest prosty i tani w wykonaniu, ponieważ komponent potrzebny do połączenia komputera z siecią bezprzewodową kosztuje ok. 30zł.

Niestety, nadal część punktów, w których docelowo ma znaleźć zastosowanie projektowany system nie posiada sieci bezprzewodowej. Występuje tam jednak sieć przewodowa, której można użyć. Raspberry Pi posiada wbudowane wejście RJ45, więc w takim przypadku dostęp do Internetu nie sprawia żadnego problemu, a jedynym dodatkowym kosztem jest wartość kupionego przewodu sieciowego.

Trzecią możliwością jest sytuacja, w której placówka nie posiada dostępu do Internetu lub nie zgadza się na jej udostępnianie użytkownikom zewnętrznym. W takim przypadku w celu

połączenia z Internetem, zastosować można bezprzewodowy modem łączący się z Internetem przy pomocy sieci komórkowej. Jednak jest to sposób zawodny i problematyczny w użyciu. Często sytuacją jest rozłączanie się takiego rodzaju modemu, a dojazd w celu ponownego połączenia byłby kłopotliwy. Dodatkowo koszt tego sposobu łączności jest wysoki. Dlatego też ta metoda powinna być wybierana jedynie w ostateczności.

Po podliczeniu kosztów okazuje się, iż podłączenie systemu do istniejącej w placówce sieci (bezprzewodowej lub przewodowej) nie posiada uzasadnienia ekonomicznego. Łączny koszt kamery i Raspberry Pi wyniesie około 200zł, dodatkowo, moduł Wi-Fi lub przewód sieciowy to wydatek około 30zł. Nie wliczając kosztów ramienia, na którym przymocowana będzie kamera oraz obudowy komputera, całkowity koszt przekracza granicę opłacalności. Jednak, biorąc pod uwagę czas pracowników spędzony na wyjazdach, który po użyciu systemu wizyjnego, można spożytkować bardziej konstruktywnie oraz możliwość późniejszego demontażu urządzenia w celu ponownego użycia, można przyjąć założenie, że automatyzacja tego stanowiska przyniesie zysk. Sposób instalacji systemu przy użyciu komunikacji z użyciem sieci komórkowej jest zdecydowanie droższym przedsięwzięciem. Oprócz kosztów kupna modemu, należy także wziąć pod uwagę cenę transmisji, która w dwuletnim okresie czasu, może być znacząca. Ceny modemów na abonament w sieci Play to około 40zł miesięcznie. Więc, biorąc pod uwagę koszt roczny, łączny wydatek związany z instalacją systemu wizyjnego zdecydowanie przekracza próg opłacalności. Na tym etapie projektu, nie są znane żadne rzetelne informacje na temat możliwego obniżenia ceny przy zamówieniu na tak dużą skalę, dlatego też należy założyć tymczasową nieopłacalność zastosowania tej metody.

### **3.3. Charakterystyka najczęściej występujących usterek kasetonów reklamowych.**

Przed przystąpieniem do tworzenia programu wykrywającego uszkodzenia, należy zastanowić się, jakie usterki występują w kasetonach, oraz które z nich są najbardziej istotne z punktu widzenia klienta, aby zapewnić wysoką skuteczność ich wykrywania przez system wizyjny. Kaseton reklamowy posiada bardzo prostą budowę. Oprócz lica, profilu i zasilacza nie posiada praktycznie żadnych innych elementów, dlatego też każde możliwe uszkodzenie ma związek z jedną z tych części oznaczenia reklamowego.

Profil kasetonu raczej nie ulega usterek. Nawet, jeżeli zdarzy się przypadkowe wgniecenie czy zadrapanie, to z powodu małej widoczności ramy kasetonu, jest to

uszkodzenie o niewielkiej szkodliwości. Dlatego też projekt nie uwzględnia tego typu przypadków.

Uszkodzenia lica to zwykle, w zależności od użytego materiału, pęknięcia lub rozdarcia a także zabrudzenia - między innymi akty wandalizmu w postaci graffiti czy wylana farba podczas odnawiania budynku.



Zdjęcie 3.3 Pęknięcie lica kasetonu. Źródło: materiały własne.



Zdjęcie 3.4 Kaseton z licem zabrudzonym farbą. Źródło: materiały własne.

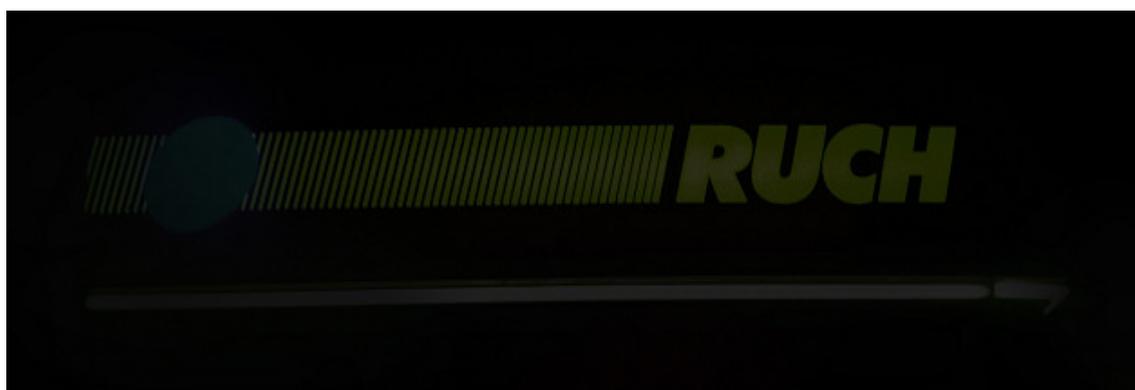
Jak widać na zdjęciach Zdjęcie 3.4 i Zdjęcie 3.3 są to usterki o dość znaczącym wpływie na odbiór promowanych placówek. Pęknięty lub zabrudzony kaseton nadal spełnia swoją rolę jako reklama, nie robi tego jednak optymalnie, ponieważ jej odbiorca może w takim przypadku pomyśleć, że promowana firma jest niechlujna lub nie dba o swój wizerunek. Dlatego też takie uszkodzenia powinny być brane pod uwagę podczas projektowania systemu

technicznego. Te dwa główne rodzaje usterek lica posiadają jedną bardzo ważną różnicę, która może wpłynąć na sposób jej wykrycia przez system. Podczas rozdarcia, zmiana intensywności na kasetonie jest bardzo duża - w miejscu szczeliny oświetlenie wewnętrzne świeci dużo mocniej. Natomiast zabrudzenie nie musi być tak dobrze widoczne. W zależności od koloru, wielkości i miejsca plamy defekt ten może być zauważalny przez system lub nie. W przypadku obiektu ze zdjęcia Zdjęcie 3.4 jest to kaseton o białym tle i kolor farby mocno się odznacza, jednak mogą zdarzyć się sytuacje, w których będzie bardzo ciężko wykryć takiego rodzaju usterkę.

Ostatnim, jednak najważniejszym i najczęstszym uszkodzeniem kasetonu, jest częściowy lub całkowity brak zasilania. Dzieje się tak, ponieważ większość kasetonów, w ramach oszczędności, nie posiada odprowadzania wody ze środka obiektu. W przypadku dużych opadów deszczu poziom wody w środku oznakowania może być tak wysoki, że zaleje zasilacz całkowicie go niszcząc. W takim wypadku oznakowanie reklamowe przestaje spełniać swoje zadanie, zamiast przyciągać w nocy wzrok i promować daną firmę czy produkt, uszkodzony kaseton albo nie świeci wcale albo szpeci placówkę, ukazując tylko część logotypu. Jest to sytuacja, której żaden posiadacz kasetonu nie chce doświadczać, więc wykrycie takich usterek powinno być priorytetem podczas implementacji elementów przetwarzania obrazów do analizy kasetonów.



Zdjęcie 3.5 Kaseton z całkowitą awarią zasilania. Źródło: materiały własne.



Zdjęcie 3.6 Kaseton z częściową awarią zasilania. Źródło: materiały własne.

Przypadek częściowej awarii zasilania przedstawiony na zdjęciu Zdjęcie 3.65 powinien być dość wygodny do wykrycia, zmiany intensywności światła są znaczne i dość rozległe. Natomiast sytuacja, w której zasilanie zostało odcięte całkowicie, zobrazona zdjęciem Zdjęcie 3.56, może sprawiać kłopoty podczas analizy obrazu. Z powodu zupełnego braku oświetlenia w kasetonie, jakość zdjęcia może być niewystarczająca do przetworzenia cyfrowego obrazu (nie będzie możliwe połączenie w pary odpowiadających punktów kluczowych obiektu), aby pozwolić na wykrycie usterki.

Oprócz działań serwisowych firmy produkujące kasetony reklamowe zajmują się również ich odświeżaniem. Z biegiem czasu, z powodu powolnego zużywania się żarówek oraz osadzającego się brudu, oznakowanie reklamowe nie świeci tak mocno jak na początku. W takich przypadkach odpowiedzialna za kaseton firma powinna wyczyścić obiekt oraz ewentualnie wymienić oświetlenie, aby reklama znów działała należycie. Jednak z powodu codziennej obserwacji reklamy, właściciel takiego oznakowania może nie zauważyć różnicy w intensywności oświetlenia. Projektowany wizyjny system techniczny, dzięki porównywaniu kolejnych zdjęć ze zrobionym zaraz po montażu wzorcem, będzie w stanie bardzo łatwo wykryć moment, w którym dany kaseton powinien zostać odnowiony.

### **3.4. Implementacja elementów przetwarzania obrazów do detekcji uszkodzeń w kasetonach reklamowych**

Pierwszą istotnym krokiem wymaganym dla poprawnego działania systemu jest filtracja.. Wszystkie zabiegi filtracyjne w projekcie mają na celu takie przygotowanie zdjęcia obiektu, aby na etapie analizy porównanie go z wzorcem nie sprawiało problemów. Aby to osiągnąć musimy przede wszystkim zadbać o tożsame położenie obu przedmiotów względem siebie. Jest to kluczowe dla poprawnego działania systemu, ponieważ przy zmianach temperatury czy silnym wietrze często doświadczają się ruchu kamer, co może całkowicie zniekształcić wyniki analizy obrazu. Pożądane jest również zniwelowanie wpływu tła a także wyeksponowanie szczegółów w przypadku niewyraźnych zdjęć. Dodatkowo kontrast obu zdjęć powinien być porównywalny, ponieważ podczas końcowej analizy może wpłynąć to na wyniki.

Tak więc do trafnej detekcji usterek badane zdjęcia po zakończonym etapie filtracji muszą spełniać następujące wymagania:

- szczegóły zdjęć powinny być możliwie najbardziej wyeksponowane;
- tło zdjęć nie może wpływać na późniejszą analizę;
- w miarę możliwości powinien zostać zapewniony jednakowy dla każdego zdjęcia kontrast;
- w przypadku, gdy badany obiekt jest zbyt duży i nie mieści się na jednym zdjęciu, obrazy te powinny być „zszyte” w taki sposób, aby po tej operacji rzeczywisty kaseton znajdował swoje odzwierciedlenie w obiekcie badanym widniejącym na zdjęciu;
- położenia obu zdjęć muszą się pokrywać.

Eksperymenty wykazały, iż z racji charakterystyki kasetonów i ich bardzo intensywnego podświetlenia, szczegóły zdjęć nie wymagają większego eksponowania, zwykle zdjęcia są wystarczająco wyraźne dla porównania.

### **3.4.1. Niwelowanie wpływu tła na późniejszą analizę**

Kolejnym bardzo ważnym elementem, który mógłby wpłynąć na pogorszenie skuteczności systemu jest wpływ tła. W przypadku drugiego planu o mocno zróżnicowanym charakterze lub dużej jasności analiza obrazu bez osłabienia jego wpływu może przynieść wiele błędów. Wycinanie przedmiotu można zaimplementować na wiele sposobów. Realizacja tej czynności przy pomocy detektora Canny’ego opiera się na:

- zastosowaniu detektora Canny’ego na uprzednio rozmytym szarościowym obrazie;
- odnalezieniu konturów obiektu przy pomocy funkcji **findContours**;
- wypełnieniu kolorem konturu obiektu wycinanego;
- użyciu przed chwilą otrzymanego obrazu, jako maski do przekopiowania oryginalnego obrazu.

Podczas eksperymentów praktycznych okazało się, że z powodu skomplikowanych wzorów na kasetonach oraz niejednolitego i zróżnicowanego podłoża, zadanie to przysparza wielu problemów. Pierwszy problem pojawia się przy doborze optymalnych progów dla funkcji Canny. Kolejnym zagadnieniem wymagającym uwagi jest zróżnicowanie kasetonów.

System wizyjny w założeniu ma poprawnie wykrywać uszkodzenia na każdym możliwym obiekcie. Niestety, kasetony reklamowe należą do przedmiotów mocno różniących się od siebie i może okazać się wymagane zaprojektowanie odrębnej specyfikacji funkcji findContours dla każdego obiektu.

Głównymi założeniami programu wykrywającego uszkodzenia jest prostota i uniwersalność. Program powinien działać przy możliwie najmniejszej ingerencji człowieka. Wybranie poprawnych progów czy sposobów analizy w elementach przetwarzania obrazu w celu ujednoczenia tła nie jest proste. Dodatkowo, z powodu dużych różnic w wyglądzie, kolorze czy kształcie kasetonów nie ma możliwości stworzenia uniwersalnego programu, który wyeksponuje jedynie badany obiekt ze zdjęcia. Dla każdego nowego kasetonu należałoby stworzyć nowy, unikatowy zestaw ustawień. Dlatego też niemożliwe jest zawarcie wyżej opisanej metody w finalnej wersji projektu.

Kolejnym rozwiązaniem niwelującym wpływ tła może być ręczne wycięcie kształtu kasetonu wzorcowego ze zdjęcia, wypełnienie tła czarnym kolorem i tak otrzymanego obrazu potraktowania jako maski wykorzystywanej w wycinaniu zdjęć badanych. Taka operacja to jednorazowy koszt kilku minut. Po zapewnieniu tożsamości położenia, otrzymana maska będzie idealnie odpowiadać rozmiarom badanemu kasetonowi więc wycięcie tła nie sprawi żadnych dodatkowych kłopotów. Jednak problem tej metody polega na braku uniwersalności, do każdego kasetonu należałoby stworzyć oddzielną maskę co przy dużej ilości placówek będzie bardzo czasochłonne. W celu zwiększenia uniwersalności metody, można tą procedurę przeprowadzić jedynie dla jednego obiektu danego typu, z założeniem, że kamera robi zdjęcia zawsze o tej samej porze oraz z tej samej odległości. W takim przypadku, będzie to bardzo przydatna metoda, jednak należałoby sprawdzić jej skuteczność, co w tym momencie jest niemożliwe z powodu zbyt małej ilości badań.

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że najłatwiejszym, a zarazem najlepszym sposobem na zmniejszenie wpływu tła na analizę usterek jest optymalny dobór pory dnia, w której robione będą zdjęcia używane do analizy. Kasetony reklamowe, z racji ich przeznaczenia, powinny możliwie najbardziej przyciągać wzrok i muszą wyróżniać się z tła. W ciągu dnia jest to zapewnione przy pomocy jaskrawych i intensywnych kolorów, w ciągu nocy - dzięki silnemu oświetleniu obiektu.

Chcąc badać kaseton za dnia - nie dość, że wpływ mocno oświetlonego tła może zaważyć na dużej liczbie błędnych wskazań systemu, to dodatkowo, nie mamy możliwości przeanalizowania sprawności działania oświetlenia, ponieważ nie jest ono widocznie. Z drugiej strony - w nocy, nie dość, że różnica jasności między obiektem, a tłem jest

zdecydowanie większa to dodatkowo, silne oświetlenie kasetonu pozwala dojrzeć wszystkie szczegóły badanego przedmiotu.

Dlatego też, optymalną porą na wykonywanie analizy jest noc, w okolicach godziny pierwszej nad ranem, aby różnice w długości dni i nocy zmieniające się w ciągu roku, nie stanowiły żadnego problemu.

Posiadanie wiedzy o bardzo dużej różnicy między jasnością tła a obiektu prowadzi do wniosku, że możliwe jest stworzenie prawie całkowicie uniwersalnego programu wycinającego tło ze zdjęcia. Metoda ta bazuje po części na opisanej powyżej wykorzystującej detektor Canny’ego. Sposób ten, do stworzenia maski o rozmiarach równych wymiarom kasetonu, wykorzystuje progowanie. Istota tej metody polega na założeniu, iż maską obiektu są piksele o wartościach wyższych niż określony, standardowy próg wynoszący 30% maksymalnej wartości. W ogólnym przypadku taka metoda eksponowania przedmiotu nie jest możliwa do zaimplementowania, jednak wybór nocy, jako pory analizy w połączeniu z intensywnym oświetleniem kasetonu umożliwił tak znaczne uproszczenie sytuacji. Nasuwa się pytanie, czy metoda ta nie ukryje przypadkiem niektórych defektów obiektu. W końcu na kasetonie mogą znaleźć się ciemne punkty lub niedoświetlone obszary, które zostaną pominięte przy progowaniu, co w następstwie doprowadzi do ich ujednoczenia z tłem. Jednak analizując charakterystyki standardowych usterek przedstawionych w rozdziale 3.3 oraz specyfikacji analizy, problemy te nie pojawiły się podczas badań eksperymentalnych.



Zdjęcie 3.7 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 3.8 Obiekt, po zastosowaniu wyżej opisanej metody wycinania tła.

Porównanie zdjęć Zdjęcie 3.7 i Zdjęcie 3.8 dowiodło skuteczności metody. W prosty sposób drugi plan obrazu został ujednoczony. W kolejnym rozdziale metoda ta zostanie dokładniej przeanalizowana, w celu sprawdzenia jej uniwersalność

### 3.4.2. „Zszywanie” zdjęć

Z powodu dużej różnorodności kształtów i rozmiarów kasetonów, mogą zdarzyć się sytuacje, w których kasetony są na tyle duże, że jedno zdjęcie nie odda wystarczająco dokładnie wszystkich szczegółów obiektu. W takim wypadku, aby dopuszczalna była analiza przedmiotu, jako całości, można zastosować metodę „zszywania” obrazów. Jest to dość skomplikowana metoda, która została opisana w rozdziale 2.2.3. Do tego celu można wybrać dwie drogi. Pierwsza wykorzystuje użycie modułu Stitching, biblioteki OpenCV, która automatyzuje wszystkie etapy tej metody, a ingerencja użytkownika polega tylko na dostarczeniu „zszywanych” zdjęć. Drugi sposób wymaga od programisty znajomości funkcji używanych we wszystkich pośrednich krokach. Obie drogi posiadają swoje plusy i minusy.

Metoda pierwsza, zautomatyzowana, z pewnością jest bardzo prosta w implementacji. Cały skomplikowany proces zawiera się w dwóch liniach kodu, a znajomość teorii procesu nie jest obligatoryjna do poprawnego działania. Użytkownik może, ale nie musi precyzować wszystkich parametrów funkcji. W celu optymalizacji algorytmu do wykonywania wyspecjalizowanych zadań, wymagane jest zaznajomienie się z potokiem modułu Stitching, jednak dla standardowych operacji automatyzacja ta jest bardzo przydatna. Bardzo dużą dodatkową zaletą użycia modułu Stitching jest wkomponowany w proces przetwarzania algorytm dokonujący blendowania koloru granicy połączenia dwu obrazów oraz wygładzania jego krawędzi. Implementacja funkcji „zszywania” z użyciem modułu Stitching w C++ wygląda następująco:

```
//przygotowanie zdjęć do formatu kompatybilnego z funkcja stitch
vector<Mat> zdjecia;
zdjecia.push_back(zdjecie1);
zdjecia.push_back(zdjecie2);
Mat zdjeciaSklejone;
//stworzenie zmiennej klasy Stitcher o standardowych parametrach
Stitcher zszywanie = Stitcher::createDefault(false);
//zszycie zdjęć
zszywanie.stitch(zdjecia,zdjeciaSklejone);
```

Przedstawiony kod programu jest bardzo krótki, zainicjowane jedynie cztery zmienne dowodzą, że metoda ta jest bardzo prosta w użyciu. Jedyne wymagane dane potrzebne do

działania funkcji to „zszywane” zdjęcia, co zdecydowanie ułatwia używanie tego algorytmu. Jednak przy bardziej złożonej analizie często wymagane jest większe wyspecjalizowane podejście do tej metody. Moduł ten świetnie radzi sobie z tworzeniem panoramy zdjęć, ale w przypadku bardziej skomplikowanej sytuacji, w której dokładność jest istotnym elementem, lepszym rozwiązaniem jest użycie własnego kodu, zdecydowanie bardziej rozbudowanego.

Przykładową metodą stosowaną do otrzymania „zszytych” zdjęć jest użycie sekwencji algorytmów, wykorzystując przy tym deskryptor SURF. Funkcje te i ich kolejność została przedstawiona w rozdziale 2.2.2, z tego względu zostanie pominięty bardziej szczegółowy opis, a zostanie opisana implementacja tej metody w C++(Kod.1):

```
//stworzenie deskryptora SURF
SurfFeatureDetector detector(700);
vector< KeyPoint > keypoints_img1, keypoints_img2;
//odnalezienie punktów kluczowych ze zdjęć szarościowych
detector.detect( gray_img1, keypoints_img1 );
detector.detect( gray_img2, keypoints_img2 );
//Obliczenie deskryptorów i łączenie w pary punktów kluczowych
SurfDescriptorExtractor extractor;
Mat descriptors_img1, descriptors_img2;
extractor.compute( gray_img1, keypoints_img1, descriptors_img1 );
extractor.compute( gray_img2, keypoints_img2, descriptors_img2 );
FlannBasedMatcher matcher;
vector< DMatch > matches;
matcher.match( descriptors_img1, descriptors_img2, matches );
double max_dist = 0; double min_dist = 100;
// Wyliczenie maksymalnej i minimalnej odległości pomiędzy sparowanymi punktami
kluczowymi.
for( int i = 0; i < descriptors_img1.rows; i++ )
{ double dist = matches[i].distance;
if( dist < min_dist ) min_dist = dist;
if( dist > max_dist ) max_dist = dist;}
// Wstępna filtracja zbyt oddalonych od siebie sparowanych punktów kluczowych
vector< DMatch > good_matches;
for( int i = 0; i < descriptors_img1.rows; i++ )
{ if( matches[i].distance < 3*min_dist )
{ good_matches.push_back( matches[i]); }}
//Stworzenie zmiennych i przypisanie do nich wyselekcjonowanych punktów kluczowych
vector< Point2f > points_img1;
vector< Point2f > points_img2;
for( int i = 0; i < good_matches.size(); i++ ){
points_img1.push_back( keypoints_img1[ good_matches[i].queryIdx ].pt );
points_img2.push_back( keypoints_img2[ good_matches[i].trainIdx ].pt );}
// Odnalezienie macierzy homografii
Mat homografia = findHomography( points_img1, points_img2, CV_RANSAC );
// Przekształcenie perspektywiczne obrazu drugiego według macierzy homografii
Mat result;
warpPerspective(image2,result,homografia,Size(image1.cols*2+image2.cols,image1
.rows));
// "Zszytanie" obu zdjęć
Mat zszyte(result,Rect(0,0,image2.cols,image2.rows));
image1.copyTo(zszyte);
```

Analiza kodu programu prowadzi do wniosków, iż to co wyróżnia ten sposób od metody wykorzystującej moduł Stitching to wymóg znajomości działania każdej poszczególnej funkcji. Jest to z pewnością utrudnienie w pracy, jednak niewielkie, ponieważ wiedza ta przydaje się do wielu innych czynności z zakresu przetwarzania obrazu. Zaletą jest na pewno fakt możliwości dostępu do wszystkich obliczonych parametrów zdjęć, takich jak opis punktów kluczowych czy macierz homografii, które później mogą zostać wykorzystane do innych celów. Oprócz tego metoda ta pozwala na większy wybór rodzaju deskryptora opisującego punkty kluczowe, co w przypadku naszego projektu jest bardzo istotne, gdyż podczas wdrożeniu programu może okazać się, że deskryptor SURF jest zbyt wymagający dla tak słabej jednostki obliczeniowej, jaką jest Raspberry Pi ( dla porównania, moduł Stitching, umożliwia wybór tylko pomiędzy deskryptorem SURF lub ORB) i może wystąpić konieczność zastąpienia go szybszym, np. często używanym w smartphonach, algorytmem LAZY.

Jednak, porównanie obu metod doprowadziło do wniosku, iż pomimo znacznego uproszczenia przy projektowaniu programu, wyborem umożliwiającym lepsze działanie będzie metoda druga, niezautomatyzowana, wykorzystująca deskryptor SURF. Poprawnie zaprojektowana działa sprawniej i z większą uniwersalnością. Dodatkową zaletą jest możliwość późniejszego użycia punktów kluczowych zdjęcia do innych przekształceń. Wadą tej funkcji natomiast jest niezrozumiałe i wybiórcze działanie. Podczas eksperymentów badane zdjęcia, przy pewnej rozdzielczości, były poprawnie przetwarzane przez algorytm, natomiast po zmianie rozdzielczości na mniejszą w funkcji pojawiały się błędy. Dodatkowo zdarzały się sytuacje, w których zdjęcia, pomimo poprawnego zszycia, były ucinane na końcach, co powodowało utratę informacji na temat badanego obiektu. Oprócz tego, zakładając, że analiza obiektu będzie opierała się na porównywaniu zdjęć robionych w przybliżeniu, w tej samej pozycji, przez tą samą kamerę, różnice w nasyceniu koloru będą wzajemnie kompensowane, więc blendowanie kolorów oraz wygładzanie krawędzi nie będzie niezbędne. Poniżej przedstawione zostało działanie używanego w projekcie algorytmu.



Zdjęcie 3.9 Lewa część „zszywanego” obiektu. Źródło: materiały własne.

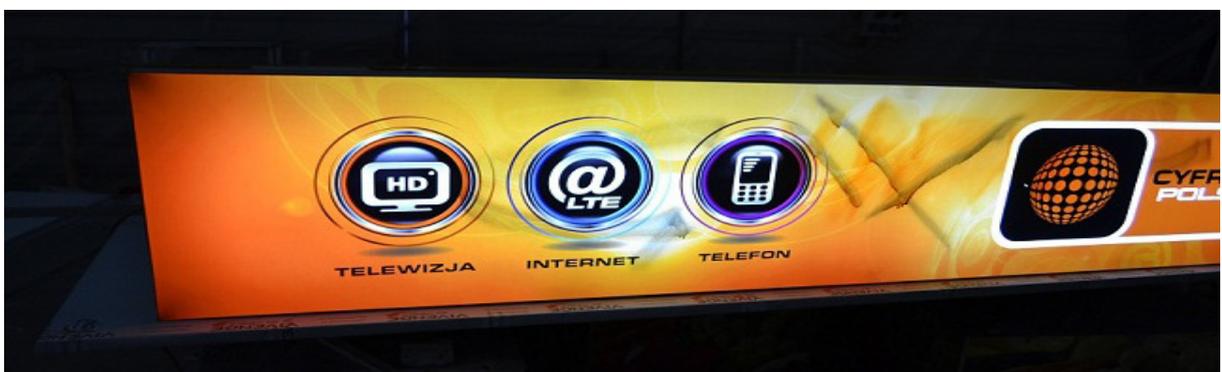


Zdjęcie 3.10 Prawa część „zszywanego” obiektu. Źródło: materiały własne.



Zdjęcie 3.11 Obiekt po „zszyciu”. Źródło: materiały własne.

Jak widać na zdjęciu Zdjęcie 3.11, obiekt został „zszyty” poprawnie. Na obrazie wynikowym nie da się zauważyć żadnych nierównych krawędzi czy nagłych zmian kolorów. W prawym dolnym rogu widać czarne pole - jego występowanie jest spowodowane uniwersalnością metody, jednak w żaden sposób nie obniża ono przydatności zdjęcia do późniejszej analizy obrazu. Podczas robienia zdjęć częściowych Zdjęcie 3.9 i Zdjęcie 3.10 zostały spełnione wszystkie wymagane do „zszycia” warunki, co w następstwie umożliwiło idealne połączenie obiektu. Poniżej został przedstawiony przykład, w którym, wymienione w rozdziale 2.2.2 wymagania, nie zostały spełnione.



Zdjęcie 3.12 Lewa część „zszywanego” obiektu. Źródło: materiały własne.



Zdjęcie 3.13 Prawa część „zszywanego obiektu”. Źródło: materiały własne.



Zdjęcie 3.14 „Zszyty” obiekt.

Zdjęcie 3.14 przedstawia obiekt połączony niepoprawnie. Oprócz różnicy w kontraście, spowodowanej odmienną jasnością kasetonu widzianego pod różnymi kątami, która jest normalna i jej szkodliwy wpływ na późniejszą analizę obrazu jest możliwy do skompensowania, występuje także niedokładność w dopasowaniu krawędzi która może znacznie zniekształcić końcowy wynik wykrywania uszkodzeń przez program. Niepoprawne dopasowanie krawędzi zostało spowodowane znacznymi różnicami w perspektywie podczas robienia zdjęć Zdjęcie 3.12 i Zdjęcie 3.13. W efekcie, pomimo identycznych rzeczywistych wymiarów, podczas analizy uszkodzeń, kształt kasetonu badanego na zdjęciu nie będzie porównywalny z wzorcem, co może doprowadzić do dużych problemów w czasie analizy uszkodzeń.

### 3.4.3. Zapewnienie tożsamości położenia obiektów

Warunkiem koniecznym do poprawnego działania zaprojektowanego systemu jest identyczne położenie obiektu wzorcowego i badanego. Jest to podyktowane tym, iż w celu wykrycia usterki obiektu badanego, musimy porównać go z wzorcem. Prawdopodobnie istnieje możliwość przeprowadzenia analizy wystąpienia uszkodzenia, gdy położenie obiektów nie jest tożsame, jednak przy spełnieniu wyżej postawionego warunku, metody

porównawcze są szybkie, łatwe i skuteczne. W przypadku idealnym, gdy kamera zawsze wisi w jednym miejscu i jest ciągle skierowana w ten sam punkt obiektu, warunek ten będzie automatycznie spełniony. Jednak w normalnym użytkowaniu koniec ramienia, do którego przymocujemy kamerę będzie zmieniał swoje położenie. Jest to spowodowane m.in. zmianą temperatury, wiatrem, zużyciem materiału. Z tego względu konieczne jest zaimplementowanie w projekcie algorytmu, który będzie zapewniał tożsamość położenia obiektów na zdjęciach.

Z racji wykorzystania w projekcie, podczas etapu „zszywania” zdjęć, algorytmu wykorzystującego deskryptor SIFT, przy zapewnianiu tożsamości położenia, posłużymy się tą samą metodą. Jest to o tyle dobre rozwiązanie, że do tego celu możemy wykorzystać dokładnie ten sam kod źródłowy (Kod.1). Jediną różnicą pomiędzy etapami jest brak kroku wykonującego „zszycie” obrazów w algorytmie odpowiedzialnym za zagwarantowanie identycznego położenia obiektów. Jest to możliwe, ponieważ w przypadku dwu zdjęć, które zawierają ten sam obiekt, funkcja `warpPerspective` wykorzystując macierz homografii przekształca zdjęcie badane tak, aby położenia obiektów były identyczne.



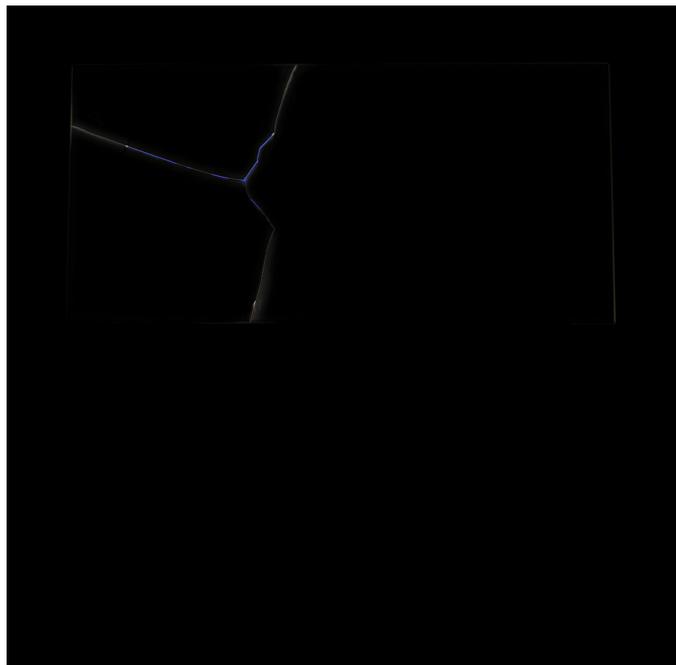
Zdjęcie 3.15 Badany obiekt.. Źródło: materiały własne.



Zdjęcie 3.16 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 3.17 Obiekt badany z zapewnioną tożsamością położenia.



Zdjęcie 3.18 Różnica obiektu wynikowego z wzorcowym.

Dokładniejsza analiza tego przypadku zostanie przedstawiona w podrozdziale 4.2

Na zdjęciu 3.17 pokazano obiekt badany po zapewnieniu tożsamości położenia. Zdjęcie 3.18 obrazuje wynik w sposób bardziej szczegółowy, wykonując działanie odejmowania kasetonu badanego od wzorca, analizując ten wynik, można zauważyć obramowanie obiektu, jednak różnica pomiędzy jasnością obramowania, a jasnością usterki jest tak duża, że nie powinno to przysporzyć problemu przy wykrywaniu uszkodzenia.

### 3.4.4. Analiza danych

W celu analizy obrazów możliwe jest użycie wielu metod, ale rozważania nad rozwiązaniami w projekcie zostały ograniczone jedynie do kilku stosunkowo prostych, ponieważ sama istota analizy nie jest skomplikowana. Aby dowiedzieć się, czy jeden obraz różni się od drugiego nie jest wymagane użycie złożonych algorytmów.

Funkcje brane pod uwagę podczas projektowania to:

- porównywanie samych obrazów przy użyciu funkcji **compare**,
- porównanie histogramów,
- odjęcie od siebie dwóch obiektów w celu zobrazowania różnic w obrazach.

Aby jakakolwiek analiza była możliwa, wymagane jest wcześniejsze zapewnienie akwizycji wzorcowego zdjęcia obiektu. Dlatego też, zaraz po montażu kasetonu w placówce, jeden z pracowników będzie zobligowany do wykonania w systemie technicznym pierwszego zdjęcia obiektu, które następnie będzie przechowywane, jako wzorzec, do którego później porównywane będą kolejne zdjęcia kasetonów.

Pierwszym pomysłem na rozwiązanie problemu wykrywania uszkodzeń była metoda porównywania obrazów, piksel po pikselu. Istota funkcji polegała na założeniu, że jeżeli oba obrazy są identyczne, to przy porównaniu wartości pikseli, macierz wyjściowa funkcji będzie zawierać same zera. W przypadku wystąpienia uszkodzenia, w macierzy wynikowej zwiększy się liczba jedynek. Korzystając następnie z funkcji `countNonZero`, która liczy występujące w zbiorze wartości niezerowe, można w bardzo prosty sposób wykryć usterkę.

Niestety, podczas eksperymentów, okazało się, że opisana metoda nie działa poprawnie w wizyjnym systemie technicznym. W macierzy wynikowej pojawia się duża liczba wartości niezerowych przy analizie nieuszkodzonego obiektu, spowodowana minimalnymi różnicami wartości odpowiadających sobie punktów. Funkcja `compare` wykrywa każdą, nawet najmniejszą różnicę w wartości pikseli, co jest argumentem przeciw użyciu jej w finalnej wersji projektu, ponieważ większość tych niezgodności jest spowodowana niedokładnością filtracji czy sprzętu pomiarowego, a nie samym uszkodzeniem.

Kolejnym sposobem na wykrycie uszkodzeń kasetonów jest porównanie histogramów. Z założenia, metoda ta powinna być bardziej dokładna od pierwszego sposobu analizy. W istocie, funkcja ta wyznacza korelacje pomiędzy dwoma histogramami, gdzie metoda jej wyliczania zależy od parametru funkcji `compareHist` zadanego przez użytkownika. Jest to

korzystne, ponieważ nawet w przypadku lekkiego przesunięcia obiektu względem wzorca, liczba poszczególnych wartości na obrazie pozostanie niezmienną.

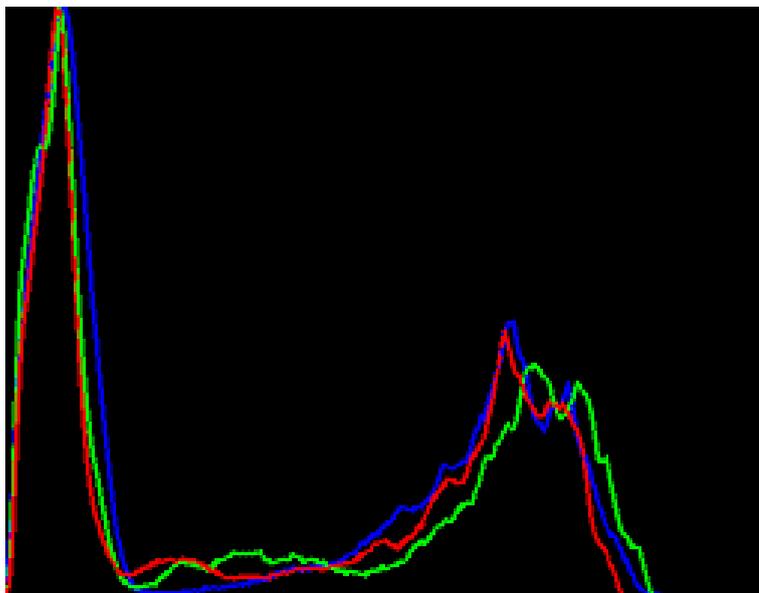
Eksperymenty pokazały, że jest to metoda poprawna, przy jej użyciu można wyciągać prawidłowe wnioski, jej zaletą jest z pewnością możliwość wykrycia niewyróżniającej się znacząco z tła, dość dużej plamy na kasetonie. Jednak w przypadku małych, punktowych uszkodzeń nie ma możliwości wykrycia usterki.



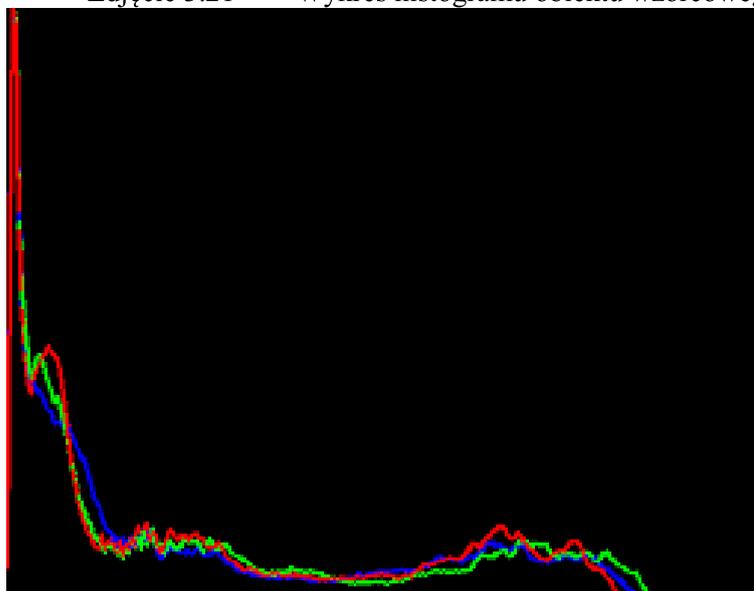
Zdjęcie 3.19 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 3.20 Obiekt uszkodzony. Źródło: materiały własne.



Zdjęcie 3.21 Wykres histogramu obiektu wzorcowego



Zdjęcie 3.22 Wykres histogramu obiektu uszkodzonego

W przypadku awarii, histogram na zdjęciu 3.22 całkowicie zmienia swój kształt w porównaniu do histogramu ze zdjęcia 3.21, ma to swoje odzwierciedlenie w znacznej zmianie współczynnika korelacji obliczonego za pomocą funkcji `compareHist`. W przypadku porównywania dwóch sprawnych kasetonów, współczynnik ten jest bliski jedynce, natomiast, gdy badany obiekt uległ awarii (Zdjęcie 3.19 i Zdjęcie 3.20) wartość korelacji wyniosła 0,57 co jest wynikiem poprawnym.

Ostatnią metodą analizującą obraz jest odjęcie od siebie obu obrazów. Jest to sposób o tyle dobry, że jest w stanie wykryć nawet małe uszkodzenia kasetonu. Jedynym problemem jest moment w którym położenie obiektów nie jest idealnie porównywalne. W przypadku, gdy przedmiot badany jest przesunięty względem wzorcowego nawet o jeden piksel, wynikiem różnicy będą bardzo niewygodne podczas analizy krawędzie. Jednak są to wartości tak małe,

że można zastosować progowanie na wyniku odejmowania, które całkowicie usunie niechciane błędy. Po progowaniu, należy zastosować funkcję countNonZero, w celu wykazania rozległości występującej usterki.

Podczas eksperymentów okazało się, że jest to bardzo skuteczna metoda do odnajdywania wszystkich uszkodzeń kasetonu o dużej zmianie jasności. Odjęcie obiektu wzorcowego od badanego ukazuje usterki o większej niż normalna jasności kasetonu, natomiast sytuacja odwrotna eksponuje uszkodzenia ciemniejsze. Co w następstwie daje bardzo przydatny dla poprawnego działania systemu algorytm.

Aby wybrać poprawne metody do analizy obrazu, należy wziąć pod uwagę, które defekty kasetonu są najbardziej istotne. Najważniejszym defektem kasetonu jest całkowity lub częściowy brak zasilania. W przypadku, gdy reklama przestaje świecić, traci wszelką przydatność, nie spełnia swojego zadania, gdyż nie przyciąga wzroku. Dlatego też sposób wykrywania usterki musi być nieomylny w wypadku, gdy oświetlenie kasetonu zawiodło. Dodatkowo system musi reagować na usterki rzadziej spotykane, takie jak pęknięcie kasetonu, oblanie farbą, wgniecenie itp.

Po analizie występujących usterek i ich szkodliwości można wysnuć wnioski, że optymalną metodą analizy uszkodzeń będzie połączenie analizy zmiany wyników podczas porównania histogramu oraz zastosowania różnicy obiektu wzorcowego i badanego. Razem te dwie metody skutecznie analizują uszkodzenia powodujące duże oraz małe zmiany w jasności kasetonu.

## **ROZDZIAŁ 4**

### **BADANIA EKSPERYMENTALNE I ANALIZA OTRZYMANYCH WYNIKÓW**

#### **4.1. Opis i zasadność badań eksperymentalnych**

Aby wizyjny system techniczny wykrywający usterki na kasetonach reklamowych był przydatny, musi bezbłędnie podejmować decyzje w przypadku wystąpienia uszkodzenia. Do osiągnięcia tego celu program musi być na tyle dokładny, aby wykryć każdą najmniejszą usterkę, jednak jego czułość nie może powodować dużej liczby fałszywie pozytywnych wskazań. Jediną możliwą drogą umożliwiającą poprawne skalibrowanie wrażliwości systemu na zmiany są badania eksperymentalne.

Testy praktyczne, którym zostanie poddany program sprawdzą jego zachowanie podczas wystąpienia wszystkich występujących uszkodzeń przy różnych położeniach kamery, aby możliwie najbardziej zbliżyć się do warunków praktycznych, w których w przyszłości, znajdzie się system techniczny. Najbardziej istotna usterka, czyli częściowa lub całkowita awaria zasilania, zostanie zbadana bardziej szczegółowo, a mianowicie, skuteczność jej wykrywania będzie skalibrowana na podstawie czterech różnych przypadków. Mniej ważne uszkodzenia, za które uznaje się zaplamienie i pęknięcie lica, a także znaczne wyeksploatowanie kasetonu, zostaną przebadane jednokrotnie. Dodatkowym przypadkiem będzie sprawdzenie poprawnego działania programu w przypadku braku usterki, a także sprawdzenie poprawności analizy obrazu w przypadku awarii zasilania liter przestrzennych.

Tak mała liczba eksperymentów jest podyktowana bardzo dużym problemem w akwizycji oznakowania reklamowego przed i po usterce. Awarie kasetonów mają miejsce w całej Polsce, co więcej serwis oznakowania, z powodu istotnych walorów reklamowych, jest wykonywany w jak najkrótszym czasie, następstwem tego są duże utrudnienia w dojazdach do placówek w celu osobistej akwizycji zdjęć. Aby temu zaradzić został pozyskany dostęp do bazy zdjęć reklamacyjnych firmy zajmującej się serwisem tych urządzeń. Niestety, dane te okazały się niewystarczające do przeprowadzenia dużej liczby badań, ponieważ pierwsze zdjęcia wykonywane były zaraz po montażu, kolejne natomiast w momencie wystąpienia usterki, zwykle dzielił je duży odstęp czasu, więc fotografie wykonywały inne osoby, z całkowicie różnych perspektyw, często, ucinając fragmenty obiektów, tym samym uniemożliwiając poprawną analizę obrazów.

Zdjęcia zostały zrobione z różnych wysokości, w każdym badanym przypadku, obiekt badany jest przesunięty względem wzorca, ma to na celu zamodelowanie realnych warunków akwizycji zdjęć, w których kamera będzie zmieniać swoje położenie w zależności od pogody, wiatru i zużycia materiału.

## 4.2. Badania eksperymentalne i analiza wyników

### 4.2.1. Pęknięcie/przerwanie kasetonu

Pierwszym badanym przypadkiem jest pęknięcie kasetonu. Usterka ta może być dość trudna do wykrycia z racji jej małego wpływu na zmianę współczynnika korelacji, na zdjęciu wygląda jak niewielka szczelina o większym natężeniu światła, a związane z nią kłopoty mogą być spowodowane zbyt małą dokładnością programu. Zdjęcie wzorcowe to zdjęcie Zdjęcie 3.15, natomiast badane, na którym występuje usterka to zdjęcie Zdjęcie 3.16.

Z racji przedstawienia już wszystkich związanych z działaniem systemu obrazów, zaprezentuję jedynie wyniki liczbowe i wnioski.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.9252	57	1059	Prawdopodobne znaczne pęknięcie/rozerwanie kasetonu

Tablica 4.1 Wyniki analizy liczbowej.



Zdjęcie 4.1 Wykryte przez program piksele jaśniejsze

Analiza zdjęcia 4.1 oraz wyników analizy liczbowej z tablicy 4.1. prowadzi do wniosku, iż program zareagował poprawnie, usterka została wykryta. Zgodnie z założeniem, takie

uszkodzenie nie wpływa znacznie na współczynnik korelacji histogramów, jednak jest dość proste do rozpoznania przy pomocy analizy jaśniejszych od badanego obiektu pikseli. Podczas badania pojawiły się również piksele ciemniejsze od wzorca, niestety, są to błędy spowodowane niepoprawną analizą, jednak ich liczba była na tyle mała, że została przez program zignorowana.

#### 4.2.2. Zaplamienie kasetonu

Następnym przypadkiem jest zaplamienie kasetonu. To uszkodzenie najtrudniejsze do wykrycia, ponieważ różnice koloru pikseli w przebarwionych miejscach może być tak mała, że program nie zakwalifikuje ich, jako różnych od wzorcowych.



Zdjęcie 4.2 Obiekt wzorcowy. Źródło: materiały własne.

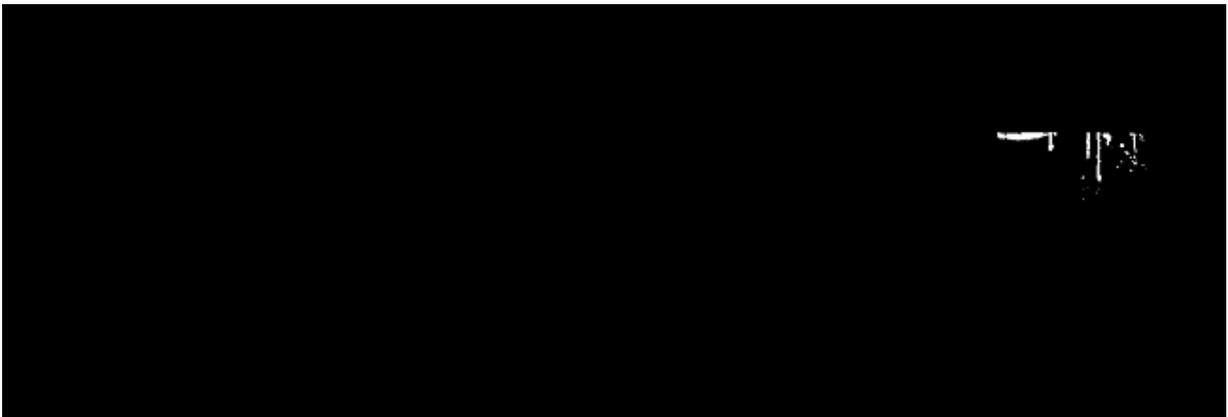


Zdjęcie 4.3 Obiekt badany. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.990457	1328	0	Prawdopodobna ciemna plama na kasetonie

Tablica 4.2 Wyniki analizy liczbowej.

Jak widać, rozmiar uszkodzenia jest niewielki, a kolor plamy jest podobny do ramy kasetonu. Co więcej na zdjęciu wzorcowym 4.2 widać jasną krawędź, która nie występuje na zdjęciu 4.3, a to uniemożliwia jej kompensację przez program. Pomimo możliwych błędów podczas analizy obrazu, system techniczny podjął dobrą decyzję odnośnie uszkodzenia co przedstawia tablica 4.2. . Można się zastanawiać czy aby na pewno ciemniejsze od wzorca piksele to w istocie uszkodzenie, a nie, spowodowane niedokładnością analizy błędy, w tym celu, poniżej przedstawiony został wynik odejmowania obrazu badanego od wzorcowego po wykonaniu progowania, załączonego w kodzie programu.



Zdjęcie 4.4 Wynik różnicy obiektu badanego od wzorcowego.

Analiza zdjęcia 4.4 całkowicie wyjaśnia proces analizy. Piksele ciemniejsze wykryte przez system to w rzeczywistości usterka, a nie błąd pomiarowy. Dlatego też, można uznać, że program poprawnie wykrywa zabrudzenia kasetonu.

### **4.2.3. Znaczne wyeksploatowanie kasetonu**

Następnym badanym przypadkiem jest wyeksploatowanie kasetonu. Nie jest to prawdziwe uszkodzenie oznakowania, jednak zmienia sposób odbierania reklamy, co wymaga od programu rozpoznawania również i takich zmian.



Zdjęcie 4.5 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 4.6 Obiekt wyeksploatowany. Źródło: materiały własne.

Różnica jasności kasetonu pomiędzy zdjęciem 4.5, a 4.6 wydaje się być znikoma, jednak czułość na eksploatację można w bardzo prosty sposób dobrać zmieniając próg wykrywalności w kodzie programu, więc został przebadany przypadek, który zapewni poprawne działanie dla każdego stopnia eksploatacji.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.798922	0	0	Możliwe zużycie kasetonu

Tablica 4.3 Wyniki analizy liczbowej.

Analiza wyników liczbowych z tablicy 4.3 pozwala sądzić, że procedura wykrywania znacznej eksploatacji kasetonów została przeprowadzona bezbłędnie. Współczynnik korelacji dość znacznie odbiega od jego maksymalnej wartości, co więcej, nie zostały wykryte żadne błędne piksele jaśniejsze czy ciemniejsze, mogące sugerować inny rodzaj usterki. Program bardzo skutecznie analizuje obiekt pod kątem jego eksploatacji.

#### 4.2.4. Częściowa awaria zasilania kasetonu

Kolejnym uszkodzeniem poddany eksperymentom jest częściowa awaria zasilania. Z racji jej charakterystyki, powinno to być łatwa do wykrycia usterka, gdyż w przypadku wystąpienia, wprowadza znaczne zmiany intensywności na dużym obszarze kasetonu. Z racji największego wpływu tego uszkodzenia na odbiór reklamy przez konsumenta, przypadek ten zostanie rozpatrzony w czterech różnych przypadkach.



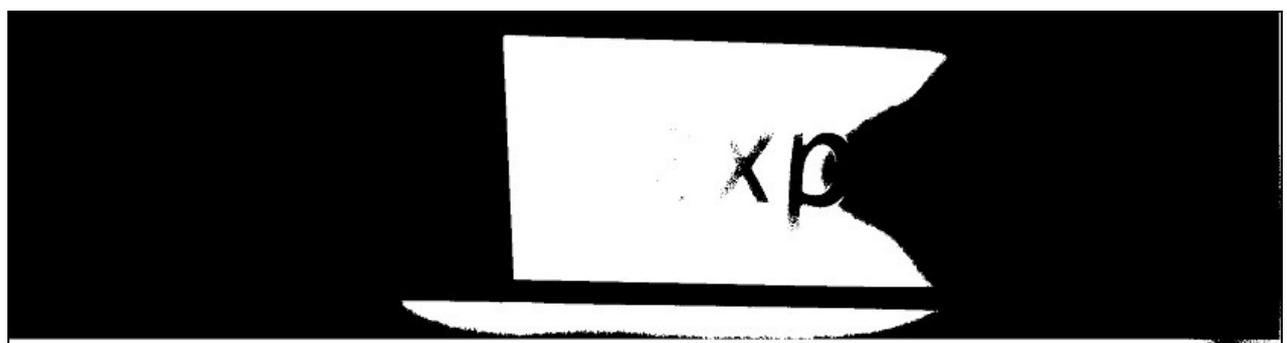
Zdjęcie 4.7 Obiekt wzorcowy Źródło: materiały własne.



Zdjęcie 4.8 Obiekt badany z częściową awarią zasilania na środku kasetonu. Źródło: materiały własne.

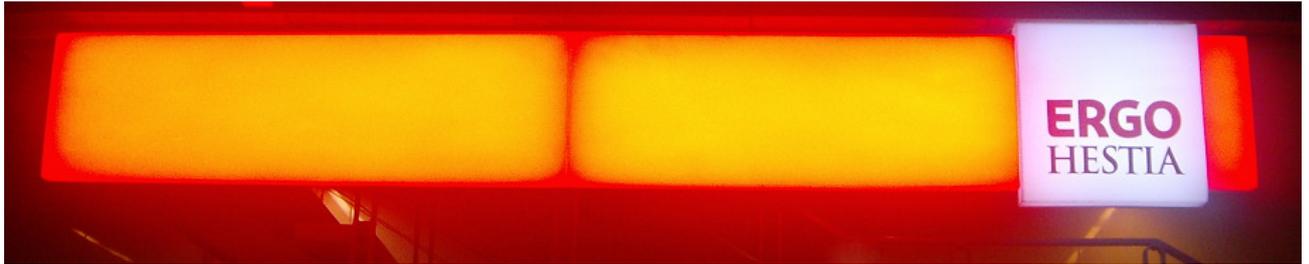
Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.526297	309066	0	Prawdopodobna częściowa awaria zasilania kasetonu

Tablica 4.4 Wyniki analizy liczbowej.

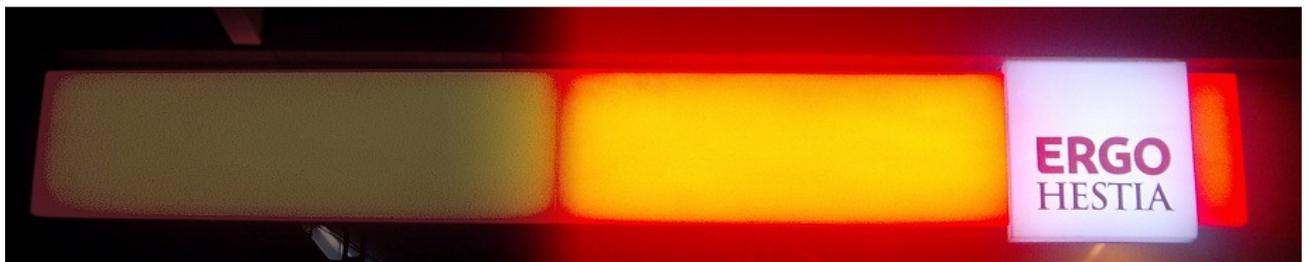


Zdjęcie 4.9 Wykryte przez program piksele ciemniejsze.

Po analizie zdjęcia nr 4.9, można stwierdzić, że program poradził sobie z wykryciem uszkodzenia. Na co należy zwrócić uwagę to dość duży biały pas, który został sklasyfikowany, jako usterka, będący tak naprawdę błędem. Jest on widoczny z powodu przesunięcia kamery, zdjęcie wzorcowe 4.7 obejmuje dużo większą część białej poświaty pod kasetonem niż zdjęcie badane 4.8. Aby upewnić się, że program nie będzie interpretował zdjęcia błędnie analizując inne przypadki, w dalszej części pracy zostanie przeprowadzone badanie sytuacji gdzie oba ujęte na zdjęciach obiekty działają bezawaryjnie.



Zdjęcie 4.10 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 4.11 Obiekt badany z awarią zasilania na połowie kasetonu. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.526297	309066	0	Prawdopodobna częściowa awaria zasilania kasetonu

Tablica 4.5 Wyniki analizy liczbowej.



Zdjęcie 4.12 Wykryte przez program piksele ciemniejsze.

Na zdjęciu 4.12 pokazano efekt działania algorytmu detekcji usterek, a w tabeli 4.5 wyniki analizy liczbowej, które wskazują na prawidłowe działanie systemu.



Zdjęcie 4.13 Zdjęcie wzorcowe. Źródło: materiały własne.



Zdjęcie 4.14 Obiekt badany z częściową awarią zasilania kasetonu. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.680962	20308	0	Prawdopodobna częściowa awaria zasilania kasetonu

Tablica 4.6 Wyniki analizy liczbowej.



Zdjęcie 4.15 Wykryte przez program ciemniejsze od wzorca piksele.

W tym przypadku również zachowanie programu było prawidłowe. Analiza zdjęcia 4.15 oraz tablicy 4.6 dodatkowo upewnia, że nie zostały wykryte żadne błędne dane.

Litery przestrzenne to bardzo ciekawy, wyjątkowy przypadek. Charakterystyka takiego rodzaju oznakowania reklamowego jest podobna do kasetonów jednak nie identyczna, dlatego też wymagane jest sprawdzenie skuteczności programu również w przypadku tego oznakowania.



Zdjęcie 4.16 Wzorcowe litery przestrzenne. Źródło: materiały własne.



Zdjęcie 4.17 Badane litery przestrzenne z awarią zasilania. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.971585	125484	340	Prawdopodobna ciemna plama na kasetonie

Tablica 4.7 Wyniki analizy liczbowej.



Zdjęcie 4.18 Wykryte przez program piksele ciemniejsze niż wzorcowe.

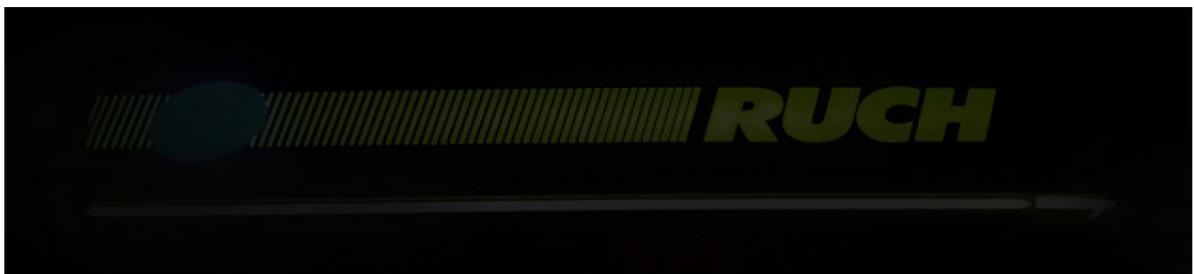
Analizując komunikat programu zawarty w tablicy 4.7. można zauważyć, iż system poprawnie sklasyfikował uszkodzenie liter przestrzennych, jako awarie powodującą przyciemnienie oznakowania, jednak błędnie uznał, że jest to plama. Wyjaśnieniem tej sytuacji jest bardzo mały wpływ uszkodzenia na współczynnik korelacji, pomimo dość

dużych widocznych na zdjęciach 4.16 oraz 4.17 różnic w natężeniu światła, histogramy były prawie identyczne, co spowodowało złe zakwalifikowanie usterki przez system. Dodatkowo, przypadek liter przestrzennych, z powodu dużej liczby błędnie zakwalifikowanych pikseli, zostanie w dalszej części pracy pogłębiony o sytuację gdzie usterka nie występuje, aby sprawdzić poprawność działania programu.

Ostatnim eksperymentalnie zbadanym przypadkiem uszkodzenia kasetonu jest całkowita awaria zasilania. Sprawdzenie poprawności analizy w takiej sytuacji jest o tyle ważne, że podczas przetwarzania obrazu może pojawić się problem z zapewnieniem tożsamości położenia obiektów. Z powodu bardzo dużej różnicy jasności dwóch zdjęć, może dojść do sytuacji gdzie nie ma żadnych symetrycznych punktów kluczowych, dzięki którym można następnie zapewnić tożsamość położenia obu kasetonów na zdjęciach.



Zdjęcie 4.19 Obiekt wzorcowy. Źródło: materiały własne.



Zdjęcie 4.20 Obiekt badany, w którym wystąpiła całkowita awaria zasilania. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.347005	899505	0	Całkowita awaria zasilania lub błąd w działaniu algorytmu

Tablica 4.8 Wyniki analizy liczbowej.



Zdjęcie 4.21 Wykryte przez program ciemniejsze od wzorca piksele.

Analizując wyniki liczbowe z tabeli 4.8. z ostatniego badania można dojść do wniosku, że program zadziałał poprawnie. Niestety, pierwsza próba zakończyła się niepowodzeniem, z powodu źle dobranych progów w funkcji zapewniającej tożsamość położenia wynik został całkowicie zniekształcony, jednak, pomimo tego, komunikat systemu był adekwatny do sytuacji. Dlatego też, można założyć poprawne wykrywanie tego typu usterek przez program.

#### 4.2.5. Analiza poprawności działania systemu podczas bezawaryjnej pracy kasetonu

Poniżej zostaną przedstawione dwa dodatkowe przypadki, w których usterka nie występuje, jednak podczas badań wystąpiły w tych sytuacjach niepokojące błędy, zbyt duża liczba pikseli została uznana za różne od wzorca.

Pierwszy z nich dotyczy kasetonu Alior Bank, obraz badany ze zdjęcia 4.22 będzie porównywany z wzorcem ze zdjęcia Zdjęcie 4.13 .



Zdjęcie 4.22 Obraz badany. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.992605	45105	0	OK

Tablica 4.9 Wyniki analizy liczbowej.



Zdjęcie 4.23 Wykryte przez program ciemniejsze od wzorca piksele.

Analizując tablicę 4.9 oraz zdjęcie 4.23 można dojść do wniosku, iż pomimo dużej liczby błędnie wykrytych przez program ciemniejszych pikseli od wzorca, system nadal poprawnie dokonał oceny sytuacji. Dzieje się tak, dlatego, że niezgodności w zdjęciach Zdjęcie 4.13 oraz 4.22 miały bardzo mały wpływ na korelacji dwu histogramów, co w następstwie pozwoliło na poprawne zakwalifikowanie sytuacji, jako poprawnej pracy. Powodem dużej liczby ciemniejszych od wzorca pikseli wykrytych przez system jest znaczne przesunięcie obiektów wzorcowego i badanego względem siebie, a także specyficzny sposób oświetlenia kasetonu. Z powodu różnicy w zawartych szczegółach na zdjęciach, bardzo jasne dolne oświetlenie oznakowania nie zostało całkowicie skompensowane.

Ostatni przypadek zbada poprawność działania systemu podczas analizy liter przestrzennych, które działają nie wykazując znaków uszkodzenia. Zdjęciami wzorcowym i badanym będą:



Zdjęcie 4.24 Obraz wzorcowy. Źródło: materiały własne.



Zdjęcie 4.25 Obraz badany. Źródło: materiały własne.

Współczynnik korelacji	Liczba pikseli badanego obiektu ciemniejszych od wzorca	Liczba pikseli badanego obiektu jaśniejszych od wzorca	Komunikat programu
0.989351	35310	0	Prawdopodobna ciemna plama na kasetonie

Tablica 4.10 Wyniki analizy liczbowej.

Tym razem, analiza tablicy 4.10, prowadzi do wniosku, iż został napotkany fałszywie pozytywny błąd systemu. Komunikat programu informuje o awarii w momencie, gdy takowa nie występuje. Powodem takiej sytuacji może być znaczne przesunięcie obiektów względem siebie. W tym przypadku, część jasnych punktów wzorca nie jest zawarta na zdjęciu badanym 4.25, z tego powodu, podczas analizy, dokonując różnicy obu obrazów, niektóre, jasne piksele wzorca nie są kompensowane. Dodatkowo, to samo przesunięcie powoduje tak duże zmiany współczynnika korelacji, że system dokonuje błędnej klasyfikacji.

## ROZDZIAŁ 5

### PODSUMOWANIE I WNIOSKI

Przeprowadzona analiza doprowadziła do wyjaśnienia wszystkich problematycznych kwestii związanych projektem. Wszystkie kluczowe cele zostały spełnione. Zostały zaproponowane rozwiązania techniczne pozwalające na wdrożenie systemu technicznego do placówek korzystających z oznakowania reklamowego. Został stworzony program wykorzystujący elementy przetwarzania obrazów biblioteki OpenCV do wykrywania uszkodzeń na kasetonach reklamowych. Poprawność działania programu analizującego uszkodzenia została zbadana przy pomocy testów praktycznych. Na podstawie obrazu wzorcowego i badanego, system miał za zadanie rozróżnić, czy zaistniała usterka kasetonu i wysłać odpowiedni komunikat.

Analiza zagadnienia i testy oprogramowania ukazały, czy zaprojektowany system techniczny jest opłacalny oraz sprawny. Badania odpowiedziały na pytanie, z jaką pewnością i w jakim zakresie, układ może być stosowany do wykrywania uszkodzeń na oznakowaniu reklamowym. Analiza wszystkich dostępnych w tej chwili przypadków pozwala przypuszczać, że działanie systemu jest prawidłowe, jednakże do ostatecznego potwierdzenia jego skuteczności niezbędne będzie przeprowadzenie szerokich badań w terenie. Oczywiście, pojawiały się błędy w działaniu, jednak mogły być one wymuszone dość skrajnymi warunkami, w jakich pracował program. Projekt pozwolił Autorowi na zgłębienie zagadnień związanych z programowaniem i cyfrowym przetwarzaniem obrazu.

#### 5.1. Wnioski z przeprowadzonych badań

Głównym celem projektu było stworzenie uniwersalnego, skutecznego i taniego wizyjnego systemu technicznego który przy pomocy elementów przetwarzania obrazów przeanalizuje kaseton w celu wykrycia wystąpienia usterki. Cel ten został osiągnięty, zaprojektowany system spełnia wszystkie kluczowe wymagania.

Analiza rynku i zaproponowane rozwiązania techniczne pozwalają w ekonomiczny sposób podejść do kwestii sprzętu wykorzystywanego przez układ. Zaprojektowany program, okazał się skuteczny, sprawny i uniwersalny. Problemy przetwarzania obrazów zostały rozwiązane przy użyciu skomplikowanych funkcji, jednak samo działanie programu jest na tyle proste, że nawet osoba bez wiedzy programistycznej oraz znajomości biblioteki OpenCV będzie mogła nim operować. Badania eksperymentalne wykazały dużą skuteczność wykrywania wszystkich najczęstszych uszkodzeń przez system. Program z dużą efektywnością decydował o zaistnieniu usterki lub jej braku. Dodatkowo, skutecznie rozróżniał rodzaje usterki, w

zależności od zaistniałej sytuacji, co może być bardzo przydatną funkcją w przypadku wdrożenia systemu do większej liczby placówek.

Program, niestety, nie był nieomylny. Podczas eksperymentów system źle określił rodzaj usterki, był to mały błąd niewymagający większej uwagi. Problemem, który stawia procedurę analizy usterki pod znakiem zapytania jest fałszywie pozytywna decyzja o awarii kasetonu. Jednak należy zauważyć, iż była to sytuacja szczególna, w której zostały połączone niekorzystne warunki świetlne z dużym przesunięciem kamery, dlatego też, pomimo błędnie przeanalizowanego przypadku, nadal można uważać, że wizyjny system techniczny wykrywający uszkodzenia na kasetonach reklamowych działa sprawnie i skutecznie. Trzeba również dodać, iż jest to wersja testowa, a do jej ewentualnego wdrożenia w życie, system będzie wiele razy poprawiany, co jest niemożliwe w tym momencie z powodu braku odpowiednich zdjęć do badań.

Projekt okazał się opłacalny, bardzo niska cena maksymalna niestety została przekroczona, jednak zaproponowane rozwiązania pozwoliły na uważanie systemu jako projektu rentownego. Dodatkowo, udowodniono efektywne działanie programu przy analizie liter przestrzennych, co pozwoli na rozszerzenie zakresu implementacji systemu.

## **5.2. Możliwości rozbudowy**

Zakres pracy inżynierskiej nie obejmuje całości materiału związanego z projektem wizyjnego systemu technicznego. Jest jeszcze wiele dziedzin, w których projekt wymaga dopracowania. Szczególnie dużo uwagi należałoby poświęcić stworzeniu części sprzętowej systemu, która w pracy została zawarta tylko pobieżnie. Do stworzenia projektu, który mógłby być zainstalowany w placówce wymagane jest zaprojektowanie ramienia, do którego można będzie przyczepić kamerę. Wymaga zaprojektowania wodoszczelna obudowa do Raspberry Pi.

Kolejną kwestią jest problem znalezienia optymalnego miejsca gdzie komputer mógłby pracować bezawaryjnie. W kwestii komunikacji serwera z placówką oraz samej kamery z jednostką centralną również też pojawia się wiele problemów, które wymagają rozwiązania. Główną przeszkodą do poprawnej komunikacji z serwerem jest zaprojektowanie programu, który umożliwi zdalne sterowanie Raspberry Pi m.in. wydawanie rozkazów mających na celu wysłanie przez komputer zdjęcia obiektu badanego, reset systemu i jego zdalny restart oraz dostęp do kodu źródłowego programu. Dodatkowo należy przeprojektować program analizujący usterki tak, aby pobierał informacji tylko z podłączonych kamer internetowych.

Również program systemu wymaga dopracowania. Najważniejszym problemem jest dobranie progów odpowiadających za pojawianie się fałszywie pozytywnych wyników.

Istotną kwestią jest zwiększenie liczby badań eksperymentalnych, którym poddany jest program. Można również zastanowić się nad innymi zastosowaniami systemu w branży oznakowania reklamowego.

## BIBLIOGRAFIA

- [1] Gary Bradski, Adrian Kaehler, „Learning OpenCV: Computer Vision with the OpenCV Library”, O’Reilly Media, 2008
- [2] Witold Czajewski, Marcin Iwanowski, Maciej Sławinski, „Inteligentne Maszyny i Systemy preskrypt”, Program rozwojowy Politechniki Warszawskiej, Warszawa, 2010
- [3] Dokumentacja biblioteki OpenCV, dostępna w Internecie: <http://docs.opencv.org/> [dostęp 3 lutego 2013]
- [4] Maciej Radziński, „Cyfrowe przetwarzanie obrazów” W: Korekcja histogramów dostępna w Internecie: <http://atol.am.gdynia.pl/tc/Radzienski/Polepszanie.htm> [dostęp 3 lutego 2013]
- [5] Canny, J., “A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 1986. Dostęp z Internetu: [http://perso.limsi.fr/Individu/vezien/PAPIERS\\_ACS/canny1986.pdf](http://perso.limsi.fr/Individu/vezien/PAPIERS_ACS/canny1986.pdf) [dostęp 3 lutego 2013]
- [6] Lowe, D. G., “*Distinctive Image Features from Scale-Invariant Keypoints*”, International Journal of Computer Vision, 2004. Dostęp z internetu: <http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf> [dostęp 2 lutego 2013]
- [7] Artykuł porównujący charakterystyki deskryptorów. Dostęp z Internetu: <http://computer-vision-talks.com/2011/08/feature-descriptor-comparison-report/> [dostęp 7 luty 2013]
- [8] Strona producenta OLinuxIno. Dostęp z Internetu: <https://www.olimex.com/Products/OLinuXino/A13/A13-OLinuXino-MICRO/> [dostęp 5 lutego 2013]
- [9] Strona producenta Raspberry Pi. Dostęp z Internetu: <http://raspberrypi.rsdelivers.com/product/raspberry-pi/raspberry-pi-type-b/> [dostęp 5 lutego 2013]





