

Politechnika Poznańska
Wydział Elektroniki i Telekomunikacji
Katedra Telekomunikacji Multimedialnej i Mikroelektroniki

Praca dyplomowa magisterska

**REKTYFIKACJA OBRAZÓW Z SYSTEMU
STEREOWIZYJNEGO Z KAMERĄ
NAGRYWAJĄCĄ W PODCZERWIENI**

Bartosz Siewior

Promotor
dr inż. Sławomir Maćkowiak

Poznań, 2010 r.

Tutaj przychodzi karta pracy dyplomowej;
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy
ksero.

Spis treści

1	Wstęp	1
1.1	Cel pracy	1
1.2	The aim of the thesis	1
1.3	Zakres pracy	1
1.4	Streszczenie pracy	2
2	Podstawy teoretyczne	6
2.1	Właściwości kamer i kalibracja	6
2.1.1	Model i parametry wewnętrzne kamery	6
2.1.2	Zniekształcenia wprowadzane przez obiektywy	7
2.1.3	Macierz rotacji i wektor przesunięcia	8
2.1.4	Operacja rzutowego odzwierciedlenia	10
2.1.5	Kalibracja kamery	12
	Obiekt kalibrujący	12
	Algorytm kalibrujący	12
2.2	Stereowizja i geometria epipolarna	16
2.2.1	Stereowizja	16
2.2.2	Geometria epipolarna	16
2.2.3	Macierz zasadnicza i podstawowa	17
	Macierz zasadnicza	18
	Macierz podstawowa	19
2.3	Kalibracja systemu dwóch kamer i rektyfikacja	19
2.3.1	Kalibracja systemu dwóch kamer	19
2.3.2	Rektyfikacja	20
	Rektyfikacja bez uprzedniej kalibracji kamer	21
	Poszukiwanie macierzy podstawowej	21
	Algorytm Hartley’a	21
	Rektyfikacja poprzedzona uprzednią kalibracją kamer	22
	Algorytm Bouguet’a	23
2.4	Właściwości kamer termowizyjnych	24
3	Praca własna	28
3.1	Badany system stereowizyjny	28
3.1.1	Kamera termowizyjna	28
3.1.2	Kamera tradycyjna	29
3.2	Wstępna kalibracja	31
3.2.1	Konwersja sekwencji filmowej do obrazów	32
3.2.2	Dopasowanie rozdzielczości badanych obrazów	32
	Sekwencja w podczerwieni	33

	Sekwencja w zakresie światła widzialnego	33
3.2.3	Wybór korespondujących ze sobą klatek	34
	Kamera działająca w zakresie podczerwieni	35
	Kamera działająca w zakresie światła widzialnego	35
	Program wybierający odpowiednie klatki	37
3.2.4	Program do wstępnej kalibracji	39
3.3	Kalibracja algorytmem Zhanga	40
3.3.1	Program kalibrujący	40
	Narożniki pól szachownicy	41
	Parametry wewnętrzne i parametry zniekształceń	42
	Niwelacja zniekształceń	43
3.3.2	Kalibracja w całości	44
3.3.3	Wyniki kalibracji	44
3.4	Kalibracja i rektyfikacja systemu stereowizyjnego	45
3.4.1	Program do kalibracji i rektyfikacji	45
	Kalibracja z rektyfikacją	46
	Przekształcenie rektyfikacyjne	47
3.4.2	Rektyfikacja w całości	48
3.4.3	Wyniki rektyfikacji	48
3.5	Rektyfikacja badanego systemu stereowizyjnego	50
3.5.1	Metoda Hartley'a	50
	Program do znajdowania punktów charakterystycznych.	50
	Wyniki rektyfikacji	52
3.6	Rektyfikacja i kalibracja badanego systemu.	53
3.6.1	Tworzenie obrazu szachownicy	54
	Program wyznaczający środki punktów charakterystycznych	54
3.6.2	Rektyfikacja metodą Bouguet'a	59
3.6.3	Rektyfikacja metodą Hartley'a	59
3.6.4	Kalibracja i rektyfikacja metodą Hartley'a	62
4	Zakończenie	64
4.1	Podsumowanie wyników	64
4.2	Wnioski z przeprowadzonej analizy	65
A	Zawartość płyty CD	68
B	Opis plików konfiguracyjnych wchodzących w skład pracy	70
B.1	Plik programu do znajdowania narożników szachownicy	70
B.2	Plik programu do kalibracji pojedynczej kamery	71
B.3	Plik programu tworzącego obrazy pozbawione zniekształceń	72
B.4	Plik programu kalibrującego i rektyfikującego	72
B.5	Plik programu dokonującego przekształcenia rektyfikacyjnego	74
B.6	Plik programu znajdującego środki punktów charakterystycznych	75
	Literatura	76

Rozdział 1

Wstęp

1.1 Cel pracy

Celem pracy jest sprawdzenie działania tradycyjnych metod rektyfikacji obrazów z systemu dwóch kamer, dla systemu stereowizyjnego w którym jedna z kamer nagrywa sekwencję filmową w zakresie światła podczerwonego. Wykorzystując algorytmy wchodzące w skład biblioteki OpenCV¹ należy stworzyć oprogramowanie które rektyfikuje obrazy uzyskane z dwóch kamer działających w zakresie światła widzialnego. Następnie obrazy nagrane przez system dwóch kamer, w którym jedna z kamer działa w zakresie podczerwieni muszą zostać zaadoptowane do działania z napisanym wcześniej programem, działającym dla tradycyjnego systemu stereowizyjnego. Ostatecznie należy wybrać najlepszą metodę rektyfikacji obrazów z badanego systemu dwóch kamer.

1.2 The aim of the thesis

The aim of the thesis is to check the possibility of using traditional rectification methods for the stereovision system where one of the cameras is the thermovision camera. Using the algorithms that are included in OpenCV² library one has to create software that rectifies images from traditional stereovision system (system where two cameras records sequences in the visual spectrum range). The next step is to adopt the images from the system with thermovision camera to work with the previously created application. Once it is done, one has to choose the best method for rectifying images taken by examined system.

1.3 Zakres pracy

W skład pracy wchodzi przegląd istniejącej literatury na temat, tworzenia obrazu przez kamery, kalibracji kamer, geometrii epipolarnej i rektyfikacji, w szczególności opis wykorzystanych algorytmów kalibrujących i rektyfikujących obrazy z systemu stereowizyjnego, oraz krótki opis właściwości kamer działających w zakresie podczerwieni, powodujących, że rektyfikacja obrazów z takich kamer jest tak trudna.

¹Biblioteka funkcji wykorzystywanych podczas obróbki obrazu, oparta o otwarty kod i zapożyczona przez Intela

²OpenCV is a computer vision library originally developed by Intel.

W zakres pracy wchodzi wybór algorytmów rektyfikujących na których przetestowane zostaną obrazy pochodzące z badanego systemu, stworzenie oprogramowania umożliwiające wstępne dopasowanie do siebie obrazów z kamery tradycyjnej i kamery na podczerwień, stworzenie programu umożliwiającego kalibrację i rektyfikację obrazów z badanego systemu oraz analiza wyników i wnioski płynące z przeprowadzonego postępowania.

Struktura pracy jest następująca. W rozdziale 2 przedstawiono przegląd literatury na temat kalibracji, geometrii epipolarnej, rektyfikacji oraz algorytmów kalibrujących i rektyfikacyjnych użytych w pracy. Rozdział 3 zawiera sekcje dokumentujące prace własną dzięki której możliwe było przeprowadzenie zawartej w celu pracy analizy oraz uzyskanie wyników umożliwiających sformułowanie wniosków podsumowujących przeprowadzone badanie. W szczególności rozdział ten poświęcony jest opisowi zawierającemu wszystkie działania jakie zostały wykonane w celu rozwiązania problemu zawierającego się w celu pracy. Rozdział 4 zawiera zbiorcze wyniki analizy i wnioski z niej płynące. Zawiera również opis wszystkich programów i plików konfiguracyjnych dzięki którym możliwe jest przeprowadzenie podobnej analizy w przyszłości lub udoskonalenie metod rektyfikacji wchodzących w skład pracy własnej.

1.4 Streszczenie pracy

Stereowizja komputerowa to próba symulacji możliwości ludzkich oczu. Stereowizja zakłada postrzeganie tych samych obiektów świata trójwymiarowego dwiema kamerami odsuniętymi od siebie na jakąś odległość. Dzięki takiemu podejściu otrzymujemy dwa obrazy. Każdy punkt na obrazie z jednej kamery odpowiada punktowi na obrazie z drugiej kamery, punkty te są przesunięte o jakąś odległość. Posiadając informację na temat położenia korespondujących ze sobą punktów oraz informację na temat odległości pomiędzy kamerami, możemy, korzystając z algorytmów stereowizyjnych symulować zachowania ludzkich oczu. Możemy na przykład wyliczyć położenie korespondujących punktów w przestrzeni trójwymiarowej a co za tym idzie otrzymać trójwymiarowy obraz.

Celem stereowizji jest pozyskiwanie informacji na temat sceny (obrazu) posługując się widokami z obu kamer jednocześnie. Zastosowanie jakichkolwiek algorytmów stereowizyjnych wiąże się z poszukiwaniem cech obrazu (odpowiadających sobie punktów) na jednym jak i na drugim obrazie. Takie poszukiwanie jest znacznie łatwiejsze a sam algorytm posiada dużo mniejszą złożoność obliczeniową jeżeli wiadome jest gdzie dokładnie poszukiwać tych punktów. Można powiedzieć, iż celem rektyfikacji jest maksymalne zawężenie obszaru poszukiwań punktów korespondujących ze sobą na dwóch obrazach. Algorytmy rektyfikacyjne dopasowują do siebie obrazy z kamer systemu stereowizyjnego w taki sposób aby poszukiwanie korespondujących cech na obu obrazach odbywało się tylko w obrębie jednego wiersza. Obrazy po procesie rektyfikacji mają być w tej samej płaszczyźnie (współpłaszczyznowe), a korespondujące ze sobą punkty mają znajdować się w odpowiadających sobie wierszach na każdym z obrazów.

Każda kamera, czy aparat fotograficzny posiadają własne, unikalne właściwości, takie jak ogniskowa, lub punkt główny. Dzięki technikom przetwarzania obrazu możliwe jest symulowanie zachowań kamer w programach komputerowych. Stosując odpowiednie algorytmy można wyznaczyć parametry odpowiedzialne za zachowanie

kamer w rzeczywistym świecie. Proces wyznaczania wszystkich tych parametrów nazywa się kalibracją kamery. Skalibrowana kamera to kamera której wszystkie właściwości są znane w stopniu umożliwiającym jej symulację w świecie wirtualnym. Ponieważ kalibracja wyznacza wszystkie właściwości kamer, to wyznacza również parametry zniekształceń wprowadzanych przez obiektywy. Bazując na informacji na temat tych zniekształceń i stosując odpowiednie techniki przetwarzania obrazów możliwe jest uzyskanie obrazów pozbawionych zniekształceń. W celu odpowiedniej kalibracji kamery potrzebny jest obiekt którego wymiary są nam dobrze znane. W szczególności potrzebna jest wiedza na temat rozkładu punktów charakterystycznych na obiekcie kalibrującym. Sam algorytm kalibrujący działa tylko w obrębie tych punktów i wykorzystuje informacje na temat odległości tych punktów od siebie. Do przeprowadzenia procesu kalibracji potrzebny jest również sposób wyznaczenia lokalizacji tych punktów na obrazach obiektu. Najczęściej spotykanym obiektem kalibrującym jest szachownica, a punktami charakterystycznymi są narożniki pól szachownicy. Istnieje wiele algorytmów wyznaczających lokalizację narożników pól szachownicy, a produkty zwracane przez te algorytmy (położenie punktów na obrazach) mogą zostać bezpośrednio podane do algorytmu kalibrującego.

Wszelkiego rodzaju algorytmy stereowizyjne korzystają ze specjalnej geometrii stworzonej na potrzeby stereowizji. Geometria ta zwana jest geometrią epipolarną. Geometria epipolarna różni się nieco od geometrii tradycyjnej i opisuje wzajemne relacje między widokami z obu kamer, nie zależy w żaden sposób od filmowanej sceny a jedynie od właściwości kamer i ich wzajemnego położenia. Dzięki wprowadzeniu geometrii epipolarnej możliwe jest wyznaczenie wzajemnych relacji między kamerami tworzącymi układ stereowizyjny. Proces kalibracji systemu stereowizyjnego bazuje na geometrii epipolarnej i wyznacza wzajemne relacje między kamerami w systemie. Kalibracja układu dwóch kamer to proces w którym wyznaczane są wszystkie parametry kamer umożliwiające ich symulację komputerową oraz parametry odpowiedzialne za wzajemne relacje kamer w przestrzeni trójwymiarowej. Kalibracja systemu stereowizyjnego odbywa się podobnie do kalibracji pojedynczej kamery. Do przeprowadzenia postępowania kalibracyjnego potrzebny jest obiekt o znanych rozmiarach, taki na którym punkty charakterystyczne rozłożone są w znanych miejscach. Podobnie jak w przypadku kalibracji jednej kamery najczęściej stosowanym obiektem kalibrującym jest szachownica. Narożniki pól szachownicy wyszukiwane są za pomocą algorytmów przetwarzania obrazów i są one podawane jako dane wejściowe do algorytmu wyznaczającego właściwości i wzajemną relację obu kamer.

Rektyfikacja wyznacza wzajemne relacje między obrazami pochodzącymi z lewej i prawej kamery. Na podstawie informacji o tych relacjach możliwe jest przekształcenie obrazów pochodzących z systemu do postaci współpłaszczyznowej, dzięki czemu szukanie korespondujących punktów na obrazach jest szybsze i łatwiejsze. Istnieje wiele algorytmów służących do rektyfikacji obrazów z systemu dwóch kamer. Niektóre algorytmy korzystają z informacji na temat wzajemnego położenia kamer w przestrzeni, a więc informacji wyznaczanych w procesie kalibracji. Inne działają podobnie jak algorytmy kalibrujące na zasadzie dopasowania do siebie lokalizacji korespondujących punktów. Jeżeli system stereowizyjny jest mechanicznie jak najlepiej dopasowany (kamery są bliskie położenia współpłaszczyznowego) to każdy z dwóch typów algorytmów powinien zwracać podobne wyniki. W przypadku gdy kamery nie są mechanicznie dopasowane teoretycznie lepsze wyniki można uzyskać się za pomocą algorytmu, który wykorzystuje informacje na temat wzajemnego po-

łożenia kamer w przestrzeni. Do sprawdzenia działania obydwu typów algorytmów można użyć tego samego obiektu kalibrującego. Jeżeli korzystamy z algorytmu wymagającego kalibracji to użycie obiektu kalibrującego jest konieczne. Jeżeli natomiast używamy algorytmu wyznaczającego przekształcenie rektyfikacyjne na podstawie położenie korespondujących punktów, to obiekt kalibrujący może posłużyć jako źródło tych punktów. Rektyfikacja i kalibracja jest więc swego rodzaju "żonglerką" algorytmami przetwarzania obrazów, z których każdy musi zostać wywołany w odpowiedniej kolejności, określoną ilość razy.

Jeżeli system stereowizyjny składa się z kamery tradycyjnej i kamery działającej w zakresie światła podczerwonego, powstaje problem wyboru obiektu kalibrującego. Szachownica nie jest już tak dobrym wyborem jak w przypadku tradycyjnej stereowizji. Pola szachownicy będą widziane przez kamerą nagrywającą w zakresie światła widzialnego, ale jeżeli nie wydzielają one żadnego ciepła to nie są one możliwe do rozróżnienia na obrazach pochodzących z kamery termowizyjnej. Do przeprowadzenia badań w zakresie kalibracji i rektyfikacji takiego systemu, potrzebny jest obiekt, na którym punkty charakterystyczne będą jednakowo widziane przez obie kamery. Obiekt taki musi posiadać punkty które są dobrze rozróżnialne przez kamery tradycyjne i dodatkowo wydzielają ciepło. Można podjąć próbę stworzenia szachownicy na której narożniki pól będą wydzielają ciepło, dzięki czemu będą rejestrowane przez kamerę termowizyjną. Jednym z pomysłów może być na przykład użycie lampek choinkowych które zamontowane zostaną w miejscu, w którym znajdują się narożniki szachownicy. Dzięki temu możliwe będzie stworzenie obrazu szachownicy na zdjęciach pochodzących z kamery działającej w podczerwieni. Tak przygotowane zdjęcia mogą zostać podane do algorytmów znajdujących lokalizację narożników pól szachownicy i dalej do algorytmu kalibrującego i rektyfikującego.

Poza problem wyboru obiektu służącego do kalibracji i rektyfikacji istnieje jeszcze problem różnicy we właściwościach obrazów pochodzących z kamery tradycyjnej i kamery termowizyjnej. Obrazy pochodzące z badanego systemu różnią się rozdzielczością i dodatkowo kamera tradycyjna rejestruje więcej klatek na sekundę niż kamera nagrywająca w podczerwieni. W celu zbadania takiego systemu potrzebne jest więc rozwiązanie które dopasuje do siebie rozdzielczości obrazów pochodzących z obu kamer oraz usunie niepotrzebne klatki z sekwencji nagranej kamerą tradycyjną. Klatki pochodzące z obu sekwencji muszą zostać dopasowane w taki sposób żeby obrazy biorące udział w dalszych przekształceniach były ze sobą tożsame (wszystkie obiekty muszą znajdować się odpowiednich miejscach na lewym i prawym obrazie).

Ostatecznie jeżeli obrazy z kamer zostały do siebie odpowiednio dopasowane i stworzony został wymagany obiekt kalibrujący możliwe jest przeprowadzenie kalibracji oraz rektyfikacji badanego systemu. Zrektyfikowane obrazy mogą zostać pokazane na rysunkach w celu sprawdzenia zniekształceń jakie występują w obrazach po przekształceniach. W celu sprawdzenia dokładności rektyfikacji obrazy zawierające obiekt kalibrujący można poddać procedurze rektyfikacyjnej. Dzięki temu możliwe jest ponowne odszukanie punktów charakterystycznych na każdym z obrazów. O dokładności rektyfikacji świadczy położenie punktów na obrazach zrektyfikowanych. W przypadku rektyfikacji idealnej wszystkie punkty znajdują się w tym samym miejscu po nałożeniu na siebie obrazu lewego i prawego. Ponieważ nie jest możliwe uzyskanie obrazów zrektyfikowanych idealnie to położenie punktów na obrazach zrektyfikowanych będzie się nieco różnić od siebie. Błąd rektyfikacji może zostać określony jako średnia różnica położenie każdego punktu na wszystkich obrazach, a zestawienie błędów dla każdej metody może zostać pokazane na

wykresie.

Rozdział 2

Podstawy teoretyczne

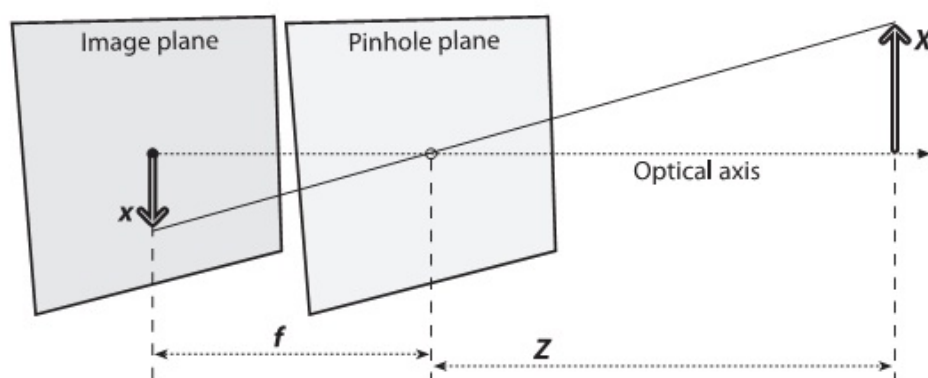
2.1 Właściwości kamer i kalibracja

Podrozdział opisuje matematyczne instrumenty dzięki którym możliwe jest symulowanie podstawowych zachowań kamer, przedstawia wprowadzane przez kamery zniekształcenia i pokazuje sposoby niwelowania tych zniekształceń.

2.1.1 Model i parametry wewnętrzne kamery

Analizując najprostszy model kamery (*Pinhole camera model*) możemy zgromadzić i zrozumieć niezbędne informacje na temat macierzy wewnętrznej¹ kamery, która jest niezbędna do zrozumienia procesu kalibracji kamer.

Najprostszy model kamery [Bradski08] jest przedstawiany jako urojony mur z niewielką dziurą na środku, światło jest postrzegane jako wychodzące z oddalonego obiektu, promienie słoneczne przechodzą tylko i wyłącznie przez dziurę na środku muru. W takim modelu obiekt powstający na płaszczyźnie obrazu jest zawsze ostry a jego wielkość zależy tylko od ogniskowej kamery (rys. 2.1).



Rysunek 2.1: Najprostszy model kamery ([Bradski08], str. 372.)

¹Parametry kamery dzięki którym rejestrowany świat trójwymiarowy przedstawiany jest na dwuwymiarowej płaszczyźnie.

Z rysunku 2.1 wynika równanie:

$$-\frac{x}{f} = \frac{X}{Z} \implies -x = f \frac{X}{Z} \quad (2.1)$$

Zamieniając miejscami płaszczyznę obrazu i model kamery [Bradski08] na rysunku 2.1, możemy przekształcić wzór (2.1) do postaci:

$$\frac{x}{f} = \frac{X}{Z} \quad (2.2)$$

W rzeczywistym modelu kamery środek matrycy nie znajduje się dokładnie na osi optycznej dlatego do przedstawionego modelu trzeba dodać dwa dodatkowe parametry: c_x oraz c_y . W rezultacie otrzymujemy prosty model w którym obiekt rzeczywisty o współrzędnych (X, Y, Z) jest wyświetlany na dwuwymiarowym ekranie w miejscu o współrzędnych (x_{ekranu}, y_{ekranu}) , gdzie:

$$x_{ekranu} = f_x \left(\frac{X}{Z} \right) + c_x, \quad y_{ekranu} = f_y \left(\frac{Y}{Z} \right) + c_y, \quad (2.3)$$

$$\text{gdzie :} \quad f_x = s_x F, \quad f_y = s_y F,$$

F - rzeczywista ogniskowa, s_x, s_y - wielkości elementów na których powstaje obraz.

Parametry f_x, f_y, c_x i c_y są parametrami wewnętrznymi kamery, natomiast macierz utworzona z tych parametrów nazywa się wewnętrzną macierzą kamery:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

2.1.2 Zniekształcenia wprowadzane przez obiektywy

Każdy obiektyw wprowadza dwa typy zniekształceń: zniekształcenie radialne [Bradski08], powodowane kształtem obiektywu oraz zniekształcenie tangensowe [Bradski08], powodowane sposobem łączenia obiektywu z aparatem.

1. Zniekształcenie radialne:

Zniekształcenie to jest zerowe w punkcie optycznego środka przestrzeni obrazu i rośnie proporcjonalnie do odległości od tego środka. W praktyce zniekształcenia te opisywane są za pomocą wzoru Taylora [Bradski08].

Uwzględniając zniekształcenia radialne obiektywu, położenie punktu w przestrzeni obrazu liczone jest za pomocą poniższych wzorów²:

$$x_r = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.4)$$

$$y_r = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (2.5)$$

²Wzory wyliczone są na podstawie wzoru Taylora, więcej informacji można znaleźć w [Bradski08]

Współrzędne x oraz y to współrzędne punktu na obrazie bez uwzględnienia zniekształcenia radialnego, punkt (x_r, y_r) to położenie tego punktu ale z uwzględnieniem zniekształceń radialnych. Zmienne k_1 , k_2 oraz k_3 to zniekształcenia radialne obiektywu.

2. Zniekształcenie tangensowe:

W celu zniwelowania tego typu zniekształceń wprowadza się dwa dodatkowe parametry p_1 oraz p_2 . Położenie punktu liczone z uwzględnieniem tego typu zniekształceń liczone jest za pomocą dwóch poniższych wzorów³:

$$x_t = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (2.6)$$

$$y_t = y + [p_1(r^2 + 2y^2) + 2p_2x] \quad (2.7)$$

Zbierając wszystkie parametry na temat zniekształceń i zapisując je w odpowiedniej formie otrzymujemy wektor parametrów zniekształceń kamery:

$$T = [k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$$

Wygodnie jest umiejscowić ostatni parametr zniekształcenia radialnego k_3 na ostatnim miejscu w wektorze zniekształceń, ponieważ parametr ten jest używany tylko dla niewielkiej liczby obiektywów znajdujących się na rynku. Dla większości obiektywów dwa pierwsze parametry opisujące zniekształcenie radialne w zupełności wystarcza, do skompensowania tego zniekształcenia.

2.1.3 Macierz rotacji i wektor przesunięcia

Fotografując jakiś obiekt, możemy określić jego położenie względem systemu współrzędnych kamery jako macierz rotacji oraz wektor przesunięcia (rys. 2.2).

Rotacja może zostać przedstawiona jako mnożenie wektora współrzędnych i macierzy kwadratowej o odpowiednich rozmiarach, Rysunek (rys. 2.3) pokazuje rotację jako mnożenie macierzy w dwóch wymiarach.

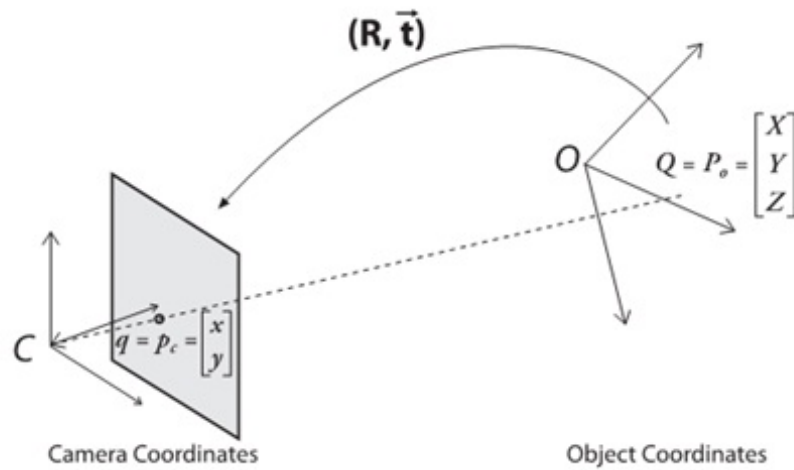
Rotacja w przestrzeni trójwymiarowej może zostać zdekomponowana jako rotacja w dziedzinie dwuwymiarowej wokół każdej osi. Jeżeli obiekt w dziedzinie trójwymiarowej będzie "przekręcany" w dziedzinie dwuwymiarowej dookoła osi x , y oraz z , po kolei, odpowiednio o kąty rotacji ψ , φ , i θ , rezultatem będzie macierz rotacji R , na którą składają się macierze $R_x(\psi)$, $R_y(\varphi)$, i $R_z(\theta)$, gdzie:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix}, \quad R_y(\varphi) = \begin{bmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix}$$

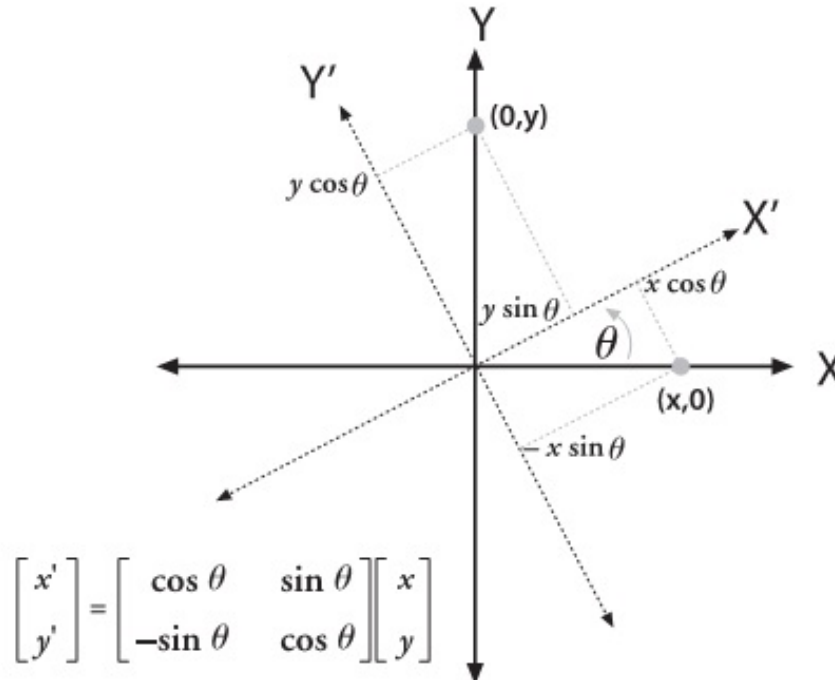
$$R_z(\Theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R(\theta) + R(\varphi) + R(\psi)$$

³Więcej informacji na temat pochodzenia wzorów znajduje się w [Bradski08]



Rysunek 2.2: [Bradski08], przekształcenie współrzędnych obiektu w przestrzeni na współrzędne kamery, punkt P na obiekcie jest widziany jako punkt p na płaszczyźnie obrazu, wzajemną relację między tymi punktami (układami współrzędnych) opisuje macierz rotacji oraz wektor przesunięcia.



Rysunek 2.3: [Bradski08], pokazuje, że rotacja punktu o kąt θ jest jednoznaczna z rotacją układu współrzędnych o ten sam kąt. Pokazuje również jak zmieniają się współrzędne punktu.

Wektor przesunięcia jest to sposób na reprezentację przesunięcia z jednego układu współrzędnych do drugiego, inaczej mówiąc jest to wektor reprezentujący offset między początkami układów współrzędnych.

Zbierając informacje na temat macierzy rotacji oraz wektora przesunięcia można wyprowadzić równanie na położenie rzeczywistego punktu (punktu obiektu ze świata trójwymiarowego) we współrzędnych kamery. Przyjmując punkt P_o jako punkt rzeczywisty oraz punkt P_c jako ten sam punkt we współrzędnych kamery, to relacja między tymi punktami wygląda następująco:

$$P_c = R(P_o - T) \quad (2.8)$$

Gdzie R to macierz rotacji, a T to wektor przesunięcia.

Powyższe równanie wraz z równaniami na korekcję parametrów wewnętrznych kamery tworzy układ równań, którego rozwiązywanie nazywa się procesem kalibracji kamery.

2.1.4 Operacja rzutowego odzwierciedlenia

Operacja rzutowego odzwierciedlenia[Bradski08] w komputerowym przetwarzaniu obrazów to termin oznaczający odzwierciedlenie jednej przestrzeni w drugą. Przykładem takiej operacji jest właśnie rzutowanie punktów rzeczywistego świata na matrycę w kamerze. Przyjmując punkt Q jako punkt rzeczywisty, rzutowany w punkt q i używając współrzędnych jednorodnych⁴ do określenia położenia obu tych punktów, operację rzutowego odzwierciedlenia możemy przedstawić jako mnożenie macierzy.

$$\begin{aligned} \tilde{Q} &= [X \ Y \ Z \ 1]^T \\ \tilde{q} &= [x \ y \ 1]^T \end{aligned}$$

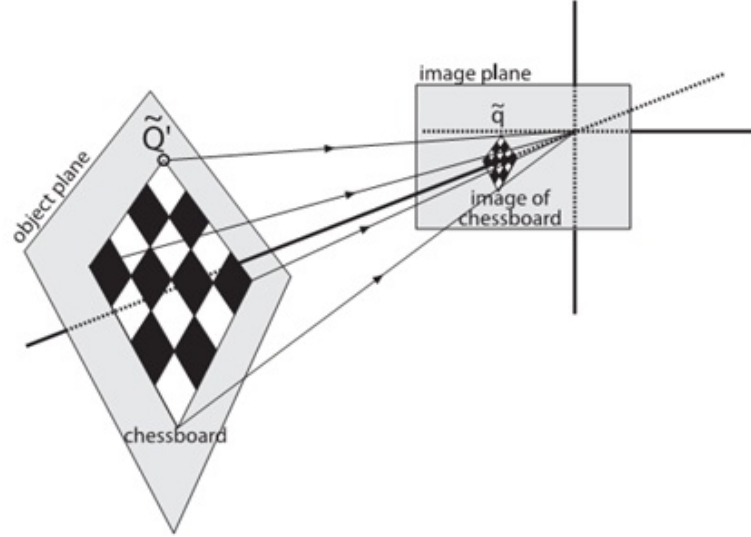
Traktując powyższe wzory jako odpowiednio położenie punktu rzeczywistego i położenie punktu na matrycy kamery, to procedurą rzutowego odzwierciedlenia (rys. 2.4) można przedstawić jako:

$$\tilde{q} = sH\tilde{Q} \quad (2.9)$$

Współczynnik s w powyższym wzorze jest odpowiedzialny za przeskalowanie przedmiotu rzeczywistego do wymiarów przedmiotu powstałego na płaszczyźnie obrazu. Operacja H składa się z dwóch części:

- fizycznej transformacji która wykrywa w gruncie rzeczy płaszczyznę obiektu który widzimy.
- projekcji przedstawiającej wewnętrzne współczynniki kamery.

⁴[Wiki01], współrzędne jednorodne to sposób reprezentacji punktów n -wymiarowych za pomocą $n + 1$ współrzędnych.



Rysunek 2.4: [Bradski08], widok płaskiego obiektu, operacja rzutowego odzwierciedlenia płaszczyzny obiektu na płaszczyznę obrazu.

Fizyczna transformacja jest sumą jakiejś rotacji R oraz jakiegoś przesunięcia $\mathbf{t}$, ponieważ opisywane operacje funkcjonują na współrzędnych jednorodnych, to taka fizyczna transformacja może zostać zapisana w jednej macierzy jako:

$$W = \begin{bmatrix} R & \mathbf{t} \end{bmatrix}$$

Ponieważ operacja projekcji przedstawia wewnętrzne współczynniki kamery, a wcześniej powiedziane było, że operacja rzutowego odzwierciedlenia może zostać zapisana jako iloczyn macierzy, wzór (2.9) może zostać zapisany jako:

$$\tilde{\mathbf{q}} = sMW\tilde{\mathbf{Q}}, \quad (2.10)$$

gdzie M to wewnętrzna macierz kamery.

Przyjmując takie położenie obiektu rzeczywistego w przestrzeni, że współrzędna Z wynosi 0, oraz zapisując macierz rotacji jako $R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$, to wzór (2.10) można zapisać jako:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & r_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = sM \begin{bmatrix} r_1 & r_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Z powyższego toku rozumowania wynika, że macierz H odzwierciedlająca punkty z płaszczyzny obiektu na punkty w płaszczyźnie obrazu określona jest następującym wzorem:

$$H = sM \begin{bmatrix} r_1 & r_2 & \mathbf{t} \end{bmatrix} \quad \text{gdzie} \quad M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

2.1.5 Kalibracja kamery

Kalibracja kamery jest w zasadzie operacją znajdowania wewnętrznych parametrów kamery oraz parametrów odpowiedzialnych za powstawanie zniekształceń obrazów filmowanych tymi kamerami. Znając wszystkie współczynniki wewnętrzne kamer jak i parametry zniekształceń można za pomocą kolejnych przekształceń matematycznych stworzyć obrazy pozbawione tych zniekształceń. Do przeprowadzenia procesu kalibracji kamery potrzebny jest obiekt o znanych kształtach i rozmiarach. Obiekt ten jest fotografowany wiele razy i w wielu różnych pozycjach, a następnie obrazy tego obiektu biorą udział w procesie kalibracji.

Obiekt kalibrujący

Obiekt biorący udział w procesie kalibracji może być w zasadzie dowolny, ponieważ jednak wymiary tego obiektu muszą być dobrze znane, punkty pochodzące z obiektu muszą być łatwe do znalezienia, a odległości między tymi punktami też muszą być znane. Dobrym i często wykorzystywanym obiektem w procesie kalibracji jest szachownica, a punktami biorącymi udział w kalibracji są wewnętrzne narożniki pól szachownicy. Odległości między narożnikami pól szachownicy mogą zostać łatwo zmierzone, z góry wiadomo ile jest pól szachownicy, a więc wiadomo również ile jest narożników które wezmą udział w kalibracji kamery. Kolejnym argumentem przemawiającym za szachownicą jest to, iż istnieje wiele algorytmów znajdujących narożniki szachownicy, a zwracane przez te algorytmy położenia punktów na szachownicy mogą zostać podane bezpośrednio jako dane wejściowe do algorytmu kalibrującego.

Zakładając, iż szachownica została wybrana jako obiekt wykorzystywany w procesie kalibracji, należy zastanowić się jak wiele narożników (zdjęć szachownicy) jest potrzebne do przeprowadzenia kalibracji i uzyskania wysokiej jakości wyników. Wiadomo, iż mamy cztery parametry wewnętrzne kamery, dodatkowo należy wyznaczyć pięć parametrów zniekształceń oraz sześć parametrów odpowiedzialnych za rotację i przesunięcie między układami współrzędnych fotografowanego obiektu i kamery. Postępując zgodnie z procedurą zawartą w [Bradski08] można stwierdzić iż dwa widoki szachownicy o rozmiarach 3×3 to wystarczająca ilość do kalibracji kamery, praktyka jednak pokazuje iż aby uzyskać wysokiej jakości wyniki należy użyć co najmniej dziesięciu ujęć szachownicy o wymiarach 7×8 . Pozycja szachownicy na każdym ujęciu powinna być inna, tylko w takim przypadku kalibracja zostanie przeprowadzona poprawnie.

Algorytm kalibrujący

Prezentowany w pracy algorytm kalibrujący jest połączeniem dwóch innych algorytmów. Do wyznaczenia parametrów wewnętrznych kamery używana jest metoda Zhanga [Zhang98], natomiast parametry zniekształceń wyznaczane są na podstawie algorytmu Browna [Brown71]. Algorytm użyty w procesie kalibracji będący połączeniem dwóch wymienionych algorytmów opisany jest w [Bradski08].

Kalibracja metodą Zhanga - współczynniki wewnętrzne

Proponowana technika wymaga tylko kilku (co najmniej dwóch) zdjęć znanego wzoru, który uchwycony jest w różnych miejscach. Zarówno kamera jak i fotogra-

fowany wzór może być dowolnie ruszany. Metoda składa się z jawnego rozwiązania oraz nieliniowego uściślenia bazującego na kryterium maksymalnego prawdopodobieństwa.

Notacja

Punkt w przestrzeni dwuwymiarowej jest określony jako $m = [u, v]^T$. Punkt w przestrzeni trójwymiarowej jest określony jako $M = [Z, Y, Z]^T$. Do określenia punktu posługując się współrzędnymi jednorodnymi wykorzystano notację $\tilde{x}$, a więc współrzędne jednorodne punktów odpowiednio w przestrzeni dwuwymiarowej i przestrzeni trójwymiarowej wyglądają następująco: $\tilde{m}=[u, v, 1]^T$ oraz $\tilde{M}=[Z, Y, Z, 1]^T$. Wzajemna zależność między punktem przestrzeni dwuwymiarowej a punktem przestrzeni trójwymiarowej bazuje na opisanym wcześniej rzutowym odzwierciedleniu i wygląda następująco:

$$s\tilde{m} = A \begin{bmatrix} R & t \end{bmatrix} \tilde{M}, \quad (2.11)$$

gdzie s to współczynnik odpowiedzialny za skalę, R i t , to odpowiednio macierz rotacji i wektor przesunięcia, a A to wewnętrzna macierz kamery, określona jako:

$$M = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

gdzie (u_0, v_0) to współrzędne głównego punktu matrycy, α i β to parametry wewnętrzne określone jako f_x oraz f_y , a γ to parametr opisujący współczynnik skośności pomiędzy osiami obu obrazów.

Dodatkowo użyte zostały następujące skróty $A^{-T} = (A^{-1})^T = (A^T)^{-1}$.

Rzutowe odzwierciedlenie płaszczyzn

Opisywany algorytm bazuje na koncepcji rzutowego odzwierciedlenia opisywanej wcześniej. Macierz opisująca taką operację mapowania H wygląda następująco:

$$H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \quad (2.12)$$

Ograniczenia dla parametrów wewnętrznych

Posiadając obraz rzeczywistego obiektu można znaleźć macierz H [Zhang98] odpowiedzialną za rzutowe odzwierciedlenie. Zakładając, że H jest postaci $H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$, z równania (2.12) otrzymujemy:

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

gdzie λ to dowolny skalar. Ponieważ r_1 i r_2 są ortonormalne możemy zapisać:

$$h_1^T A^{-T} A^{-1} h_2 = 0 \quad (2.13)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (2.14)$$

Powyższe dwa równania to podstawowe ograniczenia dla wewnętrznych parametrów.

Algorytm Zhanga

Algorytm zakłada znalezienie analitycznego rozwiązania które jest następnie optymalizowane zgodnie z kryteriami maksymalnego prawdopodobieństwa.

1. Jawne rozwiązanie.

Zakładając

$$B = A^{-T} A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (2.15)$$

B jest macierzą symetryczną i może zostać zapisana jako następujący wektor:

$$b = [B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33}]^T \quad (2.16)$$

Zapisując, że i -ta kolumna macierzy H to $h_i = [h_{i1} \ h_{i2} \ h_{i3}]^T$ otrzymujemy równanie:

$$h_i^T B h_j = v_{ij}^T b \quad (2.17)$$

$$v_{ij} = [h_{i1} h_{j1} \ h_{i1} h_{j2} + h_{i2} h_{j1} \ h_{i2} h_{j2} \ h_{i3} h_{j1} + h_{i1} h_{j3} \ h_{i3} h_{j2} + h_{i2} h_{j3} \ h_{i3} h_{j3}]$$

Równania (2.13) oraz (2.14) mogą zostać zapisane jako dwa jednorodne równania:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (2.18)$$

Jeżeli użyto n obrazów obiektu służącego do kalibracji, można zapisać n równań (2.18):

$$Vb = 0 \quad (2.19)$$

V jest macierzą o wymiarach $2n \times 3$. Jeżeli $n \geq 3$ dostajemy jedno unikalne rozwiązanie dla b , mając rozwiązanie dla b wszystkie współczynniki wewnętrznej macierzy A mogą zostać znalezione [Zhang98].

Kiedy tylko macierz A jest znana można znaleźć rozwiązania dla wszystkich zewnętrznych parametrów kamery bezpośrednio z równania (2.12):

$$\begin{aligned} r_1 &= \lambda A^{-1} h_1 \\ r_2 &= \lambda A^{-1} h_2 \end{aligned}$$

$$\begin{aligned}
r_3 &= r_1 \times r_2 \\
t &= \lambda A^{-1} h_3 \\
\text{gdzie } \lambda &= 1 / \|A^{-1} h_1\|
\end{aligned}$$

2. Estymacja maksymalnego prawdopodobieństwa.

Zaproponowane powyżej rozwiązanie polega na minimalizowaniu algebraicznego dystansu które nie ma sensu fizycznego, rozwiązanie to może zostać udoskonalone metodą maksymalnego prawdopodobieństwa.

Mamy n obrazków rzeczywistego obiektu na którym znajduje się m punktów biorących udział w kalibracji, zakładając, że wszystkie te punkty są jednakowo zniekształcone przez identyczny szum estymacja maksymalnego prawdopodobieństwa polega na zminimalizowaniu następującej formuły:

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2 \quad (2.20)$$

gdzie $\hat{m}(A, R_i, t_i, M_j)$ jest projekcją punktu M_j na obrazku i , zgodnie z równaniem (2.15). Minimalizacja równania (2.20) jest problemem nieliniowym który jest rozwiązywany za pomocą algorytmu Levenberg-Marquardt⁵. Algorytm ten potrzebuje startowych wartości dla macierzy wewnętrznej A , która jest liczona we wcześniej opisanej procedurze.

Kalibracja metodą Browna - zniekształcenia obiektywu

Metoda Browna[Brown71] rozwiązuje równania dla zniekształceń obiektywów bazując na początkowych wartościach dla współczynników wewnętrznych kamery oraz współczynników zniekształceń. Zaproponowany algorytm jako wartość startową macierzy wewnętrznej podaje wyliczoną macierz za pomocą algorytmu Zhanga, a jako startowe współczynniki zniekształceń podaje 0.

Przyjmując, że punkt o współrzędnych (x_p, y_p) jest punktem obrazu gdy nie bierzemy pod uwagę zniekształceń, a więc korzystając z najprostszego modelu kamery i wzorów (2.3) można zapisać:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_x X/Z + c_x \\ f_y X/Z + c_y \end{bmatrix}$$

Przyjmując, że punkt o współrzędnych (x_d, y_d) jest punktem obrazu gdy mamy do czynienia z zniekształceniami i korzystając z równań (2.4), (2.5), (2.6) i (2.7), możemy zapisać następujące podstawienie:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1 x_d y_d + p_2(r^2 + 2x_d^2) \\ p_1(r^2 + 2y_d^2) + 2p_2 x_d y_d \end{bmatrix}$$

Korzystając z dwóch przedstawionych równań dla wielu punktów algorytm znajduje parametry zniekształceń obiektywu.

⁵[Wiki02], obecnie jednym z najczęściej wykorzystywanych algorytmów optymalizacji nieliniowej

2.2 Stereowizja i geometria epipolarna

Przed przystąpieniem do opisu procesu rektyfikacji oraz algorytmów używanych w tym procesie, niezbędne jest zapoznanie się z pojęciem stereowizji oraz geometrii epipolarnej. W tym podrozdziale zawarte są również informacje na temat macierzy zasadniczej i podstawowej.

2.2.1 Stereowizja

Stereowizja komputerowa to nic innego jak próba symulacji możliwości ludzkich oczu. Stereowizja zakłada postrzeganie tych samych obiektów świata trójwymiarowego dwiema kamerami odsuniętymi od siebie na jakąś odległość. Dzięki takiemu podejściu otrzymujemy dwa obrazy. Każdy punkt na obrazie z jednej kamery odpowiada punktowi na obrazie z drugiej kamery, punkty te są przesunięte o jakąś odległość. Posiadając informację na temat położenia korespondujących ze sobą punktów oraz informację na temat odległości pomiędzy kamerami, możemy, korzystając z algorytmów stereowizyjnych symulować zachowania ludzkich oczu, a więc możemy na przykład wyliczyć położenie tych punktów w przestrzeni trójwymiarowej a co za tym idzie otrzymać trójwymiarowy obraz. W praktyce próba odzyskania trójwymiarowego obrazu składa się z czterech kroków:

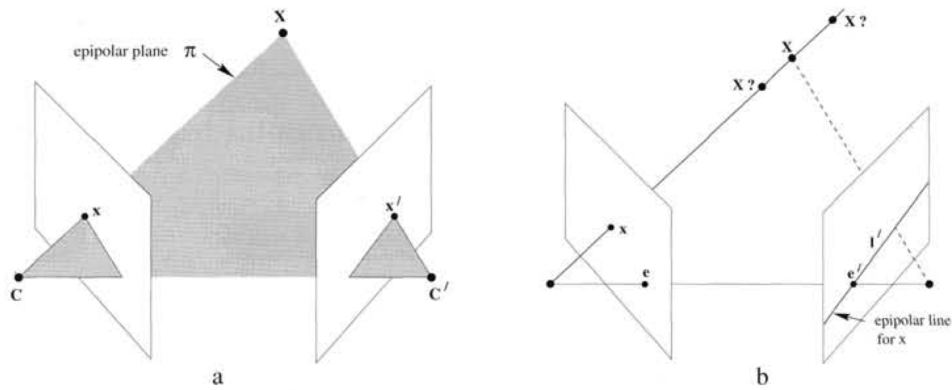
1. Matematyczne usunięcie zniekształceń powodowanych przez obiektywy, na przykład zgodnie z procedurą kalibracyjną opisana w poprzednim podrozdziale.
2. Dopasowanie kątów i odległości pomiędzy kamerami w procesie rektyfikacji. Wynikiem są zrektyfikowane obrazy, dopasowane do siebie wierszami⁶
3. Znalezienie tych samych cech w prawym i lewym obrazie, czyli korespondujących ze sobą punktów. Po tym kroku otrzymujemy tak zwaną mapę rozbieżności między korespondującymi punktami. Rozbieżność między punktami to różnica współrzędnych x tych punktów: $x_l - x_r$.
4. Znając geometryczne cechy kamer, możliwe jest przekształcenie mapy rozbieżności na odległości między punktami w rzeczywistym świecie. Ten krok nazywa się reprojekcją i dzięki niemu otrzymujemy mapę głębi.

2.2.2 Geometria epipolarna

Geometria epipolarna to geometria rzutowa[Hartley03] pomiędzy dwoma widokami. Geometria ta nie zależy w żaden sposób od filmowanej sceny a jedynie od wewnętrznych parametrów kamer i ich położenia względem siebie.

Geometria epipolarna jest w zasadzie geometrią przecięć płaszczyzn obrazu, w której wszystkie płaszczyzny przecinają się w jednej prostej, jest to postać łącząca punkty główne kamer. Załóżmy, że obie kamery filmują jeden wspólny punkt X , wtedy x jest obrazem punktu X w pierwszym widoku (widoku z pierwszej kamery), a punkt x' jest jego obrazem na drugim widoku. Wszystkie te trzy punkty leżą w jednej płaszczyźnie π , są więc współpłaszczyznowe (rys 2.5a).

⁶Wszystkie korespondujące ze sobą punkty na jednym i drugim obrazie znajdują się w tych samych wierszach.



Rysunek 2.5: [Hartley03], (a) obserwowany punkt X , jego obraz na lewym obrazie x , jego obraz na prawym obrazie x' oraz punkty główne obu kamer C i C' leżą na tej samej płaszczyźnie π . (b) Znając położenie punktu x , oraz punkty główne kamer, można stwierdzić, że obraz punktu X na prawym obrazie musi leżeć na linii l' .

Założmy, że znamy tylko położenie punktu x i chcemy znaleźć położenie punktu x' . Płaszczyzna π jest określona przez punkt x oraz podstawową linię⁷. Wiadome jest, że punkt x' musi leżeć gdzieś na płaszczyźnie π , a więc szukany punkt leży gdzieś na linii l' łączącej płaszczyznę π z płaszczyzną prawego obrazu (rys 2.5b). Linia l' jest linia epipolarna w odniesieniu do punktu x . W stereowizji nie trzeba więc przeszukiwać całej płaszczyzny prawego obrazu w celu znalezienia punktu x' , obszar poszukiwań można zageścić do linii l' .

Geometryczne wielkości opisujące geometrię epipolarną widoczne są na rysunku (rys 2.5b) i oznaczają:

- Epipole (e lub e') - punkt przecięcia linii podstawowej z płaszczyzną obrazu [Hartley03].
- Płaszczyzna epipolarna - płaszczyzna zawierająca linię podstawową (np. płaszczyzna π na rysunku (rys. 2.5a) [Hartley03].
- Linia epipolarna - linia w której przecina się płaszczyzna epipolarna z płaszczyzną obrazu (np. linia l' na rysunku (rys. 2.5b) [Hartley03].

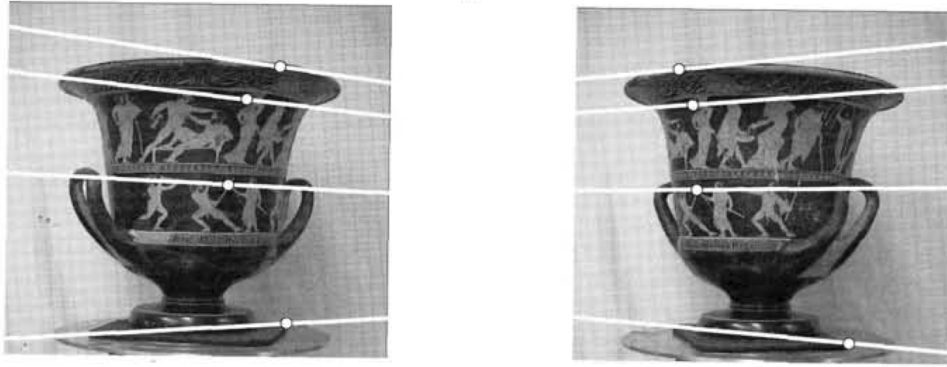
Rysunek (rys 2.6) pokazuje przykład geometrii epipolarnej.

2.2.3 Macierz zasadnicza i podstawowa

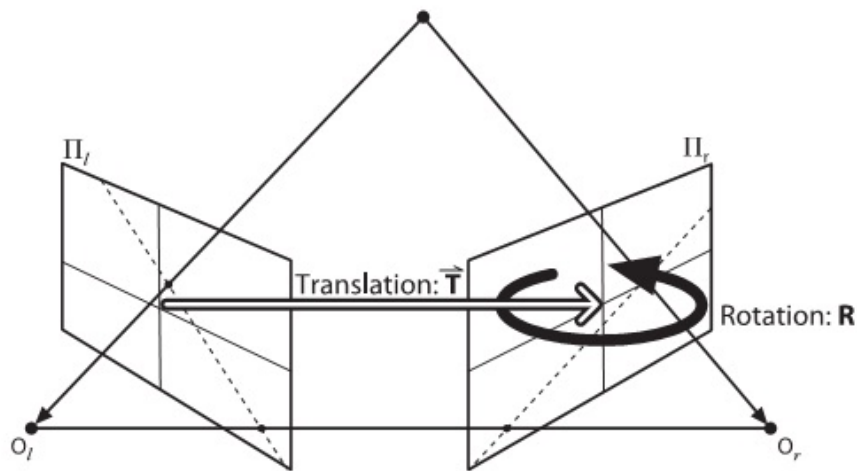
Macierz zasadnicza E zawiera informacje na temat rotacji i przesunięcia które łączą widoki z obu kamer (rys 2.7), macierz podstawowa F zawiera dokładnie takie same informacje z dodatkiem informacji na temat parametrów wewnętrznych obu kamer [Bradski08].

Macierz podstawowa przedstawia relację punktów na płaszczyźnie obrazu jednej kamery, we współrzędnych obrazu, do punktów na płaszczyźnie obrazu drugiej kamery również we współrzędnych obrazu. Macierz zasadnicza pokazuje wzajemną

⁷Linia podstawowa to linia łącząca ze sobą punkty główne obu kamer.



Rysunek 2.6: [Hartley03], przykład geometrii epipolarnej.

Rysunek 2.7: [Bradski08], macierz zasadnicza E zawiera informacje na temat rotacji i przesunięcia między kamerami w systemie stereowizyjnym.

relację punktów w przestrzeni fizycznej, nie zawiera natomiast żadnej informacji na temat wzajemnego położenia punktów we współrzędnych obrazu.

Macierz zasadnicza

Przyjmując punkt widziany przez obie kamery jako P , macierz zasadnicza E pokazuje relację, we współrzędnych fizycznych, punktu P widzianego przez lewą kamerę (punkt p_l), do tego samego punktu widzianego przez prawą kamerę (punkt p_r).

Postępując zgodnie z rozumowaniem przedstawionym w [Bradski08], można wykazać, że wzajemna relacja między punktami p_l i p_r jest określana przez macierz podstawową i określa się wzorem:

$$p_r^T E p_l = 0 \quad (2.21)$$

więcej informacji na temat pochodzenia wzoru i macierzy podstawowej znajduje się w [Bradski08].

Macierz podstawowa

W praktyce bardziej interesującym zagadnieniem jest wzajemna relacja między punktami na jednym i drugim obrazie określana we współrzędnych obrazu. Aby znaleźć relacje pomiędzy punktem na jednym obrazie i odpowiadającą mu linią epipolarną w drugim obrazie musimy posłużyć się parametrami wewnętrznymi obu kamer. Jeżeli za punkt q_l przyjmujemy obraz punktu P na lewym obrazie, we współrzędnych obrazu, a za punkt q_r obraz punktu P na prawym obrazie, w takich samych współrzędnych, to postępując zgodnie z tokiem rozumowania przedstawionym w [Bradski08], możemy zauważyć iż relacja między tymi punktami określana jest przez macierz podstawową i opisywana jest wzorem:

$$q_r^T F q_l = 0 \quad (2.22)$$

Wzajemna relacja między macierzą podstawową a macierzą zasadniczą określana jest wzorem:

$$F = (M_r^{-1})^T E M_l^{-1} \quad (2.23)$$

Rektyfikując obrazy z lewej i prawej kamery doprowadzamy do sytuacji w której macierz podstawowa i macierz zasadnicza są takie same $E = F$.

2.3 Kalibracja systemu dwóch kamer i rektyfikacja

W tej sekcji opisana zostanie kalibracja systemu złożonego z dwóch kamer, która bazuje na algorytmie opisanym w sekcji o kalibracji kamery [Zhang98], oraz rektyfikacja jak i algorytmy użyte do jej przeprowadzenia.

2.3.1 Kalibracja systemu dwóch kamer

Kalibracja stereowizyjna to proces w którym wyliczane są wzajemne zależności między dwiema kamerami w przestrzeni. Taka kalibracja polega na wyliczeniu macierzy rotacji R oraz wektora przesunięcia T pomiędzy kamerami. Algorytm kalibracji stereowizyjnej może dodatkowo wyliczyć macierze zasadniczą i podstawową dla danego systemu dwóch kamer.

Macierze zasadnicza E oraz macierz podstawowa F są wyliczane zgodnie ze wzorami pokazanymi w sekcji dotyczącej tych macierzy (dokładniejsze informacje znajdują się w [Bradski08]).

Aby wyliczyć macierz rotacji oraz wektor przesunięcia posłużono się procedurą kalibrującą pojedynczą kamerą (zgodnie z [Zhang98] i [Brown71]). Jeżeli przyjmujemy jako P położenie rzeczywistego punktu we współrzędnych obiektu na którym ten punkt się znajduje i zastosujemy postępowanie opisane w [Zhang98] i [Brown71], to dostaniemy następujące rezultaty dla lewej i prawej kamery:

$$P_l = R_l P + T_l \quad P_r = R_r P + T_r \quad (2.24)$$

Dodatkowo z rysunku (rys. 2.7) można zauważyć, że wzajemna relacja między P_l a P_r wygląda następująco:

$$P_l = R^T(R_r - T) \quad (2.25)$$

gdzie R oraz T to odpowiednio macierz rotacji oraz wektor przesunięcia jak na rysunku (rys. 2.7). Żeby znaleźć R oraz T wystarczy rozwiązać układ równań złożony z (2.24) i (2.25):

$$R = R_r(R_l)^T \quad (2.26)$$

$$T = T_r - RT_l \quad (2.27)$$

Reasumując wiadomości na temat kalibracji systemu stereowizyjnego dostajemy gotowy algorytm, działający w następujący sposób:

- Wiele obrazów z lewego i prawego widoku przetwarzanych jest zgodnie z algorytmem opisanym w sekcji na temat kalibracji pojedynczej kamery (zgodnie z [Zhang98] i [Brown71]). Dzięki temu wszystkie parametry wewnętrzne, macierze rotacji i wektory przesunięcia, są wyliczane dla każdej kamery osobno.
- Następnie za pomocą wyliczonych macierzy rotacji i wektorów przesunięcia dla każdej z kamer osobno wyliczane są macierz rotacji R i wektor przesunięcia T między kamerami zgodnie ze wzorami (2.26) i (2.27).
- Macierz zasadnicza E oraz macierz podstawowa F wyliczane są zgodnie ze wzorami (2.21) oraz (2.22).

Jeżeli posiadamy informacje na temat macierzy rotacji i wektora przesunięcia między kamerami, lub znamy macierz zasadniczą łączącą widoki z obu kamer możemy przystąpić do procesu rektyfikacji.

2.3.2 Rektyfikacja

Celem stereowizji jest pozyskiwanie informacji na temat sceny (obrazu) posługując się widokami z obu kamer jednocześnie. Zastosowanie jakichkolwiek algorytmów stereowizyjnych wiąże się z poszukiwaniem cech obrazu (odpowiadających sobie punktów) na jednym jak i na drugim obrazie. Takie poszukiwanie jest znacznie łatwiejsze a sam algorytm posiada dużo mniejszą złożoność obliczeniową jeżeli wiadome jest gdzie dokładnie poszukiwać tych punktów. Można powiedzieć, iż celem rektyfikacji jest maksymalne zawężenie obszaru poszukiwań punktów korespondujących ze sobą na dwóch obrazach. Algorytmy rektyfikacyjne dopasowują do siebie obrazy z kamer systemu stereo w taki sposób aby poszukiwanie korespondujących cech na obu obrazach odbywało się tylko w obrębie jednego wiersza. Obrazy po procesie rektyfikacji mają być w tej samej płaszczyźnie (współpłaszczyznowe), a linie epipolarne na obu obrazach mają pokrywać się z korespondującymi ze sobą wierszami na jednym i drugim obrazie.

Wybór wspólnej przestrzeni do której zostaną sprowadzone obrazy zależy od algorytmu rektyfikującego. Algorytmów służących do rektyfikacji jest wiele, w tej pracy zastosowane zostały dwa algorytmy, jeden służy do rektyfikacji obrazów nieskalibrowanych, a sama rektyfikacja odbywa się przy pomocy macierzy podstawowej,

drugi algorytm rektyfikuje obrazy na podstawie informacji o rotacji i przesunięciu między kamerami w systemie, a zatem przed przystąpieniem do takiej rektyfikacji, kamery muszą zostać skalibrowane.

Rektyfikacja bez uprzedniej kalibracji kamer

Rektyfikacja obrazów pochodzących z nieskalibrowanych kamer odbywa się za pomocą algorytmu Hartley’a [Hartley98]. Sam algorytm nie potrzebuje żadnej wiedzy na temat kamer, poza macierzą podstawową F . Do rektyfikacji służą odpowiadające sobie punkty na obu obrazach.

Algorytm znajduje takie przekształcenie rzutowego odzwierciedlenia dzięki któremu epipola na obu obrazach zostaną rozsunięte do nieskończoności, jednocześnie minimalizując rozbieżności między obrazami. Dzięki takiemu podejściu możemy ominąć konieczność wyliczania parametrów wewnętrznych kamer. Algorytm Hartley’a stosowany jest w systemach stereo wymagających rektyfikacji ”online” ponieważ wystarczającą informacją są punkty widziane przez kamery.

Poszukiwanie macierzy podstawowej

Macierz podstawowa może być znaleziona bez potrzeby uprzedniego kalibrowania kamer, wystarczającą informacją jest lista korespondujących ze sobą punktów na lewym i prawym obrazie. Posiadając listę punktów algorytm poszukuje macierzy podstawowej zgodnie ze wzorem (2.22). Dokładny opis procedury poszukiwania macierzy podstawowej znajduje się w [Bradski08].

Proponowany algorytm może wykorzystać jedno z trzech podejść przy znajdowaniu macierzy podstawowej. Wszystkie one działają mniej więcej w ten sam sposób wykorzystując wzór (2.22).

- Algorytm 8-punktów: rozwiązuje liniowy układ równań, metodą najmniejszych kwadratów minimalizuje błąd pomiędzy wszystkimi punktami.
- Algorytm RANSAC: rozwiązuje ten sam problem wiele razy na podstawie losowo wybranych punktów z całego asortymentu korespondujących ze sobą punktów na obu obrazach. Następnie wybierane jest jedno rozwiązanie, najbliższe średniej lub medianie liczonej ze wszystkich rozwiązań.
- Algorytm LMedS: na podstawie losowo wybranych punktów z całej puli oblicza rozwiązanie, następnie dodaje wszystkie inne punkty z całej puli które są spójne z wyliczonym na początku rozwiązaniem.

Macierz podstawowa F zwrócona przez opisywany algorytm jest zawsze macierzą o wymiarach 3×3 i może zostać bezpośrednio podana do algorytmu Hartley’a.

Algorytm Hartley’a

Zakładając, że macierz podstawowa została odpowiednio policzona, algorytm Hartley’a [Hartley98] działa następująco:

1. Używając macierzy podstawowej wyliczane są epipola korzystając ze wzorów:

$$Fe_l = 0 \quad i \quad (e_r)^T F = 0$$

2. Szukana jest jedna z dwóch macierzy przekształcenia rzutowego odzwierciedlenia, macierz ta musi być dobrana w taki sposób aby po przekształceniu obrazu prawe epipole znajdowało się w nieskończoności. Zgodnie z [Hartley98] macierz taka wygląda następująco:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{s} & 0 & 1 \end{bmatrix}$$

gdzie s jest dowolnym współczynnikiem skalującym.

3. Dodatkowo należy wybrać punkt na prawym obrazie w którym zniekształcenia będą minimalne. Ten punkt to początek układu współrzędnych związany z obrazem. Szukany jest taki wektor przesunięcia T i macierz rotacji R dzięki którym wybrany punkt będzie w początku układu współrzędnych. Zgodnie z [Hartley98] poszukiwana macierz odpowiedzialna za operację rzutowego odzwierciedlenia będzie wyglądała następująco:

$$H_r = GRT$$

4. Poszukiwana jest druga macierz H_l odpowiedzialna za rzutowe odzwierciedlenie, tym razem po przekształceniu obrazu lewe epipole musi znajdować się w nieskończoności. Dodatkowo przekształcenie musi dopasować do siebie wiersze na obu obrazach używając faktu, że dopasowanie wierszy minimalizuje odległości dla wszystkich korespondujących ze sobą punktów.

Takie dwie operacje rzutowego odzwierciedlenia na prawym i lewym obrazie definiują rektyfikację, obrazy po zastosowaniu opisywanych przekształceń powinny być współpłaszczyznowe, a linie epipolarne powinny znajdować się w korespondujących ze sobą wierszach.

Rektyfikacja poprzedzona uprzednią kalibracją kamer

Rektyfikacja algorytmem Bouguet'a⁸ wymaga uprzednio skalibrowanego systemu stereowizyjnego, w szczególności rektyfikacja wykonywana jest na podstawie macierzy rotacji R oraz wektora przesunięcia T między kamerami. Oba te parametry szukane są w procesie kalibracji systemu dwóch kamer, opisywanego w sekcji poświęconej takiej kalibracji. Ponieważ do poprawnego działania algorytmu potrzebne są skalibrowane kamery, cały proces trwa znacznie dłużej niż w przypadku rektyfikacji algorytmem Hartley'a i wymaga obiektu przestrzennego o znanych kształtach i rozmiarach.

Algorytm minimalizuje zniekształcenia i jednocześnie maksymalizuje wspólne pole widzenia na obrazach z obu kamer.

⁸Algorytm Bouguet'a jest zbiorem i uproszczeniem rozwiązań zaprezentowanych na początku w [Zhang98] i [Zhang99], Jean-Yves Bouguet nigdy nie opublikował tego algorytmu poza stroną internetową [Bouguet]. Algorytm został opisany w [Bradski08]

Algorytm Bouguet'a

W celu zminimalizowania zniekształceń macierz rotacji R obracająca prawą płaszczyznę obrazu w płaszczyznę lewego obrazu została podzielona na dwie macierze rotacji r_l oraz r_r . Opisywana macierz powoduje to, że obrazy są współpłaszczyznowe, nie są one jednak dopasowane wierszami. Aby wyliczyć macierz rotacji R_{rect} która przekształci widoki w taki sposób aby epipola na lewym i prawym obrazie były w nieskończoności, a linie epipolarne pokrywały się z korespondującymi wierszami, tworzone są trzy wektory korzystając z wektora przesunięcia $T = [T_x \ T_y \ T_z]^T$ między kamierami:

- Pierwszy wektor wygląda następująco:

$$e_1 = \frac{T}{\|T\|};$$

- Drugi wektor jest ortogonalny do wektora e_1 :

$$e_2 = \frac{[-T_y \ T_x \ 0]^T}{\sqrt{T_x^2 + T_y^2}};$$

- Trzeci wektor jest ortogonalny do dwóch poprzednich wektorów i jest znajdowany za pomocą iloczynu wektorowego wektora e_1 z wektorem e_2 .

$$e_3 = e_1 \times e_2;$$

Szukana macierz R_{rect} dzięki której po przekształceniu rektyfikacyjnym lewe epipole będzie w nieskończoności, a linie epipolarne staną się liniami poziomymi, wygląda następująco:

$$R_{rect} = \begin{bmatrix} (e_1)^T \\ (e_2)^T \\ (e_3)^T \end{bmatrix};$$

Dopasowanie wierszami obrazów pochodzących z lewej i prawej kamery jest osiągnięte stosując dwa poniższe wzory:

$$R_l = R_{rect}r_l;$$

$$R_r = R_{rect}r_r;$$

Algorytm wylicza również zrekyfikowane macierze wewnętrzne $M_{rect,l}$ oraz $M_{rect,r}$ odpowiednio dla lewej i prawej kamery. Zwrócone macierze są jednak przemnożone przez macierze projekcji dla lewego i prawego obrazu P_l i P_r .

$$P_l = M_{rect,l}P'_l = \begin{bmatrix} f_{x,l} & \alpha_l & c_{x,l} \\ 0 & f_{y,l} & c_{y,l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$P_r = M_{rect,r}P'_r = \begin{bmatrix} f_{x,r} & \alpha_r & c_{x,r} \\ 0 & f_{y,r} & c_{y,r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Macierze projekcji przekształcają punkty ze świata trójwymiarowego, we współrzędnych jednorodnych, na punkty świata dwuwymiarowego również, we współrzędnych jednorodnych, w sposób:

$$P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

Macierz reprojektacji dzięki której możliwe jest określenie wzajemnych zależności między punktami w świecie trójwymiarowym wygląda następująco:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{(c_x - c'_x)}{T_x} \end{bmatrix}$$

Wszystkie parametry w powyższej macierzy pochodzą od lewego obrazu, z wyjątkiem parametru c'_x który jest współrzędną x punktu głównego na prawym obrazie.

Posiadając punkty w przestrzeni dwuwymiarowej (we współrzędnych jednorodnych) oraz rozbieżności między nimi możemy, korzystając z poniższej formuły, określić położenie tych punktów w przestrzeni trójwymiarowej:

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

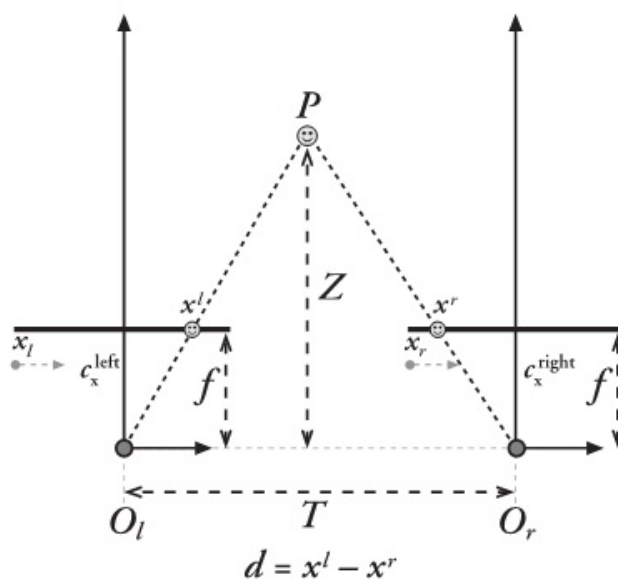
Zastosowanie algorytmu Bouguet'a dla systemu stereowizyjnego prowadzi do uzyskania idealnie skonfigurowanego systemu stereo, jak na rysunku (rys. 2.8).

Cały proces kalibracji i rektyfikacji ma więc za zadanie doprowadzenie obrazów do postaci pozbawionej zniekształceń i takich na których korespondujące punkty znajdują się w korespondujących wierszach, na prawym i lewym obrazie. Dodatkowo niektóre algorytmy rektyfikacyjne "wycinają" z lewego i prawego obrazu tylko przestrzeń wspólną, a więc taką która widziana jest przez obie kamery. Graficznie proces kalibracji i rektyfikacji pokazuje rysunek (rys. 2.9).

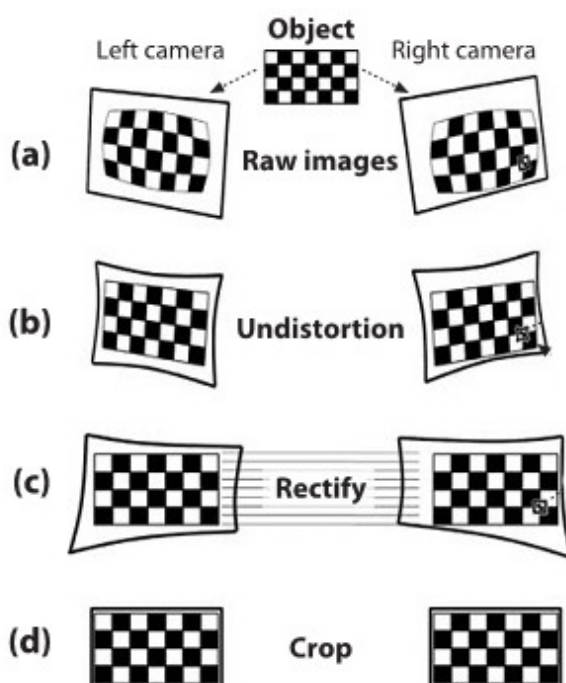
2.4 Właściwości kamer termowizyjnych

Kamera termowizyjna[Wiki04] to urządzenie wykorzystujące promieniowanie podczerwone[Wiki03] do formownia obrazu, w podobny sposób jak zwykła kamera formuje obraz w zakresie światła widzialnego.

Obrazy pochodzące z kamer nagrywających w podczerwieni z reguły są obrazami posiadającymi tylko jeden kanał koloru, ze względu na użycie matrycy która nie rozróżnia różnych długości fali promieniowania podczerwonego. Czasami monochromatyczne obrazy wyświetlane są w zakresie "pseudo koloru", gdzie wyświetlane są zmiany w kolorze zamiast zmian w nasileniu aby wyświetlić różnice w sygnale. W celu pomiaru temperatur najjaśniejsze (najcieplejsze) punkty na obrazach "maluje" się na białe. Temperatuty średnie na czerwono i żółto, natomiast najciemniejsze miejsca (najchłodniejsze) maluje się na niebiesko. Rozdzielczość w jakiej nagrywają typowe kamery na podczerwień jest znacznie niższa niż rozdzielczość



Rysunek 2.8: [Bradski08], idealny system stereowizyjny. punkt P to punkt rzeczywisty widziany przez dwie kamery, x^l oraz x^r to obrazy punktu odpowiednio na lewym i prawym obrazie, O_l i O_r to punkty główne, d to rozbieżność między punktami x^l i x^r .



Rysunek 2.9: [Bradski08], etapy przekształceń obrazów w procesie kalibracji i rektyfikacji.

w kamerach nagrywających w zakresie światła widzialnego. Jest to jedna z przyczyn dla której system stereowizyjny złożony z kamery podczerwonej i kamery pracującej w zakresie światła widzialnego jest trudny do zestawiania i rektyfikacji.

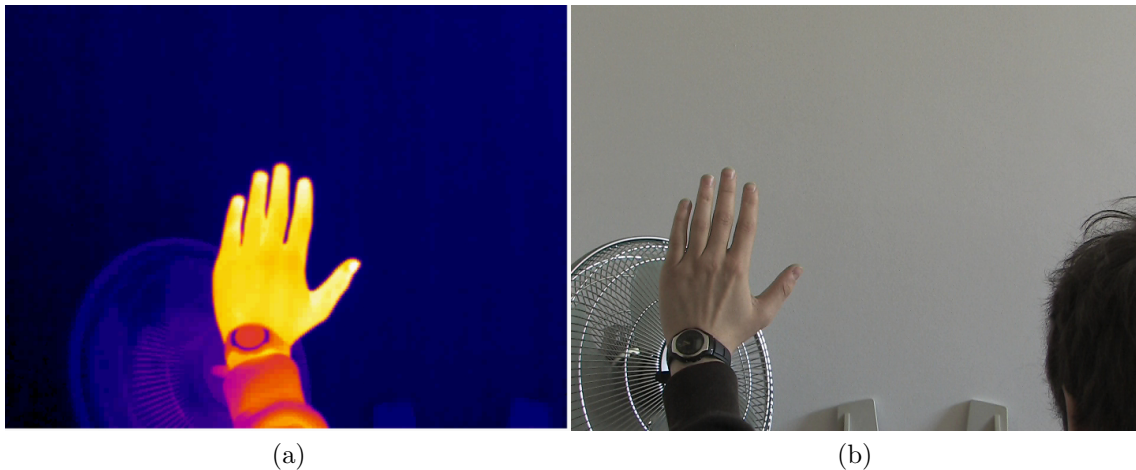
Kolejny problem to ilość klatek nagrywana przez kamery podczerwone i ilość klatek nagrywana przez kamery pracujące w zakresie światła widzialnego. Większość kamer podczerwonych nagrywa znacznie mniej klatek na sekundę niż kamery normalne, jest to spowodowane tym, że kamery nagrywające w zakresie podczerwieni służą głównie do bezkontaktowego pomiaru temperatury, nie jest więc wymagane aby taka kamera rejestrowała więcej niż 20 - 30 klatek na sekundę, bo żadne ciała nie zmienia swojej temperatury tak szybko. Przystępując więc do rektyfikacji obrazów z badanego systemu należy zastosować jakiś mechanizm który dopasuje do siebie sceny z jednej i drugiej kamery, wyrzucając przy tym nadmiarowe klatki zarejestrowane przez kamerę działającą w zakresie światła widzialnego.

Największy jednak problem który wprowadzają kamery podczerwone, mówiąc o rektyfikacji, to brak tekstur na obrazach pozyskanych kamerą podczerwoną (rys. 2.10).



Rysunek 2.10: [Wiki04], przykład obrazu pochodzącego z kamery pracującej w zakresie podczerwieni.

Wiele algorytmów stereowizyjnych pracuje na teksturach, np. podczas poszukiwania korespondujących ze sobą punktów. Brak tekstur na obrazach pochodzących z takiej kamery ogranicza w sposób znaczny pulę algorytmów jakie mogą zostać wykorzystane, nie tylko do rektyfikacji ale również do innych zastosowań. Skupiając się tylko na rektyfikacji, można dojść do wniosku, że żadna z opisywanych w poprzedniej sekcji technik nie da od razu dobrych wyników. Algorytm Hartley'a musi być zastosowany na punktach pochodzących z jednego i drugiego obrazu, ponieważ na obrazach z kamery pracującej w zakresie podczerwieni nie ma tekstur, to niemożliwym staje się użycie algorytmu który w automatyczny sposób znajdzie pasujące cechy na jednym i drugim obrazie. Na obrazie z teksturami liczba charakterystycznych cech będzie przynajmniej rząd wielkości większa niż liczba charakterystycznych cech pochodzących z obrazu termowizyjnego. Chcąc więc zastosować algorytm Hartley'a do badanego systemu stereo, należy albo opracować jakąś metodę dzięki której do algorytmu zostaną dostarczone odpowiadające sobie punkty, lub trzeba wybrać te punkty ręcznie.



Rysunek 2.11: Różnica między obrazem termowizyjnym a tradycyjnym, (a) lewa kamera, obraz w podczerwieni, (b) prawa kamera, obraz w zakresie światła widzialnego.

Z opisywanym wcześniej algorytmem Bouguet'a rektyfikacja wygląda jeszcze trudniej. Wcześniej powiedziane było, iż algorytm ten działa na obrazach z wcześniej skalibrowanych kamer, a urządzeniem dzięki któremu dokonuje się kalibracji jest najczęściej szachownica. Różnica w kolorze na obrazie termowizyjnym występuje wtedy gdy mamy różnicę w temperaturze. Ponieważ białe i czarne pola szachownicy mają taką samą temperaturę, niemożliwe jest rozróżnienie ich od siebie na obrazie pochodzącym z kamery działającej w zakresie podczerwieni. Na obrazie z takiej kamery szachownica wygląda jak zwykła nie zapisana kartka. Użycie jakiegokolwiek innego przedmiotu niż szachownica, również nie daje wielkich korzyści, obrazy z każdej z kamer niosą ze sobą zupełnie inną informację, coś co jest widoczne na kamerze działającej w zakresie światła widzialnego nie będzie widoczne na obrazie termowizyjnym i na odwrót (rys 2.11). Do przeprowadzenia kalibracji, a co za tym idzie do rektyfikacji algorytmem Bouguet'a potrzebny jest jakiś przedmiot który na obu obrazach będzie wyglądał tak samo, a więc przedmiot w którym charakterystyczne punkty widziane przez kamerę tradycyjną wydzielają ciepło i dzięki temu mogą zostać zarejestrowane przez kamerę działającą w podczerwieni.

Rozdział 3

Praca własna

Rozdział ten opisuje praktyczną część pracy magisterskiej, stanowi on swego rodzaju dokumentację wykonanej pracy, niezbędnej do zrealizowania zadań przedstawionych w podrozdziale "Zakres pracy". W pierwszej części opisane zostały kamery oraz oprogramowanie użyte w celu zarejestrowania sekwencji wizyjnych. Dalsza część poświęcona jest wstępnej kalibracji badanego systemu stereo, w szczególności dopasowaniu rozdzielczości oraz wyboru klatek biorących udział w kalibracji i rektyfikacji. Następnie zostały pokazane i opisane wyniki rektyfikacji i kalibracji obrazów z dwóch takich samych aparatów działających w zakresie światła widzialnego. Ostatecznie zaprezentowane są techniki, dzięki którym możliwe było zastosowanie dwóch wymienionych algorytmów rektyfikujących dla obrazów z badanego systemu stereowizyjnego. Na końcu prezentowane są wyniki przeprowadzonego postępowania.

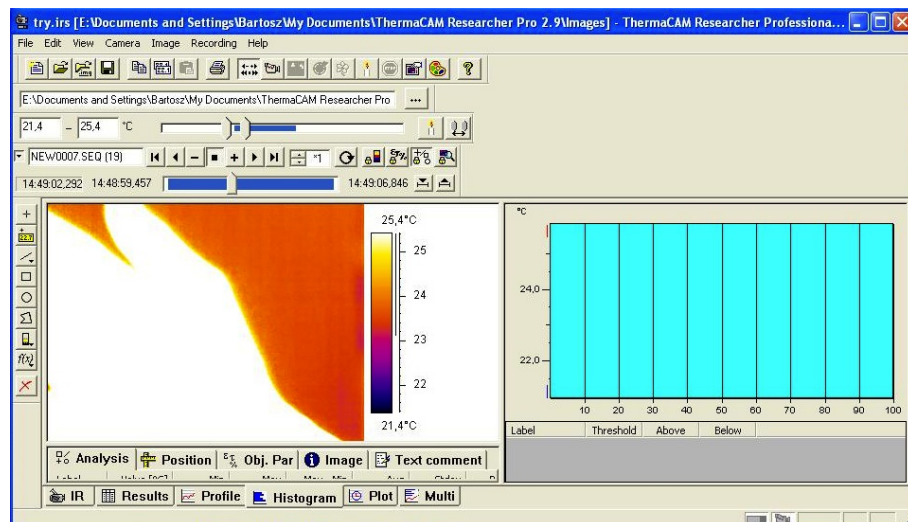
3.1 Badany system stereowizyjny

Sekcja opisuje badany system stereowizyjny wraz z użytym oprogramowaniem rejestrującym sekwencję z obu kamer. W tym miejscu również zwrócone zostały uwagi na temat właściwości rejestrowanych przez obie kamery sekwencji.

Badany system dwu-kamerowy składa się z kamery termowizyjnej i kamery działającej w zakresie światła widzialnego. Zestaw dwóch kamer zestawiony został w taki sposób aby kamera termowizyjna była lewą kamerą w systemie a kamera tradycyjna była kamerą prawą. Przy ustawianiu opisywanego zestawu stereo zwrócono szczególną uwagę na to aby kamery były możliwie jak najlepiej dopasowane do siebie mechanicznie. Wysokość obu kamer powinna być zbliżona, a w idealnym przypadku taka sama. Ze względu na różnice w obiektywach dopasowano przybliżenie tak aby obszary rejestrowane przez kamery były jak najbardziej do siebie zbliżone. Opisywany zestaw został również ustawiony w taki sposób aby płaszczyzny obrazów rejestrowanych przez obie kamery przecinały się pod jak najmniejszym kątem, w idealnym przypadku obrazy powinny być współpłaszczyznowe.

3.1.1 Kamera termowizyjna

Kamera termowizyjna użyta została jako lewa kamera zestawu stereowizyjnego. Dokładny model użytej kamery to *FLIR A300*. Kamera połączona została z komputerem za pomocą interfejsu sieciowego. Oprogramowanie jakie zostało użyte podczas nagrywania sekwencji to *ThermaCAM Researcher Pro* w wersji 2.9 (rys 3.1).



Rysunek 3.1: Program ThermoCAM Researcher Pro wersja 2.9.

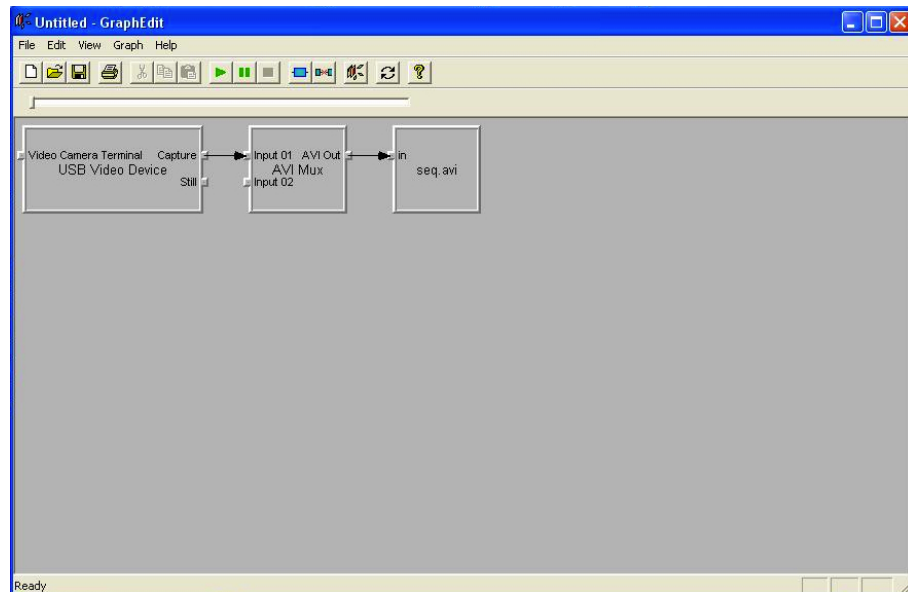
Opisywany zestaw (kamera wraz z oprogramowaniem nagrywającym) umożliwia rejestrowanie obrazów w zakresie światła podczerwonego w rozdzielczości 564×360 pikseli. Częstotliwość z jaką rejestrowana jest sekwencja wynosi 5 klatek na sekundę. Użyte oprogramowanie nie pozwalało na nagranie sekwencji bez paska ilustrującego zakres temperatur rejestrowanych przez kamerę w trakcie nagrywania (rys. 3.2).

Rysunek 3.2: Obraz nagrany kamerą termowizyjną, rozdzielczość 564×360 , pasek ilustrujący zakres temperatur rejestrowanych przez kamerę znajduje się po prawej stronie.

Sekwencje nagrywane przez opisywaną kamerę nie były w żaden sposób kompresowane i zapisywane były w postaci plików filmowych o rozszerzeniu ".avi".

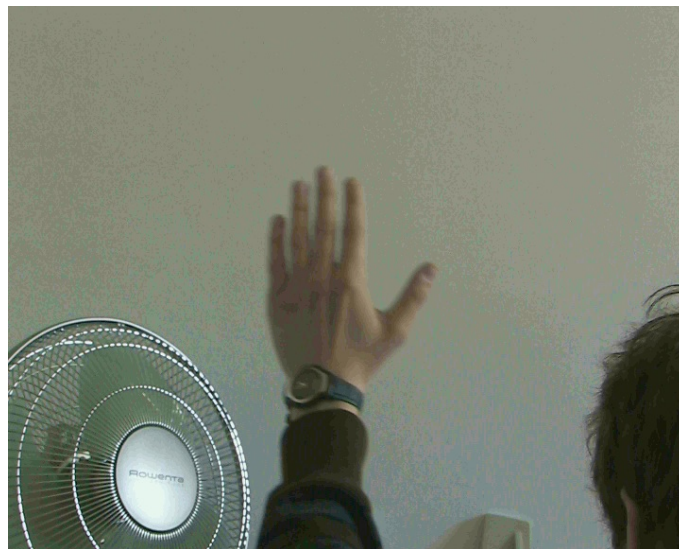
3.1.2 Kamera tradycyjna

Kamera rejestrująca w zakresie światła widzialnego została użyta jako prawa kamera badanego systemu stereo. Dokładny model kamery to Canon XHG1. Kamera została połączona z komputerem za pomocą interfejsu *IEEE 1394*. Do rejestracji obrazu opisywaną kamerą został użyty program *GraphEdit* (rys. 3.3) oraz filtry *DirectShow* pozwalające na rozkodowanie sekwencji z kamery i zapisanie jej na dysk.



Rysunek 3.3: Program GraphEdit.

Opisywany zestaw umożliwia rejestrowanie obrazów w zakresie światła widzialnego w rozdzielczości 720×576 pikseli. Częstotliwość z jaką rejestrowana jest sekwencja wynosi 25 klatek na sekundę. Przykład obrazu pochodzącego z kamery tradycyjnej pokazuje obrazek (rys 3.4).

Rysunek 3.4: Obraz nagrany kamerą tradycyjną w rozdzielczości 720×576 pikseli.

Sekwencja została zapisana na dysk twardy za pomocą standardowego filtru *FileWriter* pochodzącego z biblioteki *DirectShow*. Do zapisania sekwencji nie został użyty żaden kodek, a więc sekwencja wyjściowa nie została w żaden sposób skompresowana. Sekwencja została zapisana w pliku o rozszerzeniu ".avi".

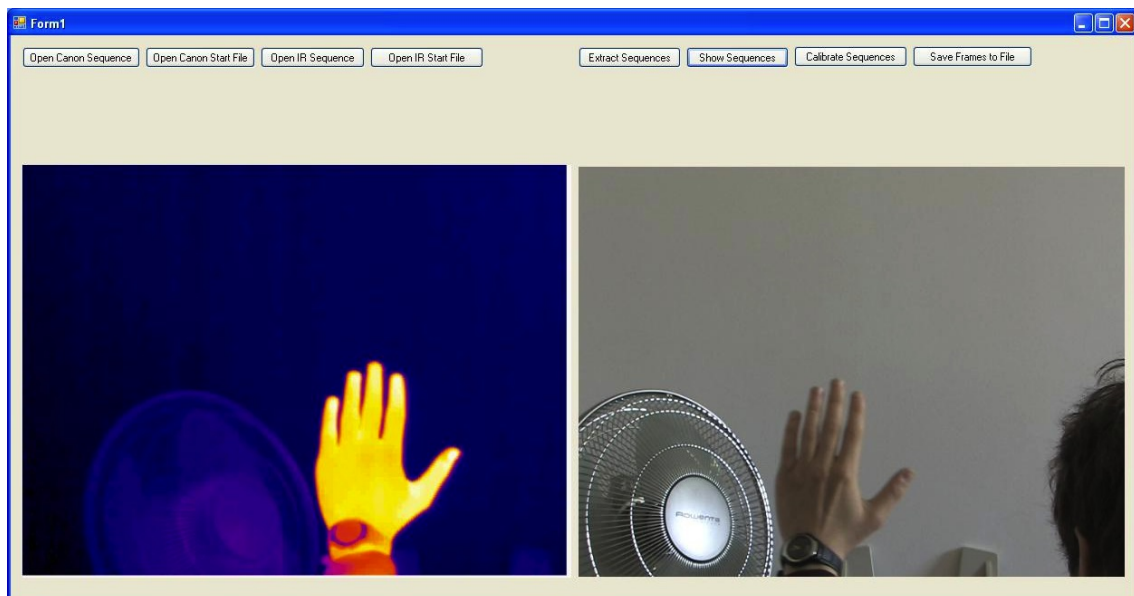
3.2 Wstępna kalibracja

Ze względu na różnice we właściwościach obrazów pochodzących z dwóch różnych kamer nie możliwa jest bezpośrednia kalibracja i rektyfikacja algorytmami przedstawionymi w poprzednim rozdziale. Aby przeprowadzić analizę i znaleźć wyniki rektyfikacji dla badanego systemu należy najpierw wstępnie dopasować obrazy z obu kamer do siebie. Dopasowanie musi nastąpić na dwóch płaszczyznach:

- Dopasowanie rozdzielczości obrazów pochodzących z nagranych kamerami sekwencji.
- Wybór obrazów rejestrowanych w tym samym czasie, inaczej mówiąc wybór odpowiadających sobie klatek z sekwencji nagranych jedną i drugą kamerą.

Dodatkowo z obrazów nagranych kamerą termowizyjną należy usunąć pasek ilustrujący zakres temperatur rejestrowanych przez kamerę w trakcie nagrywania sekwencji.

Dla potrzeb wstępnej kalibracji kamer stworzono program (rys. 3.5) w środowisku *.NET*, posługując się językiem *C#*, przy użyciu darmowej biblioteki przetwarzania obrazów *AForge.NET* w wersji 2.1.2[AForge].



Rysunek 3.5: Okno programu do wstępnej kalibracji kamer.

Stworzony program posiada interfejs okienkowy, umożliwia załadowanie dwóch, nieskompresowanych sekwencji filmowych i za pomocą dodatkowych plików umożliwia dopasowanie rozdzielczości nagranych sekwencji oraz wybór odpowiednich (korespondujących ze sobą) klatek z obu sekwencji. Stworzona aplikacja jako parametry wejściowe przyjmuje dwie sekwencje filmowe (jedna nagrana kamerą termowizyjną, druga nagrana kamerą tradycyjną) oraz pliki tekstowe z zapisem czasów¹ rozpoczę-

¹Program *ThermaCam Researcher Pro* umożliwia zapisanie takiego pliku, aby otrzymać taki sam plik dla kamery tradycyjnej stworzono osobny filtr *DirectShow* który zapisuje czas rozpoczęcia nagrywania sekwencji przez kamerę tradycyjną.

cia nagrywania poszczególnych klatek w obu sekwencjach. Program odczytuje obie sekwencje klatka po klatce, zamienia rozdzielczości na takie same oraz usuwa niepotrzebne klatki zarejestrowane kamerą tradycyjną. Aplikacja umożliwia podgląd dwóch plików filmowych zarówno w postaci wejściowej jak i wyjściowej. Jako wyjście programu możliwe jest zapisanie nowych (skalibrowanych) sekwencji lub zapis wszystkich klatek w formie obrazów *.bmp*.

3.2.1 Konwersja sekwencji filmowej do obrazów

Dwie sekwencje filmowe są w pierwszym kroku rozkodowane i zapisane w postaci obrazów poszczególnych klatek w pamięci operacyjnej. Biblioteka *AForge.NET* udostępnia interfejs, dzięki któremu możliwe jest odtworzenie nieskompresowanej sekwencji oraz zapisanie każdej klatki do osobnego obrazu[AForge01]. Obrazy pochodzące z poszczególnych klatek mogą zostać przetworzone i zapisane na dysku w formacie plików *.bmp*, co umożliwia dalsze ich przetwarzanie w celu rektyfikacji. Funkcja odpowiedzialna za rozkodowanie sekwencji i zapisanie jej klatka po klatce do plików z obrazami pokazana jest na listingu (listing 3.1).

Na wejście funkcji podawana jest lokalizacja na dysku sekwencji filmowej do przetworzenia. Parametr wyjściowy to tablica zawierające poszczególne klatki.

Listing 3.1: Funkcja tworząca obrazy z nieskompresowanej sekwencji filmowej

```
public static void ExtractAviToBmp(string aviLocation, ref Bitmap[]
    frames)
{
    AVIReader reader = new AVIReader();
    reader.Open(aviLocation);

    frames = new Bitmap[reader.Length];

    int counter = 0;
    while (reader.Position - reader.Start < reader.Length)
        frames[counter++] = reader.GetNextFrame();

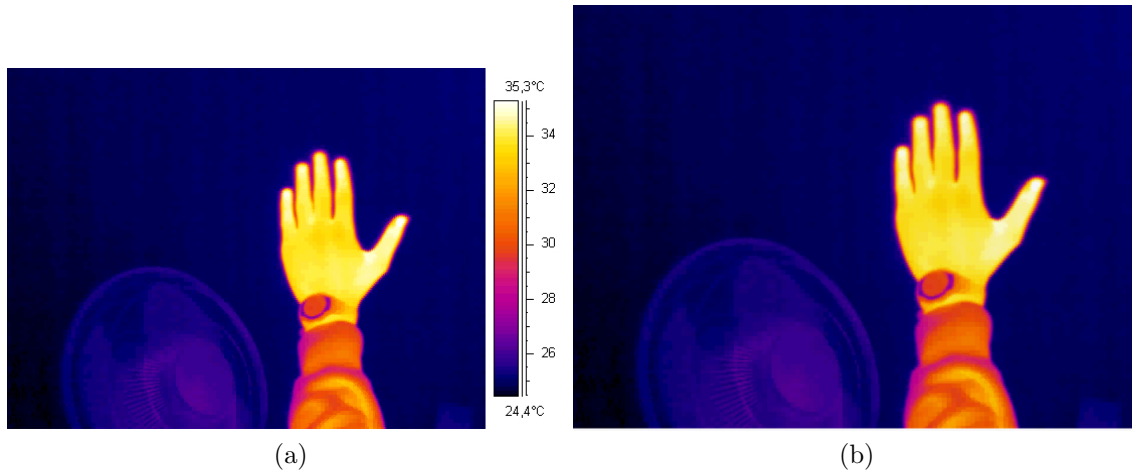
    reader.Close();
}
```

3.2.2 Dopasowanie rozdzielczości badanych obrazów

Posiadając dostęp do poszczególnych klatek z obu kamer można przystąpić do dopasowania rozdzielczości obu sekwencji. Rozdzielczości w jakich rejestrowane są sekwencje wynoszą:

- Rozdzielczość obrazu z kamery termowizyjnej - 564×360 pikseli.
- Rozdzielczość obrazu zarejestrowanego kamerą tradycyjną - 720×576 pikseli.

Stosunek wysokości do szerokości obrazu nie jest taki sam dla obydwu kamer. Nie możliwe jest więc dopasowanie rozdzielczości zmieniając parametry tylko jednej z zarejestrowanych sekwencji. Obraz z kamery działającej w zakresie światła podczerwonego jest mniejszy niż obraz pochodzący z kamery tradycyjnej. Potrzebne jest zatem zwiększenie rozdzielczości obrazu termowizyjnego.



Rysunek 3.6: Wynik zmiany rozdzielczości dla sekwencji w podczerwieni, (a) obraz wejściowy, (b) obraz wyjściowy.

Sekwencja w podczerwieni

Pierwszym wykonanym krokiem przetwarzania obrazu sekwencji termowizyjnej jest usunięcie paska ilustrującego zakres mierzonych temperatur. Pasek ten na każdej klatce jest stałej szerokości która wynosi 84 piksele. Obraz przetworzony jest za pomocą funkcji *Crop*[AForge01] z biblioteki *AForge.NET*, dzięki której wycięty zostaje niepotrzebny pasek, a obrazy termowizyjne zmieniają rozdzielczość na 480×360 pikseli. Następnie wykorzystując algorytm interpolacji dwuliniowej[AForge01] zmieniana jest rozdzielczość wszystkich podczerwonych klatek na obrazy o rozdzielczości 720×540 pikseli. Rysunek (rys. 3.6) przedstawia obraz wejściowy i wyjściowy, a listing (listing 3.2) pokazuje kod przekształcający obrazy.

Listing 3.2: Kod przekształcający klatki z sekwencji termowizyjnej.

```
// Operations required to calibrate ir sequence
for (int i = 0; i < framesIR.Length; i++)
    ImageOperations.CropImage(ref framesIR[i], 0, 0, 480, framesIR
        [0].Height);
for (int i = 0; i < framesIR.Length; i++)
    ImageOperations.ResizeImage(ref framesIR[i], 720, 540);
```

Po przekształceniach sekwencji podczerwonej szerokość obrazów z obu kamer jest już dopasowana. Obrazy z kamery termowizyjnej nie wymagają żadnych dalszych przekształceń.

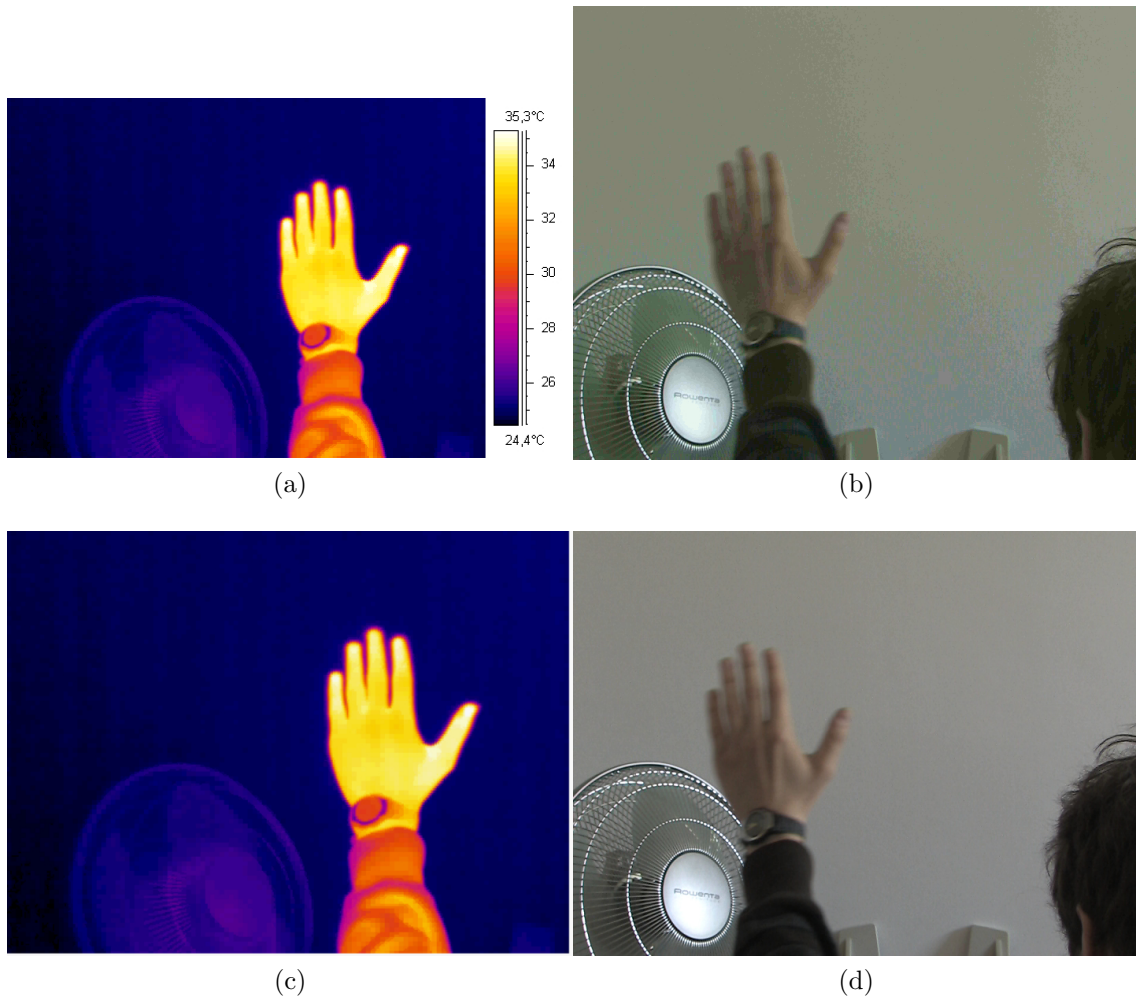
Sekwencja w zakresie światła widzialnego

Klatki pochodzące z sekwencji nagranej kamerą tradycyjną mają za dużą wysokość w stosunku do obrazów pochodzących z kamery termowizyjnej. Aby ostatecznie dopasować rozdzielczości badanych obrazów należy wyciąć nadmiarową część z obrazu tradycyjnej kamery, za pomocą funkcji *Crop*, tak aby pola filmowane przez obie kamery były jak najbardziej do siebie zbliżone. Kod przekształcający obraz z kamery działającej w zakresie światła widzialnego pokazany jest na listingu (listing 3.3).

Listing 3.3: Kod przekształcający klatki z sekwencji nagranej kamerą tradycyjną.

```
// Operations required to calibrate canon sequence
for (int i = 0; i < framesCanon.Length; i++)
    ImageOperations.CropImage(ref framesCanon[i], 0, 36,
                              framesCanon[0].Width, 540);
```

Ostateczne wyniki dopasowania rozdzielczości obrazów z badanego systemu stereowizyjnego pokazane są rysunku (rys. 3.7).



Rysunek 3.7: Ostateczne wyniki dopasowania rozdzielczości obrazów, (a) obraz termowizyjny przed przekształceniami, (b) obraz tradycyjny przed przekształceniami, (c) obraz termowizyjny po przekształceniach, (d) obraz tradycyjny po przekształceniach.

3.2.3 Wybór korespondujących ze sobą klatek

Kamery tworzące badany system stereowizyjny nagrywają sekwencję z inną częstotliwością rejestrowania klatek. Algorytmy stereowizyjne poszukują korespondujących ze sobą punktów na obu obrazach. Aby odpowiadające sobie punkty zostały znalezione, a odległość między nimi na lewym i prawym obrazie odzwierciedlała

świat rzeczywisty, należy przetwarzać tylko obrazy które zostały zarejestrowane przez obie kamery w tej samej chwili czasu. W celu przeprowadzania analizy badanego systemu konieczna jest pewność, że badane obrazy zostały zarejestrowane dokładnie w tej samej chwili, w szczególności jeżeli świat rejestrowany przez kamery jest w ruchu. Częstotliwość nagrywania kamery tradycyjnej jest wielokrotnością częstotliwości z jaką nagrywa kamera działająca w zakresie fal podczerwonych. W celu dalszej analizy obrazów pod kątem rektyfikacji konieczne jest dopasowanie do siebie pasujących klatek z obu sekwencji.

Kamera działająca w zakresie podczerwieni

Program *ThermaCAM Researcher Pro* umożliwia zapisanie sekwencji wizyjnej w formacie *Standard Archive Format* [Flir06]. Zapisując sekwencję w tym formacie otrzymujemy trzy pliki:

- *.stmov* - plik zawiera całą sekwencję zawartą w specjalnie przygotowanej (w języku *C++*) strukturze. Dzięki temu zapis filmowy może być bezpośrednio przetwarzany z poziomu języka programowania [Flir06].
- *.inc* - zawiera informację na temat struktury w jakiej została zapisana sekwencja.
- *.pod* - zawiera podstawowe informacje na temat nagranej sekwencji, takie jak rozdzielczość obrazu, częstotliwość nagrania oraz dokładny czas zapisu poszczególnych klatek w całej sekwencji.

Ze względu na zawartość ostatniego pliku, sekwencję nagrane za pomocą kamery termowizyjnej zapisywane były w formatach *.avi* oraz *Standard Archive Format*, a plik *.pod* został bezpośrednio wykorzystany do wyszukania pasujących do siebie klatek zarejestrowanych przez obie kamery. Format w jakim zapisywany jest czas rozpoczęcia każdej klatki pokazany jest na listingu (listing 3.4).

Listing 3.4: Przykładowy plik *.pod* zawierający czasy rejestracji poszczególnych klatek.

000000	12:09:01.366
000001	12:09:01.560
000002	12:09:01.898
000003	12:09:02.093
000004	12:09:02.392
000005	12:09:02.593
000006	12:09:02.871
000007	12:09:03.117
000008	12:09:03.339

...

Kamera działająca w zakresie światła widzialnego

Program *GraphEdit*, który służył do nagrania sekwencji kamerą tradycyjną, nie udostępnia żadnego mechanizmu, za pomocą którego możliwa byłaby rejestracja czasu, w którym rozpoczęto nagrywanie. Możliwe jest natomiast stworzenie filtra

DirectShow, który zarejestruje moment rozpoczęcia nagrywania i zapisze zarejestrowany czas do pliku.

Zgodnie z procedurą programowania filtrów *DirectShow* zaproponowanej w [Pec03], stworzono statyczny filtr, który rejestruje moment rozpoczęcia nagrywania sekwencji i zapisuje zarejestrowany czas do pliku tekstowego. Opisywany filtr został stworzony na podstawie przykładowego filtru, który jest zawartością *Windows SDK*.

Tworzenie i rejestracja filtru *DirectShow* w systemie

Stacyjny filtr o nazwie "Dump" stworzony został w środowisku programistycznym *Visual Studio 2008* na systemie operacyjnym *Windows XP* i użyty został w programie *GraphEdit*.

W celu stworzenia i kompilacji filtru pobrano najnowszą wersję *Windows SDK*. Nagłówki oraz biblioteki potrzebne do kompilacji znajdują się w katalogu *BaseClasses*, który zawiera się w pobranym *Windows SDK*. W Katalogu *BaseClasses* znajduje się plik programu *Visual Studio*. Przed przystąpieniem do kompilacji własnego filtru należy skompilować projekty wchodzące w skład katalogu *BaseClasses*.

W pobranym *SDK* zawiera się również szereg filtrów, które można użyć jako przykłady lub odniesienia przy tworzeniu własnego filtru. Do stworzenia opisywanego filtru posłużył przykładowy filtr o nazwie "Dump", wchodzący w skład *Windows SDK*. Przykładowy filtr pokazywał w jaki sposób możliwe jest zapisanie informacji na temat nagrywanej sekwencji filmowej do pliku tekstowego. Opisywany filtr stworzono zastępując funkcję zapisującą plik w przykładowym filtrze funkcją, która zapisuje czas rozpoczęcia nagrywania. Dodatkowo chodziło o stworzenie filtru statycznego, a więc takiego, który nie może zostać połączony z żadnym innym filtrem. Usunięto więc możliwość połączenia wejść i wyjść filtru z innymi filtrami. Po rozpoczęciu nagrywania sekwencji w programie *GraphEdit* wykonywany jest kod zawarty w listingu (listing 3.5).

Listing 3.5: Funkcja zapisująca czas rozpoczęcia nagrywania sekwencji tradycyjną kamerą.

```
SYSTEMTIME systemTime;
TCHAR output[10];

GetLocalTime(&systemTime);

int msValue = (int)systemTime.wMilliseconds;
int highBits = 0;
int lowBits = 0;

// store milliseconds to highBits and lowBits

char hours[5];
char minutes[5];
char seconds[5];
char cHighBits[5];
char cLowBits[5];

sprintf(hours, "%d", (int)systemTime.wHour);
sprintf(minutes, "%d", (int)systemTime.wMinute);
sprintf(seconds, "%d", (int)systemTime.wSecond);
sprintf(cHighBits, "%d", highBits);
sprintf(cLowBits, "%d", lowBits);
```

...

```

output[0] = hours[0];
output[1] = hours[1];
output[2] = minutes[0];
output[3] = minutes[1];
output[4] = seconds[0];
output[5] = seconds[1];
output[6] = cHighBits[0];
output[7] = cHighBits[1];
output[8] = cLowBits[0];
output[9] = cLowBits[1];

```

```
m_pDump->WriteString(output, sizeof output);
```

Filtr zapisuje na dysku tylko jeden plik, w którym zawarta jest tylko i wyłącznie informacja na temat czasu rozpoczęcia nagrywania sekwencji. Format zapisywanego czasu wygląda następująco:

godzina : minuta : sekunda : milisekunda

Podczas kompilacji filtru tworzony jest plik *Dump.dll*, który należy zarejestrować w systemie aby móc z niego skorzystać w środowisku *GraphEdit*. Rejestracja filtru w systemie *Windows XP* może zostać wykonana za pomocą polecenia:

regsvr32.exe "<ścieżka do pliku>\Dump.dll"

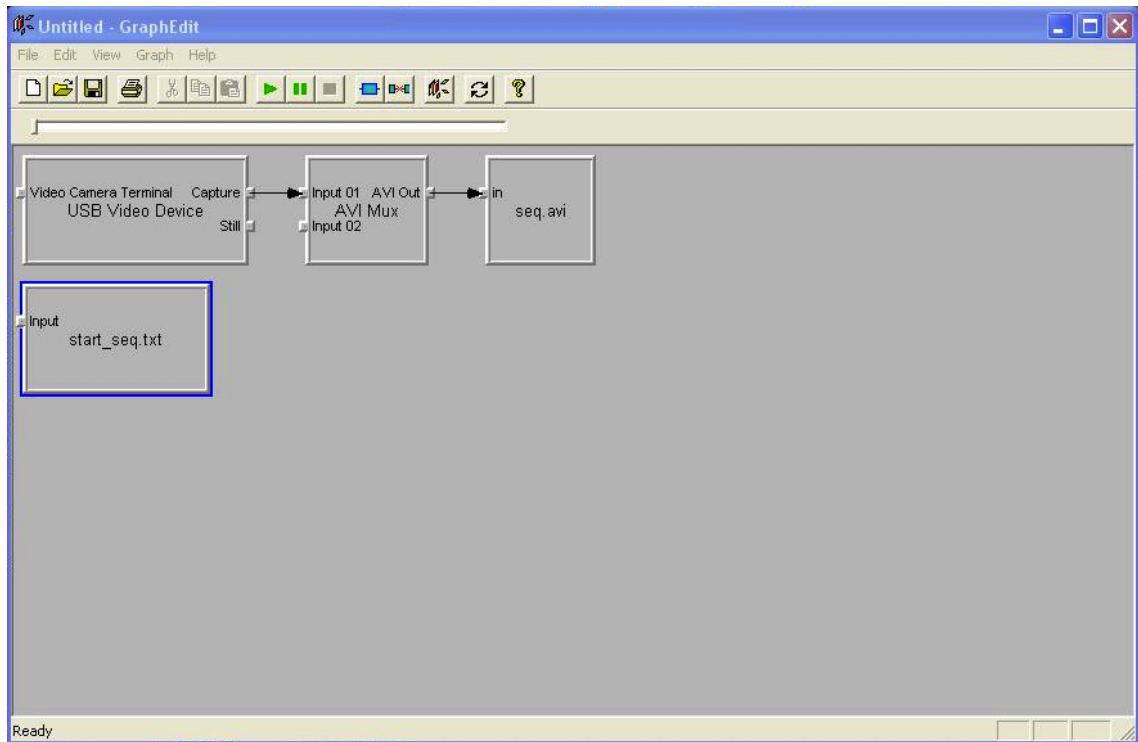
Używanie stworzonego filtru

Po prawidłowej rejestracji w systemie, filtr jest gotowy do użycia. W tym celu należy uruchomić program *GraphEdit*. Stworzony filtr zlokalizowany jest pod nazwą "*Dump*", w miejscu w którym wszystkie inne filtry *DirectShow*. W momencie dodawania filtru do grafu użytkownik automatycznie jest pytany w którym miejscu na dysku zapisać utworzony plik. Filtr pojawi się w oknie grafu nie połączony z żadnym innym filtrem. Wywołanie komendy *Play* uruchomi kod prezentowany w listingu (listing 3.5) i plik z czasem rozpoczęcia nagrywania zostanie zapisany we wskazanym wcześniej miejscu. Rysunek (rys. 3.8) ilustruje okno programu *GraphEdit* wraz ze stworzonym filtrem.

Program wybierający odpowiednie klatki

Funkcjonalność dzięki której wybierane są korespondujące ze sobą ramki została zaimplementowana w programie dokonującym wstępnej kalibracji. Program dostaje na wejściu dwa pliki, pierwszy to plik *.pod* utworzony za pomocą *ThermaCAM Researcher Pro*, drugi jest tworzony za pomocą opisywanego filtru *DirectShow*. Program wybierający odpowiednie klatki działa w następujący sposób:

1. W pierwszej części tworzone są dwie tablice po jednej dla każdej kamery. Tablice zawierają czasy rejestracji poszczególnych klatek. Tablice indeksowane są od zera, a indeks oznacza numer klatki. Odczytane czasy z obu plików wejściowych zamieniane są na milisekundy i zapisywane w tablicach. Dodatkowo dla kamery działającej w zakresie światła widzialnego potrzebny jest mechanizm



Rysunek 3.8: Program *GraphEdit* w konfiguracji pozwalającej nagranie sekwencji z kamery tradycyjnej, filtr "Dump" nie jest połączony z żadnym innym filtrem.

wyliczający czasy rejestracji wszystkich klatek poza pierwszą. Znając częstotliwość z jaką nagrywa kamera tradycyjna oraz czas pojawienia się pierwszej klatki, możliwe jest wyliczenie czasów rejestracji pozostałych klatek zgodnie ze wzorem:

$$t_i = t_s + i \cdot 40 \quad \text{gdzie} \quad i \in 1 \dots N$$

gdzie, t_i to czas rejestracji aktualnej ramki, t_s to czas rejestracji pierwszej ramki, N to ilość klatek w sekwencji.

Stworzone tablice przetwarzane są dalej w celu znalezienia odpowiadających sobie ramek.

2. Kamera tradycyjna rejestruje więcej klatek niż kamera termowizyjna, a więc z sekwencji nagranej kamerą działającą w zakresie światła widzialnego muszą zostać usunięte nadmiarowe klatki. W pierwszej kolejności wybierana jest pierwsza klatka z sekwencji tradycyjnej, która najlepiej pasuje do pierwszej klatki z sekwencji nagranej kamerą termowizyjną. W celu wybrania pierwszej klatki program w pętli sprawdza czasy pojawiania się kolejnych ramek w sekwencji nagranej kamerą zwykłą i porównuje te czasy z czasem rejestracji pierwszej klatki z sekwencji termowizyjnej. Klatka dla której różnica czasów jest najmniejsza uważana jest za pierwszą prawidłową klatkę w sekwencji (listing 3.6).

Listing 3.6: Wybór pierwszej klatki z sekwencji nagranej kamerą tradycyjną

```

for (int j = 0; j < canonList.Count; j++)
{
    long currentTimeDifference = Math.Abs(irList[0].second -
        canonList[j].second);
    if (currentTimeDifference < timeDifference)
    {
        timeDifference = currentTimeDifference;
        result = j;
    }
}
return result;

```

3. Posiadając tablice czasów, w których zarejestrowane zostały wszystkie klatki i wiedząc, która klatka z sekwencji nagranej kamerą tradycyjną jest pierwszą "dobrą" klatką program dopasowuje dalsze, korespondujące klatki do siebie. Dla każdej klatki z kamery termowizyjnej sprawdzanych jest 20 kolejnych klatek z sekwencji tradycyjnej. Kryterium wyboru odpowiednich klatek jest takie samo jak dla wyboru pierwszej klatki.

3.2.4 Program do wstępnej kalibracji

Okno aplikacji programu przeprowadzającego wstępną kalibrację pokazane jest na rysunku (rys. 3.5). Do poprawnego działania programu potrzebne są dwie sekwencje filmowe (jedna nagrana kamerą termowizyjną, druga kamerą tradycyjną) oraz dwa pliki zawierające czasy rejestracji sekwencji. Przyciski w lewym górnym rogu okna aplikacji pozwalają na załadowanie odpowiednich plików do programu.

- Po naciśnięciu przycisku "Open Canon Sequence" - wyskakuje okno dialogowe, które zapyta gdzie zlokalizowana jest sekwencja nagrana kamerą tradycyjną.
- Przycisk "Open Canon Start File" otwiera okno dialogowe, w którym należy wskazać plik tekstowy utworzony przez opisywany wcześniej filtr *DirectShow*.
- Przycisk "Open IR Sequence" otwiera okno dialogowe, gdzie należy wskazać lokalizację sekwencji z kamery termowizyjnej.
- "Open Ir Start File" otwiera okno z prośbą o wskazanie pliku *.pod* stworzonego przez program *ThermaCAM Researcher Pro*.

Z prawej strony okna opisywanej aplikacji znajdują się również cztery przyciski:

- Przycisk "Extract Sequences" rozkodowuje sekwencje z plików *.avi* i zapisuje każdą klatkę osobno w dwóch tablicach.
- "Show Sequences" umożliwia obejrzenie załadowanych sekwencji zarówno przed kalibracją jak i po niej.
- Przycisk "Calibrate Sequences" wykonuje wszystkie operacje wstępnej kalibracji opisane wcześniej. Aby nie przetwarzać niepotrzebnie ramek które i tak muszą zostać usunięte, najpierw przeprowadzane są operacje na plikach tekstowych w celu odpowiedniego dopasowania klatek. Dopasowanie rozdzielczości następuje w dalszej kolejności.

- Przycisk "Save Sequences to File" otwiera okno dialogowe w którym należy wskazać miejsce gdzie zostaną zapisane poszczególne klatki w postaci plików *.bmp*.

3.3 Kalibracja algorytmem Zhanga

Kalibracja pojedynczej kamery oraz algorytm Zhanga[Zhang98] opisane zostały w sekcji (2.1.5). Do przeprowadzenia kalibracji w sposób poprawny potrzebny jest obiekt w świecie rzeczywistym, którego wymiary są znane. Kalibracja przeprowadzana jest na podstawie punktów charakterystycznych na takim obiekcie. Odległości punktów charakterystycznych od siebie muszą być znane. Jako obiekt kalibracyjny została wybrana szachownica, a punkty charakterystyczne określone zostały jako wewnętrzne pola szachownicy.

W celu zniwelowania zniekształceń pochodzących z kamery i stworzenia obrazów pozbawionych tych zniekształceń przygotowano obiekt kalibracyjny - szachownicę o wymiarach 7×7^2 (rys 3.9), oraz oprogramowanie dzięki któremu możliwe było przeprowadzenia kalibracji i akwizycja wyników.



Rysunek 3.9: Przykładowe ujęcie szachownicy, biorące udział w procesie kalibracji.

3.3.1 Program kalibrujący

W celu kalibracji kamery stworzono program w języku *C++*, w środowisku *Visual Studio 2008*, przy pomocy biblioteki do przetwarzania obrazów *OpenCV*³. Program kalibrujący składa się z trzech mniejszych programów, każdy z nich może działać niezależnie od siebie i każdy wymaga osobnego pliku konfiguracyjnego. Pierwsza

²Szachownica o wymiarach 7×7 oznacza, że jest to szachownica na której w pionie i w poziomie ilość wewnętrznych narożników pól szachownicy wynosi 7.

³<http://opencv.willowgarage.com/wiki/>. Biblioteka funkcji wykorzystywanych podczas obróbki obrazu, oparta o otwarty kod i zapoczątkowana przez Intela.

aplikacja znajduje narożniki szachownicy i zapisuje je do pliku. Drugi program dokonuje kalibracji kamery na podstawie lokalizacji narożników szachownicy, zwróconych przez pierwszy program. Trzeci program jako parametry wejściowe przyjmuje macierz wewnętrzną i wektor zniekształceń zwrócone przez część kalibrującą i tworzy obrazy pozbawione zniekształceń.

Aby przeprowadzić cały proces kalibracji i otrzymać na wyjściu obrazy pozbawione zniekształceń należy uruchomić te trzy programy jeden za drugim.

Narożniki pól szachownicy

Program znajdujący narożniki pól szachownicy stworzony został w oparciu o funkcję *cvFindChessboardCorners()* z biblioteki *OpenCV*. Aplikacja do poprawnego działania potrzebuje pliku konfiguracyjnego, który zawiera takie informacje jak:

- Ilość obrazków zawierających szachownicę.
- Ścieżki do obrazów z szachownicami.
- Liczba wewnętrznych narożników pól szachownicy w jednym wierszu.
- Liczba wewnętrznych narożników pól szachownicy w jednej kolumnie.
- Informacja z jaką dokładnością mają zostać znalezione narożniki.
- Informacja czy zdjęcia ze znalezionymi narożnikami mają być pokazywane podczas działania programu.
- Informacja czy znalezione punkty mają zostać zapisane do plików tekstowych.
- Ścieżka do miejsca w którym mają zostać zapisane pliki wyjściowe.

Program wykonuje te same operacje dla każdego obrazka zawartego w pliku konfiguracyjnym. W pierwszej kolejności jest otwierany obrazek zawierający szachownicę. Następnie wywoływana jest funkcja *cvFindChessboardCorners()*, która lokalizuje narożniki pól szachownicy z dokładnością do jednego punktu obrazu. Następnie sprawdzana jest informacja na temat dokładności z jaką mają zostać znalezione narożniki. Jeżeli dokładność ma być większa niż jeden punkt obrazu, wywoływana jest dodatkowa funkcja która znajduje narożniki z większą dokładnością. Dalej jeżeli w pliku konfiguracyjnym znajduje się informacja mówiąca, że program powinien pokazywać znalezione narożniki w trakcie swojego działania, to obrazy ze znalezionymi narożnikami są wyświetlane (rys 3.10). Ostatecznie sprawdzany jest warunek mówiący o tym czy znalezione punkty mają zostać zapisane do odpowiednich plików. Jeżeli tak uruchomiona zostaje procedura zapisująca znalezione narożniki.

Program na wyjściu tworzy pliki zawierające lokalizację narożników pól szachownicy na poszczególnych obrazach. Plik wyjściowy powinien wyglądać jak na listingu (listing 3.7).



Rysunek 3.10: Obraz szachownicy z zaznaczonymi narożnikami, znalezionymi przez algorytm.

Listing 3.7: Postać pliku wyjściowego tworzonego przez program do znajdowania narożników pól szachownicy.

1482.46	136.148
1530.67	135.051
1578.78	133.477
1628.56	131.385
1678.4	129.81

...

Pliki tekstowe stworzone przez program są parametrami wejściowymi dla aplikacji przeprowadzającej kalibrację.

Parametry wewnętrzne i parametry zniekształceń

Kalibracja kamery to nic innego jak wyliczenie jej macierzy wewnętrznej oraz parametrów zniekształceń. Znając wszystkie właściwości można odtworzyć obrazy pozbawiając ich zniekształceń. Program został napisany przy pomocy funkcji *cvCalibrateCamera2()* pochodzącej z biblioteki *OpenCV*. Aplikacja wymaga na wejściu pliku konfiguracyjnego, który zawiera następujące parametry:

- Ilość obrazów z których pochodzą punkty znalezione przez poprzedni program.
- Ścieżka do plików w których zapisane są koordynaty narożników pól szachownicy, pliki zwrócone przez poprzedni program.
- Liczba wewnętrznych narożników pól szachownicy w jednym wierszu.
- Liczba wewnętrznych narożników pól szachownicy w jednej kolumnie.
- Szerokość w punktach obrazu.

- Wysokość w punktach obrazu.
- Informacja na temat dokładności z jaką poprzedni program znalazł narożniki.
- Informacja czy znaleziona macierz wewnętrzna i wektor zniekształceń mają zostać zapisane do plików *.xml*.
- Ścieżka do pliku *.xml* gdzie zostanie zapisana macierz wewnętrzna.
- Ścieżka do pliku *.xml* gdzie zostanie zapisany wektor zniekształceń.

Program odczytuje ścieżki do plików zawierających współrzędne narożników szachownicy i buduje tablice zawierającą informację na temat położenia wszystkich narożników. Stworzona tablica podawana jest do funkcji *cvCalibrateCamera2()* która wykonuje kalibrację według algorytmu Zhang’a [Zhang98] opisanego w sekcji (2.1.5). Następnie sprawdzane jest czy wyliczone podczas kalibracji macierz wewnętrzna i wektor zniekształceń mają zostać zapisane. W zależności od werdyktu uruchamiana jest procedura zapisująca wyliczone parametry do plików *.xml*.

Pliki w których zapisane są zwracane macierze zawierają takie parametry jak, ilość wierszy i kolumn macierzy, typ danych zapisany w macierzach, i wartość każdej komórki. Przykładowa macierz wewnętrzna pokazana jest na listingu (listing 3.8), wektor zniekształceń pokazany jest na listingu (listing 3.9).

Listing 3.8: Macierz podstawowa zapisana w pliku *.xml*.

```
<?xml version="1.0"?>
<opencv_storage>
<cameraMatrix type_id="opencv-matrix">
  <rows>3</rows>
  <cols>3</cols>
  <dt>f</dt>
  <data>
    9.30289429e+002 0. 3.28475952e+002 0. 9.34570801e+002
    2.54665466e+002 0. 0. 1.</data></cameraMatrix>
</opencv_storage>
```

Listing 3.9: Wektor zniekształceń zapisany w pliku *.xml*.

```
<?xml version="1.0"?>
<opencv_storage>
<distCoeffs type_id="opencv-matrix">
  <rows>5</rows>
  <cols>1</cols>
  <dt>f</dt>
  <data>
    -4.94901210e-001 2.55045199e+000 5.60549833e-003 -7.86983222e-003
    -1.00940552e+001</data></distCoeffs>
</opencv_storage>
```

Niwelacja zniekształceń

Znając macierz podstawową kamery oraz parametry zniekształcenia radialnego i tangensowego, można z obrazów oryginalnych (bezpośrednio z kamery) stworzyć

obrazy pozbawione zniekształceń. Program niwelujący zniekształcenia radialne i tangensowe w obrazach został napisany za pomocą funkcji *cvInitUndistortMap()* z biblioteki *OpenCV*. Tak jak pozostałe programy do poprawnego działania potrzebny jest plik konfiguracyjny który zawiera następujące informacje:

- Ilość obrazków które poddane zostaną procedurze pozbawienia zniekształceń.
- Ścieżki do obrazków które mają zostać przetworzone.
- Ścieżka do pliku *.xml* zawierającego macierz wewnętrzną kamery.
- Ścieżka do pliku *.xml* zawierającego wektor zniekształceń.
- Informacja czy przekształcone obrazy mają być pokazywane użytkownikowi podczas działania programu.
- Ścieżka do miejsca gdzie zostaną zapisane stworzone obrazy pozbawione zniekształceń.

Program na początku odczytuje macierz wewnętrzną i wektor zniekształceń z plików *.xml*. Za pomocą funkcji *cvInitUndistortMap()* tworzone są dwie macierze, które zawierają informacje na temat przekształcenia jakiemu ma zostać poddany każdy punkt obrazu wejściowego. Za pomocą wyliczonych macierzy oraz funkcji *cvRemap* z biblioteki *OpenCV* wykonywana jest procedura pozbawiania zniekształceń dla każdego obrazu zawartego w pliku konfiguracyjnym. Jeżeli plik konfiguracyjny zawiera taką informację obrazy są pokazywane użytkownikowi podczas działania programu. Na końcu przekształcone obrazy są zapisywane pod wskazane w pliku konfiguracyjnym miejsce.

3.3.2 Kalibracja w całości

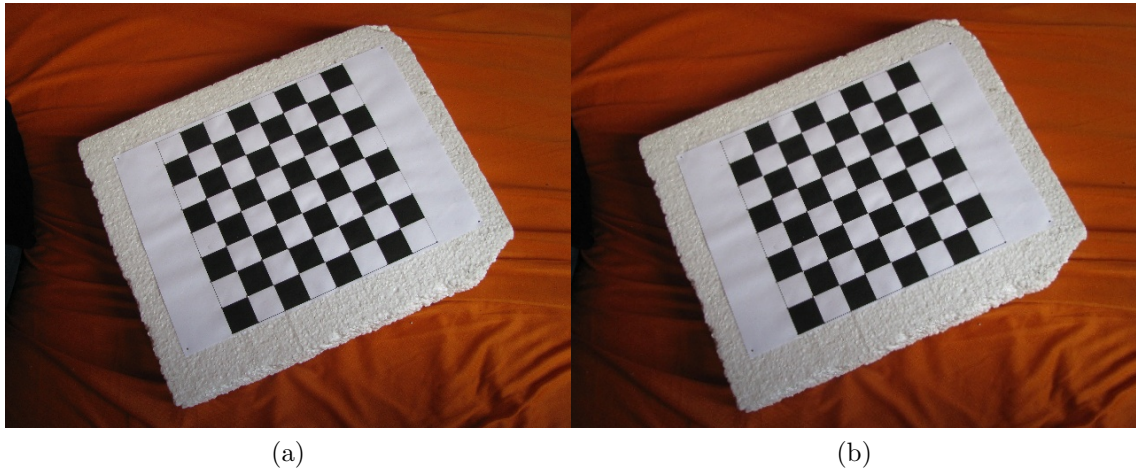
W celu przeprowadzenia kalibracji w całości, wszystkie trzy opisane wcześniej programy muszą zostać wykonane po kolei. Każdy z programów musi mieć stworzony specjalny plik konfiguracyjny ze wszystkimi opisanymi wcześniej parametrami. Aby uruchomić proces kalibracji w całości stworzono plik *.bat* uruchamiający wszystkie trzy programy jeden po drugim. Zawartość stworzonego pliku pokazana jest na listingu (listing 3.10).

Listing 3.10: Zawartość pliku *.bat* uruchamiającego wszystkie trzy funkcje biorące udział w kalibracji.

```
findChessboardCorners.exe findCorners_confFile.txt  
CameraCalibrate.exe cameraCalibrate_confFile.txt  
UndistortAfterCalibration.exe undistort_confFile.txt
```

3.3.3 Wyniki kalibracji

W celu sprawdzenia poprawności stworzonego programu przeprowadzono kalibrację dla aparatu fotograficznego Canon XHG1, na 11 ujęciach szachownicy, Szachownica na każdym zdjęciu znajdowała się w innym miejscu i w innej pozycji. Wyniki kalibracji pokazuje rysunek (rys. 3.11).



Rysunek 3.11: Wyniki kalibracji pojedynczego aparatu, (a) obraz oryginalny, (b) obraz pozbawiony zniekształceń.

3.4 Kalibracja i rektyfikacja systemu stereowizyjnego

Kalibracja i rektyfikacja systemu złożonego z dwóch kamer zostały opisane w rozdziale (2.3). Do przeprowadzenia kalibracji i rektyfikacji systemu stereowizyjnego potrzebny jest obiekt kalibrujący o znanych rozmiarach. Tak samo jak w rozdziale (3.3) jako obiekt kalibrujący wybrana została szachownica o wymiarach 7×7 (rys. 3.9). W celu kalibracji i rektyfikacji układu dwóch kamer stworzono program umożliwiający wybór algorytmu rektyfikacyjnego.

3.4.1 Program do kalibracji i rektyfikacji

Stworzone oprogramowanie zostało napisane w języku *C++*, w środowisku programistycznym *Visual Studio 2008*, przy pomocy biblioteki do przetwarzania obrazów *OpenCV*. Utworzony program składa się z trzech mniejszych programów, podobnie jak aplikacja opisywana w sekcji (3.3.1). Każdy ze stworzonych programów może działać niezależnie. Każdy program wymaga utworzenia specjalnego pliku konfiguracyjnego, za pomocą którego możliwe jest kontrolowanie działania programów. Pierwszą część programu stanowi aplikacja znajdująca i zapisująca położenie narożników pól szachownicy. Jest to dokładnie ten sam program co opisany w sekcji (3.3.1) dlatego nie będzie on opisywany w tej sekcji ponownie. Drugi program jest odpowiedzialny za kalibrację i rektyfikację technikami opisanymi w rozdziale (2.3). Trzecia aplikacja, na podstawie danych zwróconych przez poprzedni program, przekształca wejściowe obrazy w taki sposób aby na wyjściu obrazy były zrektyfikowane i gotowe do dalszych przekształceń stereowizyjnych. Aby przeprowadzić cały proces kalibracji i rektyfikacji należy uruchomić wszystkie trzy programy jeden po drugim. Program znajdujący narożniki szachownicy musi zostać wywołany dwa razy, raz dla prawej i raz dla lewej kamery.

Kalibracja z rektyfikacją

Na podstawie lokalizacji narożników pól szachownicy możliwa jest kalibracja systemu dwóch kamer. Rektyfikacja jest możliwa dzięki danym zwracanym przez część kalibrującą. Kalibracja systemu stereowizyjnego nie polega tylko na znalezieniu parametrów wewnętrznych oraz parametrów zniekształceń dla obu kamer, ale również na wyliczeniu macierzy rotacji i wektora przesunięcia między kamerami.

Rektyfikacja może zostać przeprowadzona na dwa sposoby. Dzięki stworzonemu programowi możliwy jest wybór algorytmu rektyfikującego - algorytm Bouguet'a [Bouguet], algorytm Hartley'a [Hartley98] oraz algorytm będący połączeniem tych dwóch metod. Algorytm Bouguet'a dokonuje rektyfikacji na podstawie macierzy rotacji i wektora przesunięcia zwróconych przez część kalibrującą. Algorytm Hartley'a rektyfikuje obrazy tylko na podstawie macierzy podstawowej, która wyliczana jest za pomocą lokalizacji korespondujących ze sobą punktów. Możliwe jest również aby macierz podstawowa została wyliczona w procedurze kalibracji. Taki algorytm który w części kalibrującej wylicza macierz podstawową, a sama rektyfikacja przeprowadzana jest algorytmem Hartley'a, jest trzecią możliwością przeprowadzenia rektyfikacji przez opisywany program.

Program kalibrujący i rektyfikujący wykorzystuje cztery funkcje z biblioteki *OpenCV*. Kalibracja systemu stereo wykorzystuje funkcje *cvStereoCalibrate()*, rektyfikacja algorytmem Bouguet'a przeprowadzana jest za pomocą funkcji *cvStereoRectify()*, za znalezienie macierzy podstawowej odpowiedzialna jest funkcja *cvFindFundamentalMat()*, a algorytm Hartley'a implementuje funkcja *cvStereoRectifyUncalibrated()*. Każda z czterech wymienionych funkcji potrzebuje odpowiedniej listy parametrów, większość z nich ustawiana za pomocą pliku konfiguracyjnego. Opcje nie dostępne w pliku konfiguracyjnym mogą być zmienione tylko z poziomu kodu programu, a do ich zatwierdzenia potrzebna jest ponowna kompilacja. Plik konfiguracyjny jest niezbędny do poprawnego działania programu, zawiera on następujące informacje:

- Liczba zdjęć która bierze udział w kalibracji i rektyfikacji.
- Ścieżki do plików zawierających lokalizacje punktów charakterystycznych na każdym zdjęciu, dla lewej i prawej kamery.
- Ilość narożników szachownicy w jednym wierszu i jednej kolumnie.
- Wymiary zdjęć biorących udział w kalibracji.
- Informacja na temat dokładności z jaką zapisana jest lokalizacja narożników pól szachownicy.
- Informacja czy ma zostać przeprowadzona kalibracja.
- Informacja czy przeprowadzona ma zostać rektyfikacja.
- Informacja na temat algorytmu rektyfikacyjnego, który ma zostać użyty.
- Informacja czy zapisane mają być wyniki kalibracji.
- Informacja czy zapisane mają zostać wyniki rektyfikacji.
- Ścieżki do plików *.xml*, w których mają zostać zapisane wyniki kalibracji.

- Ścieżki do plików *.xml*, w których mają zostać zapisane wyniki rektyfikacji.

Program odczytuje pliki z zapisaną lokalizacją narożników pól szachownicy na każdym zdjęciu. Współrzędne wszystkich punktów zapisywane są w jednej tablicy która podawana jest do funkcji *cvStereoCalibrate()*. Jeżeli plik konfiguracyjny wymusza kalibrację to jest ona dokonywana. Funkcja kalibrująca może zostać wywołana z dodatkowymi flagami, dzięki którym możliwe jest przeprowadzenia kalibracji bez estymacji wszystkich parametrów kamer. Flagi te mogą zostać zmienione z poziomu kodu i aby zmiana została zapisana potrzebna jest ponowna kompilacja programu. Funkcja *cvStereoCalibrate()* rozpoznaje następujące flagi:

- *CV_CALIB_FIX_INTRINSIC* - jeżeli jest ustawiona to algorytm nie wylicza parametrów wewnętrznych oraz parametrów zniekształceń.
- *CV_CALIB_USE_INTRINSIC_GUESS* - umożliwia podanie do funkcji parametrów wewnętrznych kamer, są one traktowane jako wartości startowe i podawane dalszej optymalizacji.
- *CV_CALIB_FIX_PRINCIPAL_POINT* - punkty główne obu kamer nie są wyliczane.
- *CV_CALIB_FIX_FOCAL_LENGTH* - ogniskowe obu kamer nie są wyliczane.
- *CV_CALIB_ZERO_TANGENT_DIST* - zniekształcenia tangensowe ustawione są na 0 i nie są wyliczane.

Po przeprowadzeniu kalibracji dokonywana jest rektyfikacja jeżeli plik konfiguracyjny zawiera informację o konieczności jej przeprowadzenia. Algorytm dokonujący rektyfikacji musi zostać sprecyzowany w pliku konfiguracyjnym. Jeżeli rektyfikacja ma zostać dokonana za pomocą algorytmu Hartley'a to uruchamiana jest procedura szukania macierzy podstawowej, zgodnie z procedurą opisaną w sekcji (2.3.2). Algorytm szukania macierzy podstawowej może zostać zmieniony z poziomu kodu, domyślnie ustawiony jest algorytm *RANSAC*. Po przeprowadzeniu rektyfikacji zapisywane są jej wyniki do odpowiednich plików *.xml*.

Przekształcenie rektyfikacyjne

Poprzedni program wylicza tylko macierze projekcji dla obu kamer, na podstawie których możliwe jest przekształcenie obrazów wejściowych do obrazów zrektyfikowanych. Aplikacja przekształcająca obrazy została napisana za pomocą *cvInitUndistortRectifyMap()* z biblioteki *OpenCV*. Podobnie jak inne programy aplikacja dokonująca przekształcenia rektyfikacyjnego wymaga pliku konfiguracyjnego, który zawiera następujące informacje:

- Liczba zdjęć, które biorą udział w kalibracji i rektyfikacji.
- Ścieżki do zdjęć, które mają zostać poddane przekształceniu rektyfikującemu dla prawej i lewej kamery.
- Ścieżki do plików *.xml* zawierających macierze zwrócone przez program rektyfikujący.

- Informacja na temat algorytmu, za pomocą którego dokonana została rektyfikacja.
- Informacja czy przekształcone obrazy mają zostać wyświetlane na ekranie podczas działania programu.
- Informacja czy obrazy wyjściowe mają zostać zapisane na dysku twardym.
- Ścieżki do miejsc, w których mają zostać zapisane zrektyfikowane obrazy z lewej i prawej kamery.

W pierwszej fazie programu odczytywane są macierze zwracane przez program rektyfikujący. Na ich podstawie wyliczane są macierze zawierające informacje na temat przekształcenia jakiemu musi zostać poddany każdy punkt z obrazów wejściowych. W zależności od algorytmu rektyfikacyjnego wybranego w poprzednim programie macierze przekształcające liczone są w różny sposób. Dalej każdy obraz wejściowy poddawany jest takiemu samemu przekształceniu. Sprawdzany jest warunek czy obrazy zrektyfikowane mają zostać pokazane na ekranie podczas działania programu. Ostatecznie przekształcone obrazy zapisywane są na dysk twardy w miejsce wskazane w pliku konfiguracyjnym.

3.4.2 Rektyfikacja w całości

Do przeprowadzenia kalibracji wraz z akwizycją wyników wszystkie trzy opisane programy muszą zostać uruchomione po kolei. Dla poprawnego działania każdego z programów potrzebne są pliki konfiguracyjne zawierające wszystkie wcześniej opisane parametry. Aby przeprowadzić całą rektyfikację stworzono plik *.bat* uruchamiający wszystkie trzy programy w odpowiedniej kolejności. Zawartość pliku pokazana jest na listingu (listing 3.11).

Listing 3.11: Zawartość pliku *.bat* przeprowadzającego rektyfikację. Wszystkie trzy funkcje uruchamiane w odpowiedniej kolejności.

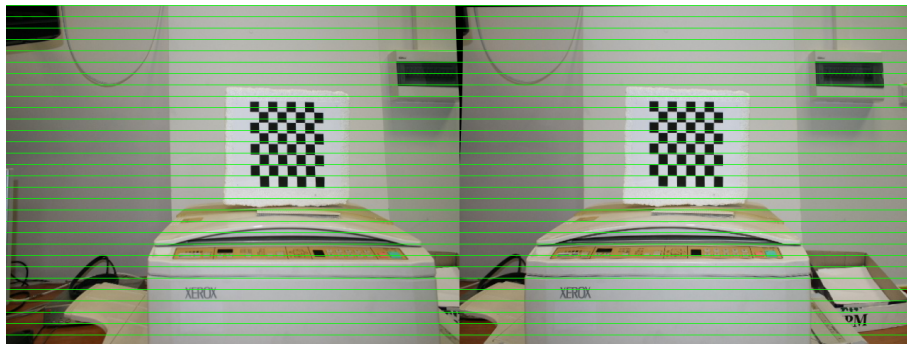
```
findChessboardCorners.exe configs\findLeftChessboard_confFile.txt  
findChessboardCorners.exe configs\findRightChessboard_confFile.txt  
StereoCalibrate.exe configs\stereoCalibrate_confFile.txt  
UndistortRectify.exe configs\remap_confFile.txt
```

3.4.3 Wyniki rektyfikacji

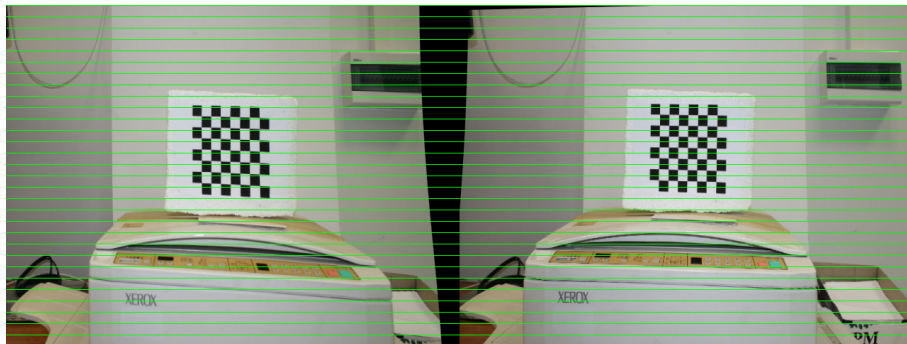
W celu sprawdzenia poprawności działania stworzonego programu zbudowano specjalny system stereowizyjny składający się z dwóch takich samych aparatów fotograficznych Canon 400D. Kalibracja wraz z rektyfikacją przeprowadzana została na 33 ujęciach szachownicy. Na każdym ze zdjęć szachownica znajdowała się w innym miejscu i innej pozycji. Wyniki rektyfikacji pokazuje rysunek (rys. 3.12).



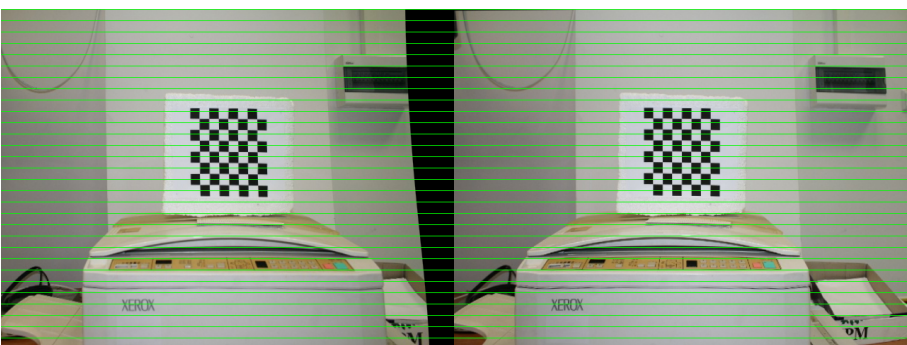
(a)



(b)



(c)



(d)

Rysunek 3.12: Wyniki rektyfikacji, (a) obrazy bezpośrednio pochodzące z kamer, (b) rektyfikacja algorytmem Bouguet'a, (c) rektyfikacja algorytmem Hartley'a, (d) rektyfikacja algorytmem Hartley'a, macierz podstawowa znaleziona w procesie kalibracji.

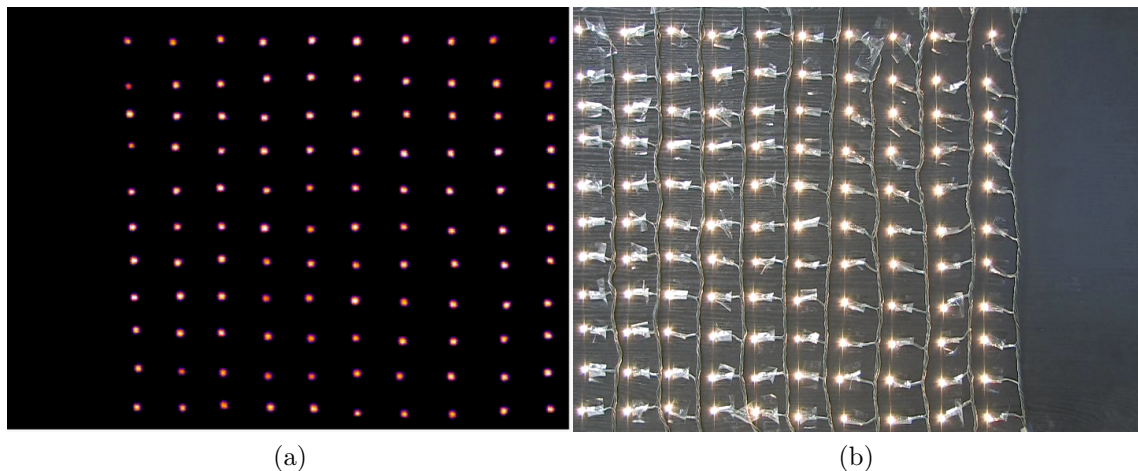
3.5 Rektyfikacja badanego systemu stereowizyjnego

Rektyfikacja badanego systemu dwóch kamer nie może zostać przeprowadzona w sposób bezpośredni, tak jak dla dwóch aparatów działających w zakresie światła widzialnego. Do zbadania systemu potrzebny jest obiekt rektyfikacyjny. W przypadku badanego systemu obiekt musi zostać wybrany w taki sposób aby punkty charakterystyczne zostały zarejestrowane przez obie kamery. Metoda Hartley'a została przebadana jako pierwsza, ponieważ łatwiejszy jest wybór obiektu dla tego algorytmu.

3.5.1 Metoda Hartley'a

Aby przeprowadzić rektyfikację zbudowano specjalny obiekt posługując się lampkami choinkowymi. Lampki choinkowe świecą jednostajnym światłem, więc są rejestrowane przez kamerę tradycyjną. Przy okazji świecenia lampki wydzielają również ciepło, mogą zatem zostać zarejestrowane przez kamerę podczerwoną.

Metoda rektyfikacji według algorytmu Hartley'a nie wymaga aby odległości punktów charakterystycznych od siebie były znane. Obiekt rektyfikacyjny zbudowany został z lampek choinkowych przyklejonych do płaskiej powierzchni (rys. 3.13). Podczas budowania obiektu postarano się aby rozkład lampek był równomierny, w każdym miejscu filmowanym przez obie kamery.



Rysunek 3.13: Obiekt służący do przeprowadzenia rektyfikacji algorytmem Hartley'a, (a) widok obiektu z kamery termowizyjnej, (b) widok obiektu z kamery tradycyjnej.

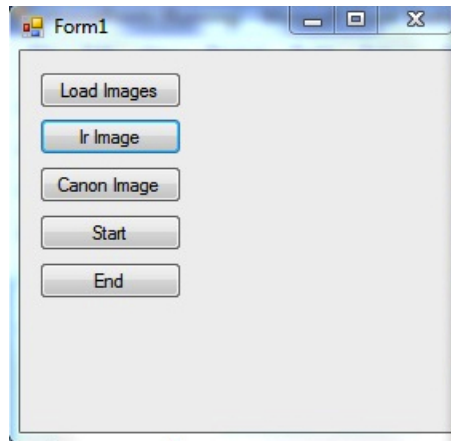
W celu znalezienia punktów charakterystycznych napisano specjalny program, w którym za pomocą myszki wskazywane są punkty charakterystyczne.

Program do znajdowania punktów charakterystycznych.

Program do znajdowania punktów charakterystycznych napisano w środowisku .NET, w języku C#. Napisany program jest programem okienkowym, umożliwia on

załadowanie i wyświetlenie dwóch obrazków. Punkty charakterystyczne wybierane są za pomocą myszki, klikając lewym przyciskiem w miejscu gdzie znajduje się szukany punkt. Na końcu lokalizacje punktów są zapisywane do plików tekstowych. Struktura plików zawierających punkty wygląda dokładnie tak jak struktura plików utworzonych przez program do szukania narożników szachownicy.

Główne okno programu pokazane jest na rysunku (rys. 3.14).

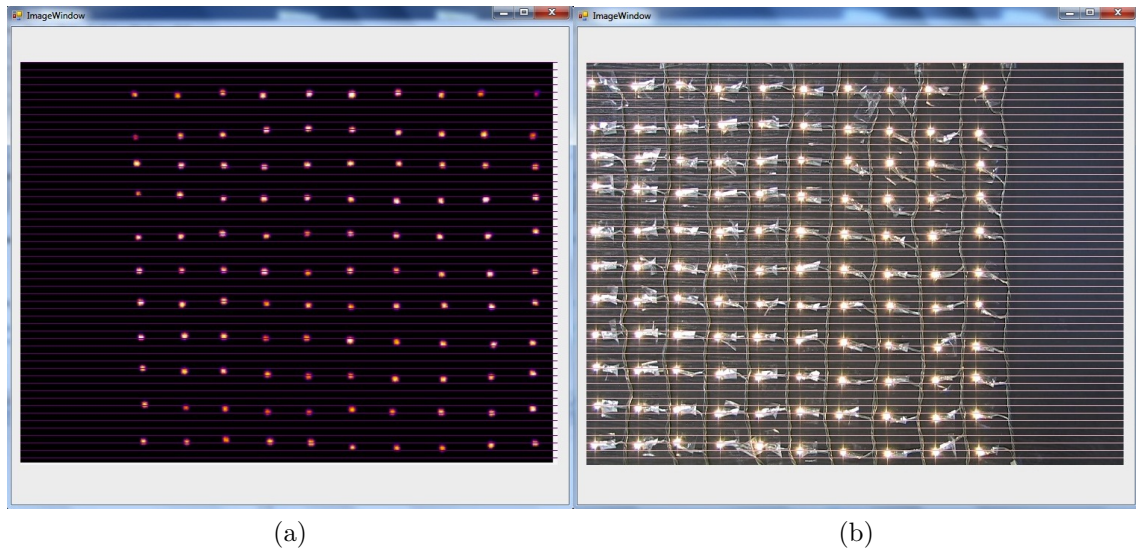


Rysunek 3.14: Główne okno programu do znajdowania punktów charakterystycznych.

Okno programu zawiera pięć przycisków dzięki którym możliwa jest jego obsługa:

- Po naciśnięciu przycisku "*Load Images*" obraz termowizyjny oraz obraz tradycyjny ładowane są do pamięci *RAM*. Obrazy które mają zostać załadowane muszą znajdować się w tym samym miejscu co plik *.exe* uruchamiający program. Obraz z kamery termowizyjnej musi nosić nazwę *i.bmb*, obraz z kamery tradycyjnej nazywa się *c.bmp*.
- Przycisk "*Ir Image*" pokazuje okno z załadowanym obrazem termowizyjnym (rys. 3.15a).
- Przycisk "*Canon Image*" pokazuje okno z załadowanym obrazem tradycyjnym (rys. 3.15b).
- Przycisk "*Start*" rozpoczyna procedurę szukania punktów. Po naciśnięciu tego przycisku wszystkie miejsca kliknięte myszką na obrazie termowizyjnym i tradycyjnym będą zapisywane do odpowiednich plików tekstowych.
- Przycisk "*End*" kończy procedurę wyszukiwania punktów charakterystycznych. Po jego naciśnięciu tworzone są dwa pliki zawierające pozycje punktów charakterystycznych. Pliki zapisywane są pod nazwami *ir_points.txt* oraz *canon_points.txt* odpowiednio dla lewej i prawej kamery.

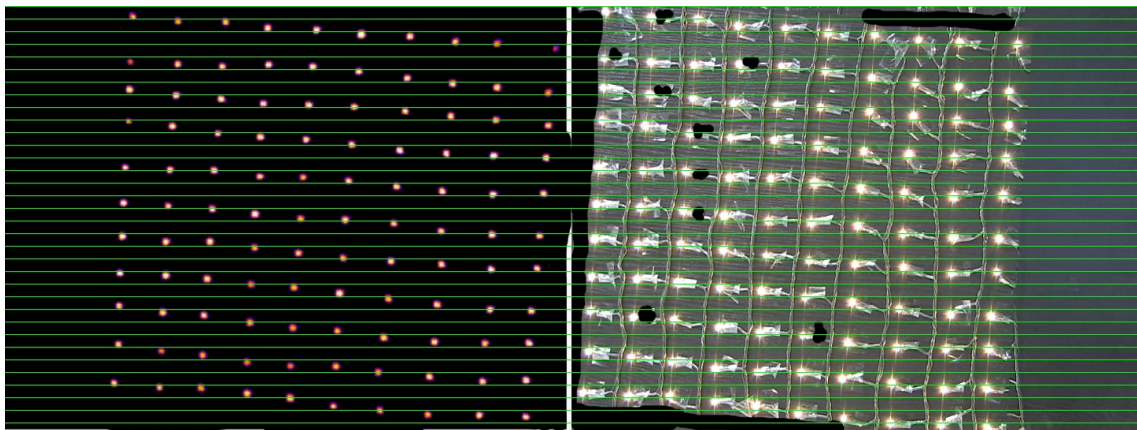
Wszystkie dalsze kroki rektyfikacyjne przeprowadzone zostały za pomocą programów opisanych wcześniej. Po znalezieniu punktów charakterystycznych utworzone zostały pliki konfiguracyjne dla dalszych kroków rektyfikacyjnych. Uruchomiony został program znajdujący macierz podstawową oraz program przekształcający obrazy do postaci obrazów zrektyfikowanych.



Rysunek 3.15: (a) okno z obrazem pochodzącym z kamery termowizyjnej, (b) okno z obrazem pochodzącym z kamery tradycyjnej.

Wyniki rektyfikacji

Algorytmem wykorzystanym w procesie szukania macierzy podstawowej był algorytm "8-points". Wyniki rektyfikacji pokazane są na rysunku (rys. 3.16).

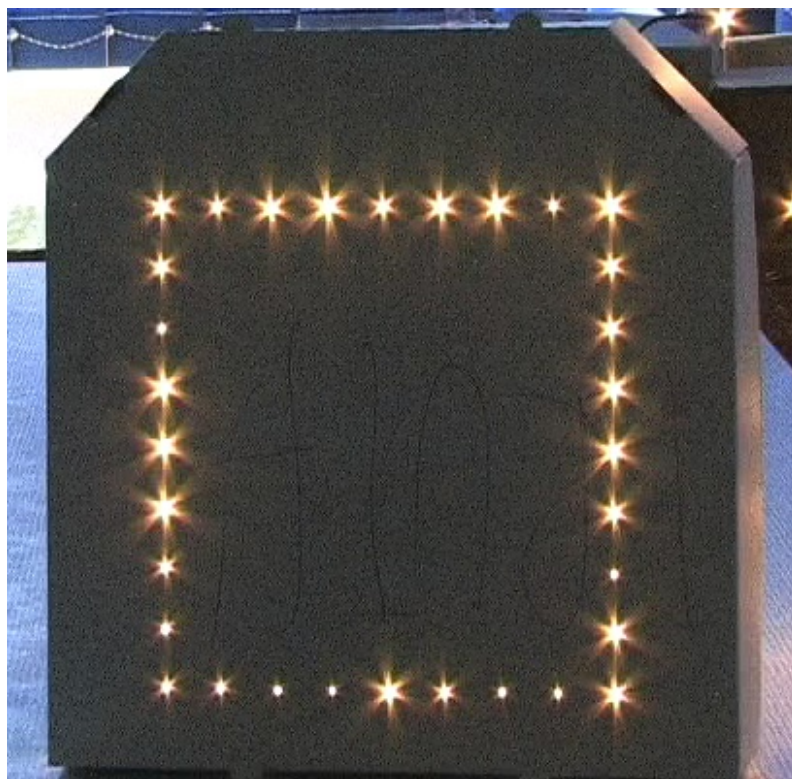


Rysunek 3.16: Wyniki rektyfikacji badanego systemu stereowizyjnego metodą Hartley'a.

3.6 Rektyfikacja i kalibracja badanego systemu.

Aby możliwe było dokonanie kalibracji kamer potrzebny jest obiekt o regularnych kształtach, gdzie odległości między charakterystycznymi punktami są dobrze znane. Wydrukowana na kartce papieru szachownica jest dobrym przykładem, nie sprawdza się jednak gdy w układzie stereowizyjnym jedna z kamer jest kamerą termowizyjną. Na potrzeby rektyfikacji algorytmem Bouguet'a, a więc i kalibracji potrzebne było stworzenie obiektu, na którym wszystkie charakterystyczne punkty są dobrze rozróżniane przez obie kamery.

Stworzono obiekt dzięki któremu możliwe było stworzenie szachownicy na gotowych już zdjęciach za pomocą technik przetwarzania obrazów. Obiektem kalibrującym było pudełko, w którym zrobiono dziurki w odpowiednich odległościach. W specjalnie przygotowane dziury włożono lampki choinkowe. Lampki tworzyły regularną siatkę, za pomocą której możliwe było stworzenie szachownicy na istniejących obrazach. Dzięki takiemu podejściu taka sama (regularna) struktura widziana była przez obie kamery, a punkty charakterystyczne mogły zostać wykorzystane w procesie kalibracji kamer. Obiekt kalibrujący pokazany został na rysunku (rys. 3.17).



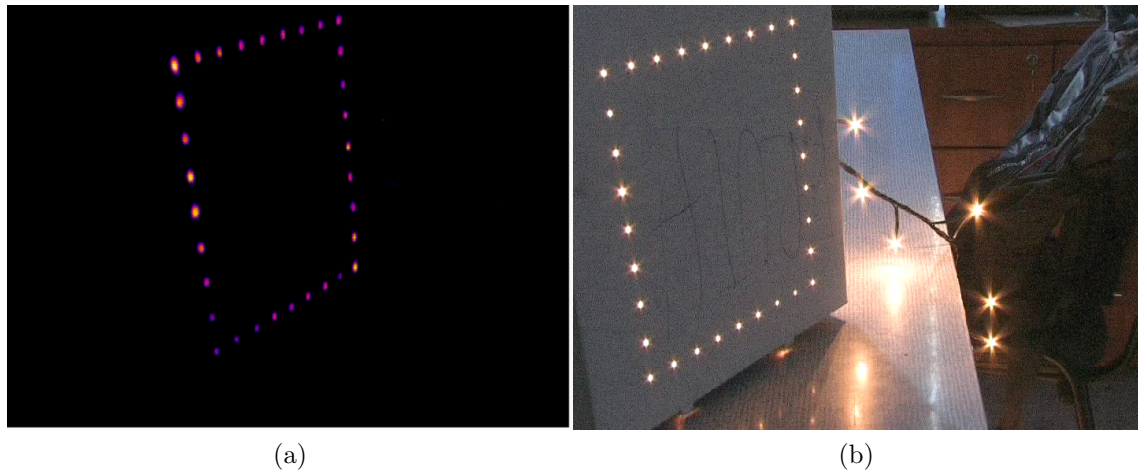
Rysunek 3.17: Obiekt służący do kalibracji i rektyfikacji algorytmem Bouguet'a.

Dzięki temu, że lampki choinkowe emitują światło i nagrzewają się jednocześnie możliwe było zarejestrowanie tych samych punktów dwiema równymi kamerami. Obrazy uzyskane kamerą termowizyjną i kamerą tradycyjną poddane zostały przekształceniom w celu wyznaczenia środków punktów charakterystycznych. Następnie odpowiednie punkty połączone zostały liniami dzięki czemu powstawały pola szachownicy. Ostatecznie powstałe pola wypełnione zostały na przemian białym oraz

czarnym kolorem. Takie postępowanie pozwalało uzyskać obrazy szachownicy, które zostały dalej wykorzystane w procesie rektyfikacji.

3.6.1 Tworzenie obrazu szachownicy

Obrazy zarejestrowane badanym systemem stereowizyjnym przedstawione zostały na rysunku (rys 3.18).



Rysunek 3.18: Obrazy obiektu kalibracyjnego zarejestrowane badanym systemem, (a) obraz termowizyjny, (b) obraz tradycyjny.

W celu wyznaczenia środków punktów charakterystycznych, rejestrowanych przez obie kamery stworzono program komputerowy. Powstała aplikacja ma za zadanie wyznaczyć środki odpowiednich punktów i w ich miejscu postawić krzyżyk. Na tak przygotowanych obrazach można bez problemu nanieść linie przy pomocy jakiegokolwiek programu graficznego. Linie łączące ze sobą odpowiednie punkty będą tworzyły siatkę szachownicy. Ostatecznie należy jeszcze wypełnić pola szachownicy na przemian białym i czarnym kolorem, również przy pomocy programu graficznego. Tak powstałe obrazy mogą być wykorzystane jako argumenty wejściowe do programu kalibrującego i rektyfikującego opisanego w sekcji (3.4).

Program wyznaczający środki punktów charakterystycznych

Opisywany program stworzono w języku *C++*, w środowisku *Visual Studio 2008* przy użyciu biblioteki *OpenCV*. Program ten do poprawnego działania potrzebuje pliku konfiguracyjnego, który zawiera następujące informacje:

- Ilość obrazków, na których mają zostać znalezione punkty charakterystyczne.
- Ścieżki do miejsca gdzie zapisane są obrazki.
- Szerokość początkowego obszaru, w którym przeszukiwany zostanie obrazek.
- Wysokość początkowego obszaru, w którym przeszukiwany zostanie obrazek.
- Informacja czy obrazek ma być pokazywany użytkownikowi w trakcie przekształceń.

- Informacja czy obrazy z zaznaczonymi środkami punktów charakterystycznych mają zostać zapisane na dysku.
- Ścieżka do miejsca, w którym zapisane zostaną obrazy wyjściowe.

W pierwszej fazie programu obrazek wejściowy poddawany jest działaniu algorytmów przetwarzania obrazów. W procedurze binarnego progowania[Jahne02] odfiltrowana zostaje ta część obrazu, która nie niesie ze sobą żadnej informacji o punktach charakterystycznych (rys. 3.19b). Kod odpowiedzialny za operacje binarnego progowania pokazany jest na listingu (listing 3.12).

Listing 3.12: Funkcja binarnego progowania z biblioteki *OpenCV*.

```
// Thermal camera
cvThreshold(dstImage, dstImage, 1, 255, CV_THRESH_BINARY );

// Canon camera
cvThreshold(dstImage, dstImage, 220, 255, CV_THRESH_BINARY );
```

Wywołanie funkcji *cvThreshold()* różni się dla obu kamer wybranym progiem poniżej którego wszystkie wartości z obrazków zostaną odfiltrowane. Parametr *CV_THRESH_BINARY* mówi o sposobie wykonania operacji progowania. Obrazy wyjściowe funkcji *cvThreshold()* mogą zostać wyświetlone na ekranie jeżeli w pliku konfiguracyjnym jest taka informacja.

Następną operacją wykonywaną na obrazach jest tak zwana operacja "distance transform"[Jahne02]. Dzięki temu możliwe jest wyznaczenie środków punktów charakterystycznych. Na środku punkty te są najjaśniejsze (rys 3.19c). Kod wykonujący przekształcenie pokazany jest na listingu (listing 3.13).

Listing 3.13: Przekształcenie "distance transform".

```
IplImage* distImage = cvCreateImage(cvSize(srcImage->width, srcImage->
    height), IPL_DEPTH_32F, 1);
cvZero(distImage);
cvDistTransform(dstImage, distImage, CV_DIST_L2, 3, NULL, NULL);
cvConvertScale(distImage, distImage, 10, 0);
```

Następnie przekształcony obraz jest wyświetlany jeżeli taka informacja znajduje się w pliku konfiguracyjnym.

Dalej obraz dzielony jest na mniejsze obszary, informacja o wymiarach tych obszarów znajduje się w pliku konfiguracyjnym. Wymiary obszaru poszukiwań zostają dostosowane do odległości między punktami charakterystycznymi (listing 3.14). Na początku wybierany jest najjaśniejszy punkt na całym obrazie i od niego zaczyna się szukanie punktów o wartości większej od "0" we wszystkich czterech kierunkach. Lokalizacja ostatnich niezerowych punktów jest zapamiętywana i na tej podstawie wyliczana jest ostateczna wielość obszaru poszukiwań.

Listing 3.14: Procedura dopasowania wielkości maski do przetwarzanego obrazu.

```
int imageWidth = srcImage->width;
int imageHeight = srcImage->height;

double biggestValue = 0;
int centerX, centerY;
for (int i = 0; i < imageHeight; i++)
    for (int j = 0; j < imageWidth; j++)
```

```

        {
            double currentValue = cvGet2D(srcImage, i, j).val[0];
            if (currentValue > biggestValue)
            {
                centerX = j;
                centerY = i;
                biggestValue = currentValue;
            }
        }

    int startX = centerX, endX = centerX, startY = centerY, endY = centerY
    ;
    while (cvGet2D(srcImage, centerY, startX).val[0] > 0)
        startX--;
    while (cvGet2D(srcImage, centerY, endX).val[0] > 0)
        endX++;
    while (cvGet2D(srcImage, startY, centerX).val[0] > 0)
        startY--;
    while (cvGet2D(srcImage, endY, centerX).val[0] > 0)
        endY++;

    maskWidth = endX - startX + 10;
    maskHeight = endY - startY + 10;

```

Następnie wszystkie punkty w każdym z obszarów są sprawdzane w celu znalezienia punktu o niezerowej wartości. Jeżeli w jakimś obszarze znajduje się dodatni punkt obrazu to oznacza to, że gdzieś w pobliżu znajduje się szukany środek punktu charakterystycznego. Środek znajduje się w miejscu, w którym znajduje się najjaśniejszy punkt w obszarze poszukiwań i jest on lokalizowany na tej podstawie (listing 3.15).

Listing 3.15: Procedura szukania punktu obrazu o najwyższej wartości w wyznaczonym obszarze.

```

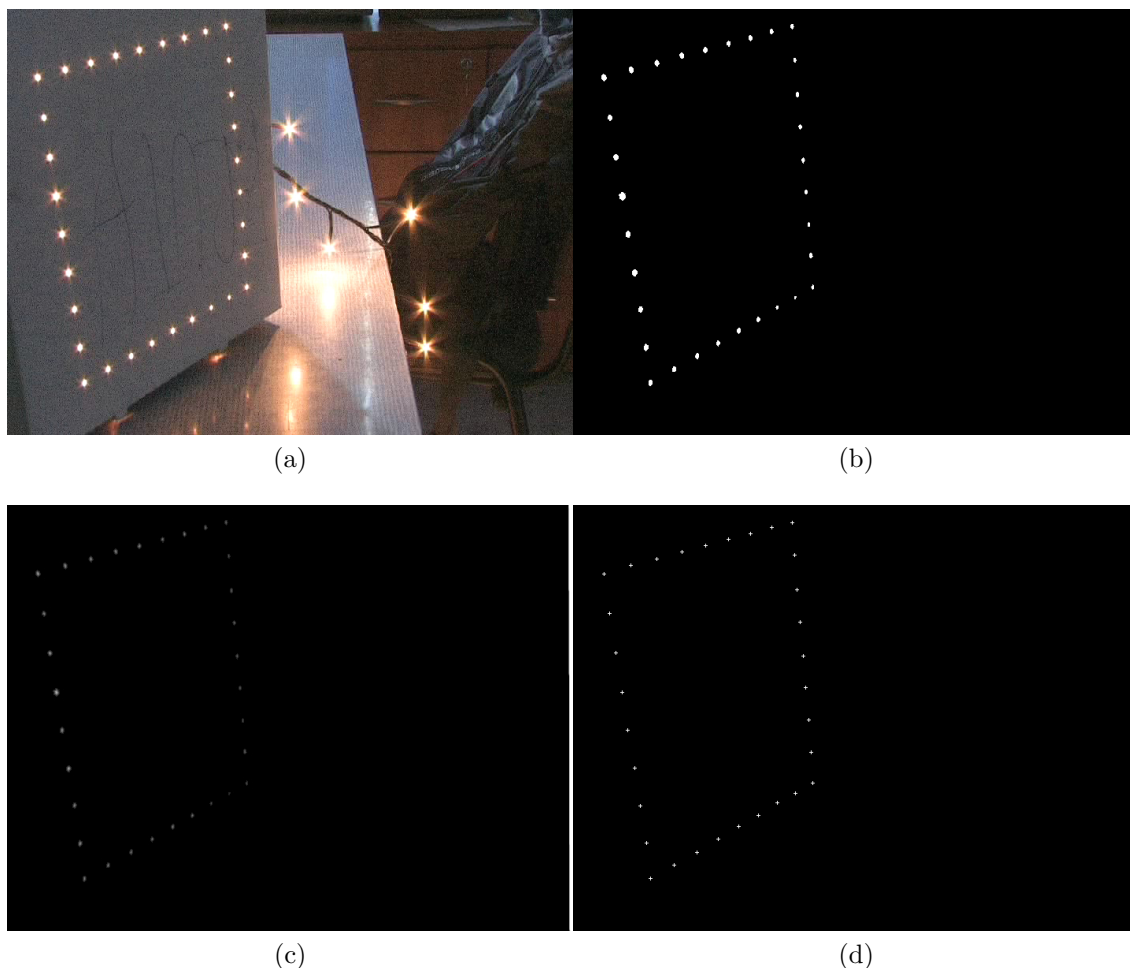
int imageWidth = srcImage->width;
int imageHeight = srcImage->height;

int startPointX = (startX - (maskX / 2)) >= 0 ? startX - (maskX / 2) :
0;
int endPointX = (startX + maskX) < imageWidth ? startX + maskX :
imageWidth;
int startPointY = (startY - (maskY / 2)) >= 0 ? startY - (maskY / 2) :
0;
int endPointY = (startY + maskY) < imageHeight ? startY + maskY :
imageHeight;

double biggestValue = 0;
for (int i = startPointY; i < endPointY; i++)
    for (int j = startPointX; j < endPointX; j++)
    {
        double currentValue = cvGet2D(srcImage, i, j).val[0];
        if (currentValue > biggestValue)
        {
            biggestPoint.y = i;
            biggestPoint.x = j;
            biggestValue = currentValue;
        }
    }
}

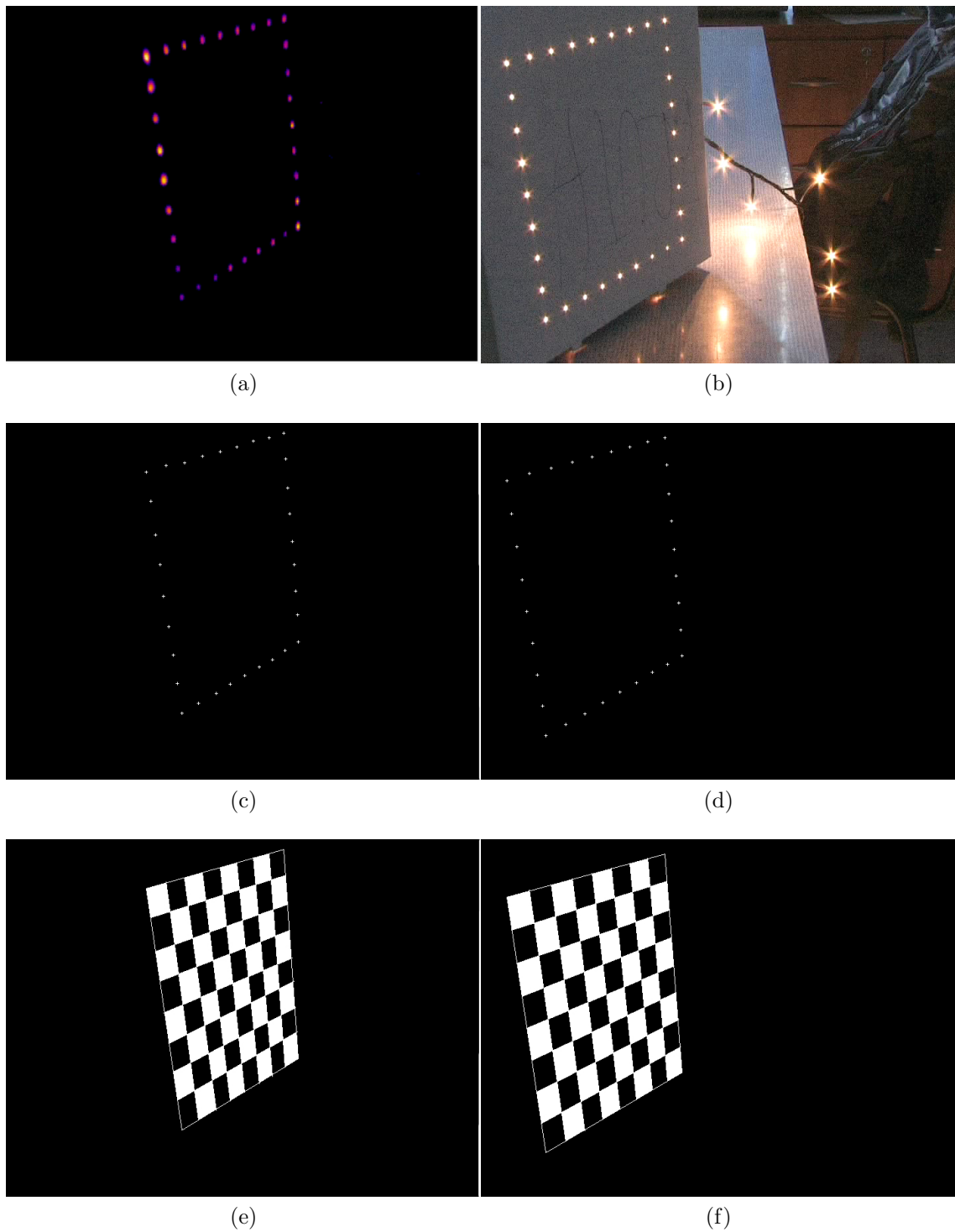
```

Współrzędne wyznaczonego środka zostają zapisane do specjalnie przygotowanego wektora, a obszar poszukiwań zostaje wyzerowany w celu uniknięcia sytuacji, w której dwa razy zostanie odszukany ten sam punkt. Ostatecznie na obrazach wyjściowych stawiane są krzyżyki w miejscu gdzie znajdują się środki punktów charakterystycznych (rys 3.19d).



Rysunek 3.19: Etapy znajdowania środków punktów charakterystycznych, (a) obraz wejściowy, (b) obraz po operacji binarnego progowania, (c) obraz po operacji "distance transform", (d) obraz wyjściowy.

Po wyznaczeniu środków punktów charakterystycznych możliwe jest połączenie tych punktów liniami oraz wypełnienie powstałych w ten sposób pól białym i czarnym kolorem za pomocą dowolnego programu graficznego. Obrazy po takiej obróbce mogą zostać bezpośrednio podane jako dane wejściowe do programu opisanego w sekcji (3.2). Powstałe obrazy widoczne są na rysunku (rys. 3.20).



Rysunek 3.20: Etapy powstawania obrazów szachownic; (a), (c), (e) obrazy z kamery termowizyjnej; (b), (d), (f) obrazy z kamery tradycyjnej.

3.6.2 Rektyfikacja metodą Bouguet'a

Obrazy uzyskane za pomocą technik opisanych w poprzednim podrozdziale zostały podane bezpośrednio do programu kalibrującego i rektyfikującego. Kalibracja przeprowadzona została na 59 obrazach szachownicy uzyskanych w sposób opisany wyżej. Rektyfikacja przeprowadzona została dla różnych ustawień parametrów funkcji kalibrującej z biblioteki *OpenCV*. Wyniki pokazane zostały na rysunku (rys. 3.21).

Kalibracja przeprowadzona została dla czterech ustawień funkcji kalibrującej:

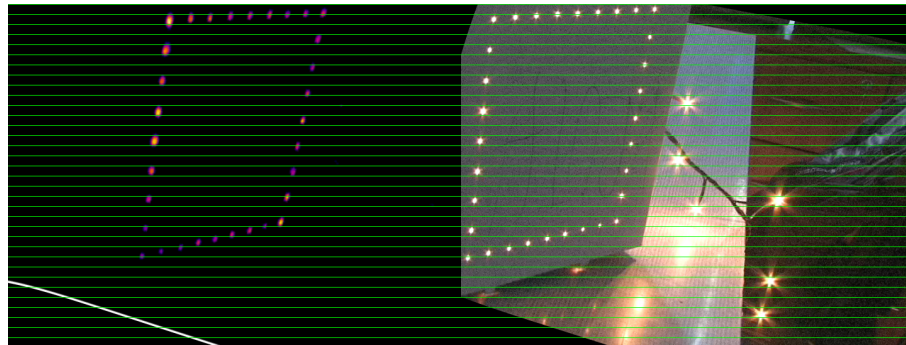
- Rysunek (rys. 3.21a) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji zniekształceń tangensowych, ogniskowych oraz punktów głównych.
- Rysunek (rys. 3.21b) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji tylko zniekształceń tangensowych.
- Rysunek (rys. 3.21c) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji punktów głównych i zniekształceń tangensowych.
- Rysunek (rys. 3.21d) pokazuje wyniki rektyfikacji w przypadku pełnej kalibracji - wszystkie parametry zostały wyliczone w procesie kalibracji.

3.6.3 Rektyfikacja metodą Hartley'a

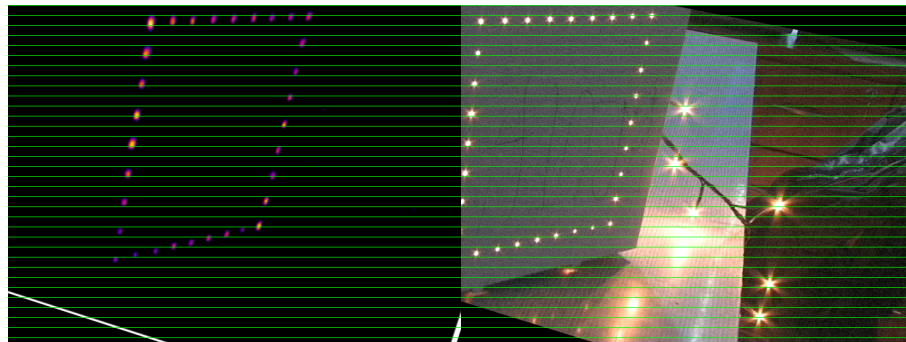
Możliwe jest również przeprowadzenie rektyfikacji metodą Hartley'a posługując się obrazami uzyskanymi w podrozdziale (3.6.1). Metoda Hartley'a nie wymaga skalibrowanego systemu stereowizyjnego, potrzebna jest tylko informacja na temat korespondujących ze sobą punktów na obu obrazach. Za pomocą algorytmu do znajdowania narożników szachownicy można wyznaczyć korespondujące punkty dla wszystkich par obrazów. W celu uzyskania wyników rektyfikacji stworzono listę wszystkich współrzędnych narożników szachownicy na lewych i prawych obrazach. Stworzona lista została podana jako parametr wejściowy do algorytmu Hartley'a. Rektyfikacja przeprowadzona została na 59 obrazach zawierających szachownicę, wyniki zebrane zostały na rysunku (rys. 2.22).

Rektyfikacja przeprowadzona została dla trzech ustawień funkcji szukającej macierzy podstawowej:

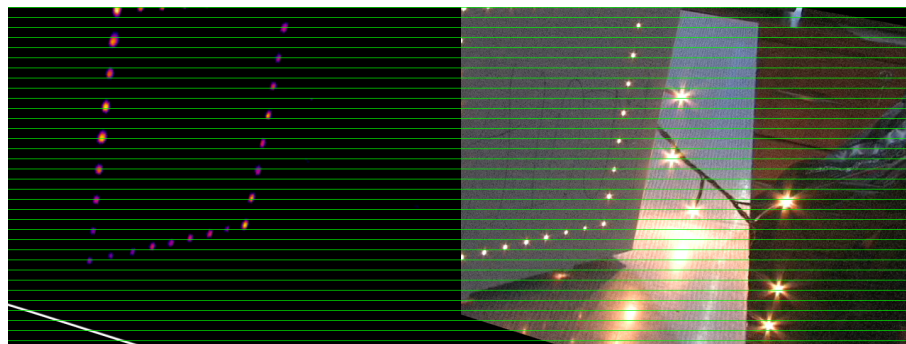
- Rysunek (rys. 3.22a) pokazuje wyniki rektyfikacji w przypadku szukania macierzy podstawowej algorytmem *RANSAC*.
- Rysunek (rys. 3.22a) pokazuje wyniki rektyfikacji w przypadku szukania macierzy podstawowej algorytmem *8-points*.
- Rysunek (rys. 3.22a) pokazuje wyniki rektyfikacji w przypadku szukania macierzy podstawowej algorytmem *LMedS*.



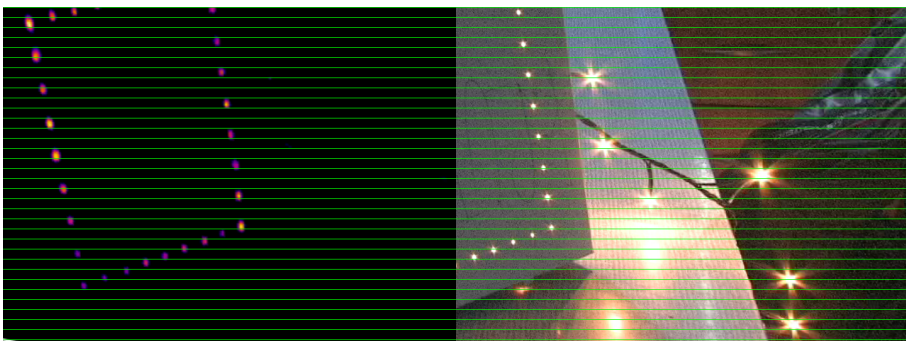
(a)



(b)

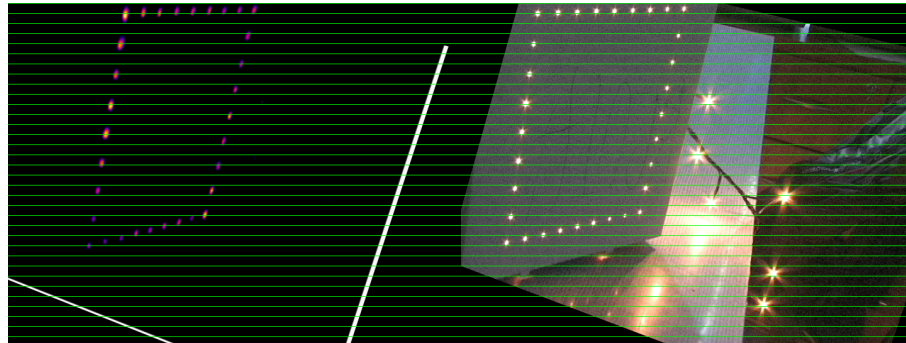


(c)

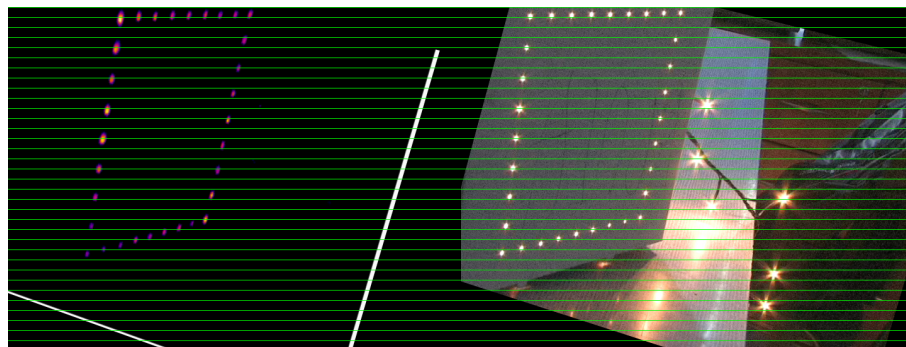


(d)

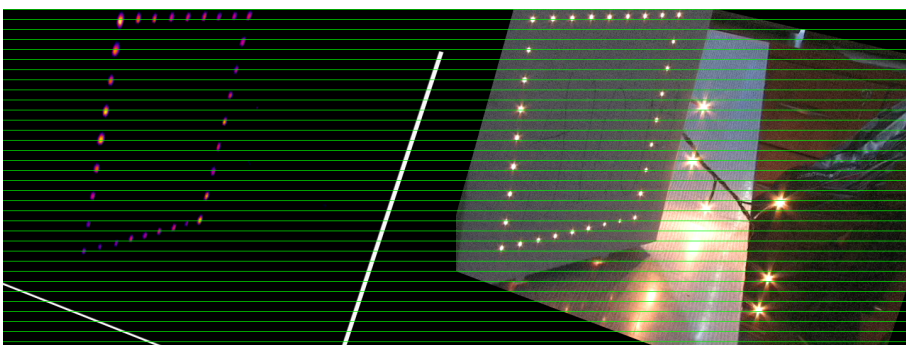
Rysunek 3.21: Wyniki rektyfikacji algorytmem Bouguet'a.



(a)



(b)



(c)

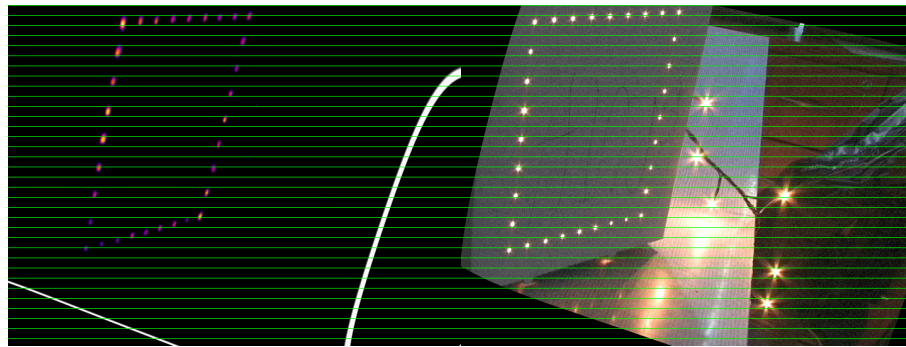
Rysunek 3.22: Wyniki rektyfikacji algorytmem Hartley'a.

3.6.4 Kalibracja i rektyfikacja metodą Hartley’a

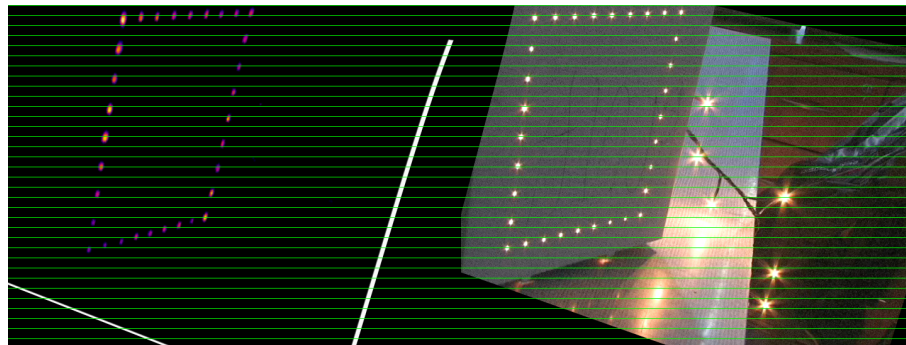
Tak jak dla systemu stereowizyjnego, w którym dwie kamery działają w zakresie światła widzialnego możliwa jest rektyfikacja metodą Hartley’a, gdzie macierz podstawowa znajdowana jest w procesie kalibracji. Rektyfikacja przeprowadzona została na 59 obrazach. Macierz podstawowa została znaleziona w procesie kalibracji. Wyniki pokazane zostały na rysunku (rys. 3.23).

Kalibracja przeprowadzona została dla czterech ustawień funkcji kalibrującej:

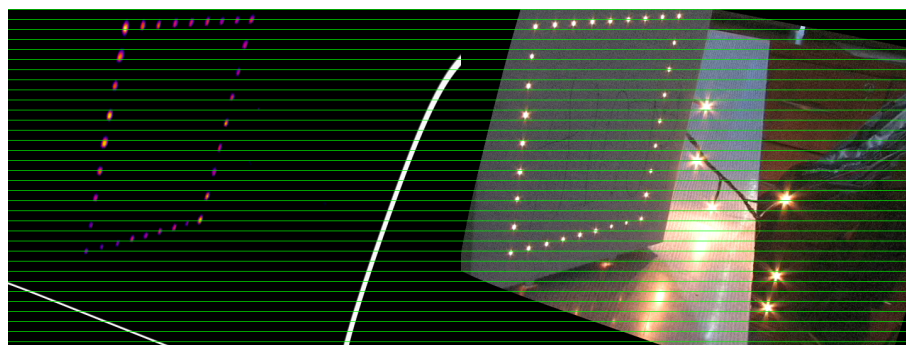
- Rysunek (rys. 3.23a) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji zniekształceń tangensowych, ogniskowych oraz punktów głównych.
- Rysunek (rys. 3.23b) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji tylko zniekształceń tangensowych.
- Rysunek (rys. 3.23c) pokazuje wyniki rektyfikacji w przypadku kalibracji z ominięciem estymacji punktów głównych i zniekształceń tangensowych.
- Rysunek (rys. 3.23d) pokazuje wyniki rektyfikacji w przypadku pełnej kalibracji - wszystkie parametry zostały wyliczone w procesie kalibracji.



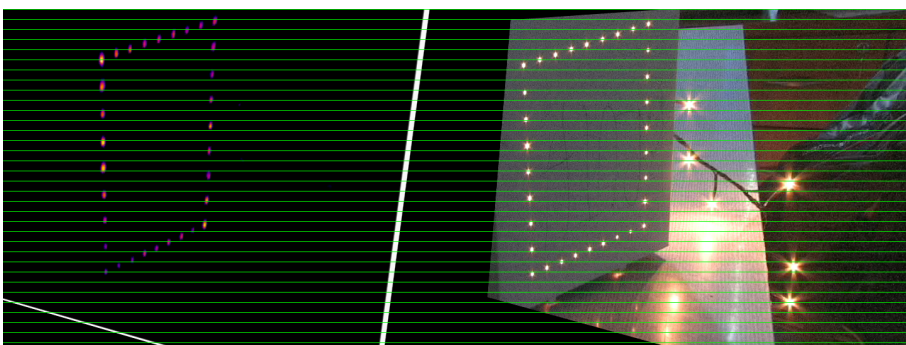
(a)



(b)



(c)



(d)

Rysunek 3.23: Wyniki rektyfikacji algorytmem Hartley'a, macierz podstawowa została znaleziona w procesie kalibracji.

Rozdział 4

Zakończenie

Rozdział czwarty podsumowuje wyniki rektyfikacji badanego systemu stereowizyjnego. Dzięki przeprowadzonym testom możliwe jest wybranie najlepszej metody rektyfikacji. Zamieszczone są również wnioski z przeprowadzonej analizy oraz pomysły dzięki którym rektyfikacja mogłaby być dokładniejsza.

4.1 Podsumowanie wyników

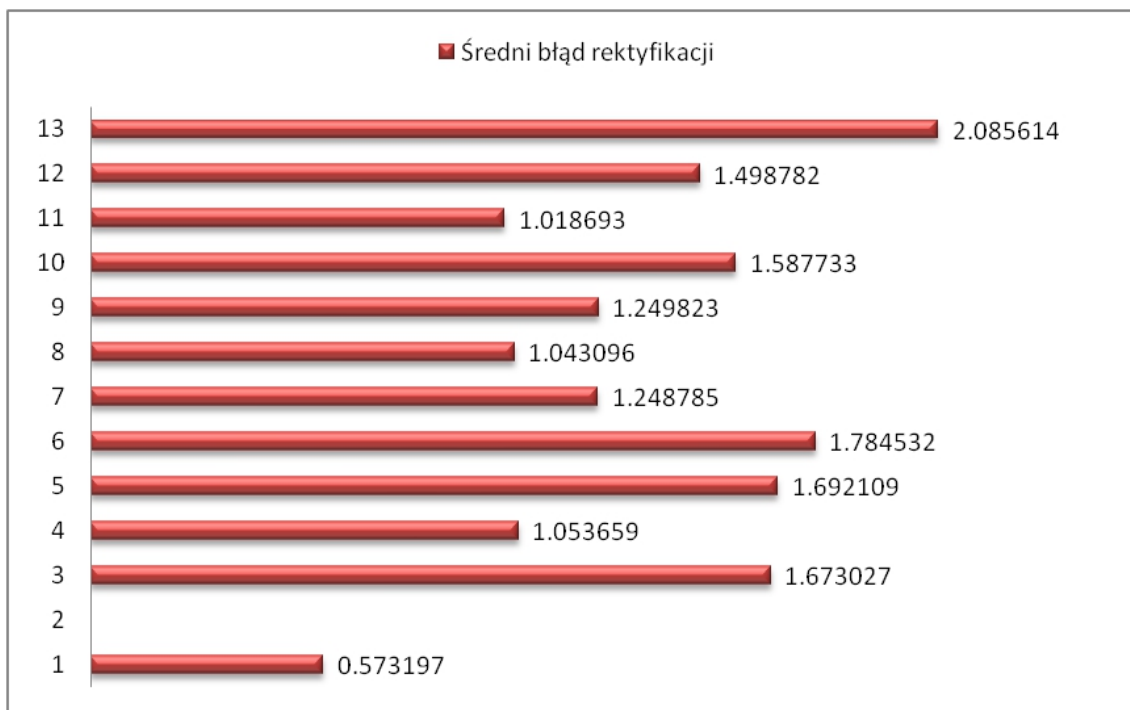
W celu podsumowania wyników oraz wyboru najlepszej metody rektyfikacji zrealizowano następujące etapy:

1. Przeprowadzono rektyfikację dla badanego systemu wszystkimi trzema proponowanymi metodami.
2. Zrektyfikowane obrazy wyjściowe poddane zostały jeszcze raz procedurze wyznaczenie punktów charakterystycznych.
3. Wyznaczone punkty zostały na siebie nałożone i wyliczone zostały różnice w położeniu punktów na obrazach zrektyfikowanych.
4. Wyliczony został średni błąd rektyfikacji jako średnia z różnic położenia punktów charakterystycznych na obrazach zrektyfikowanych.

Zbiorcze wyniki pokazujące dokładność rektyfikacji przedstawione zostały na wykresie (rys. 4.1). Wielkość błędu rektyfikacji wyrażona jest w punktach obrazu. Dla porównania dokładności rektyfikacji na wykresie zamieszczono również średni błąd rektyfikacji przeprowadzonej metodą Bouguet'a dla systemu stereowizyjnego złożonego z dwóch kamer tradycyjnych.

- 1 - Metoda Bouguet'a dla dwóch kamer tradycyjnych.
- 3 - Rektyfikacja metodą Bouguet'a, kalibracja z ominięciem estymacji zniekształceń tangensowych, ogniskowych oraz punktów głównych.
- 4 - Rektyfikacja metodą Bouguet'a, kalibracja z ominięciem estymacji tylko zniekształceń tangensowych.
- 5 - Rektyfikacja metodą Bouguet'a, kalibracja z ominięciem estymacji punktów głównych i zniekształceń tangensowych.

- 6 - Rektyfikacja metodą Bouguet'a, wszystkie parametry zostały wyliczone w procesie kalibracji.
- 7 - Rektyfikacja metodą Hartley'a, macierz podstawowa wyliczona algorytmem *RANSAC*.
- 8 - Rektyfikacja metodą Hartley'a, macierz podstawowa wyliczona algorytmem *8-points*.
- 9 - Rektyfikacja metodą Hartley'a, macierz podstawowa wyliczona algorytmem *LMedS*.
- Punkty 10, 11, 12 i 13 to błąd rektyfikacji wyliczonej metodą Hartley'a, gdzie macierz podstawowa została wyznaczona w procesie kalibracji. Parametry kalibracji zostały ustawione odpowiednio tak jak w punktach 3, 4, 5 oraz 6.



Rysunek 4.1: Średni błąd rektyfikacji dla wszystkich zastosowanych metod.

4.2 Wnioski z przeprowadzonej analizy

Z wykresu (rys. 4.1) wynika przede wszystkim, że rektyfikacja systemu stereowizyjnego, w którym jedna z kamer działa w zakresie podczerwieni jest możliwa. Średni błąd rektyfikacji dla każdej z metod zawiera się w granicach od 1 do 2 punktu obrazu. Rektyfikacja metodami 4, 8 oraz 11 jest najdokładniejsza, mimo to dokładność rektyfikacji systemu zbudowanego z dwóch kamer tradycyjnych jest prawie dwa razy wyższa. Jest to najprawdopodobniej spowodowane dokładnością przyrządu rektyfikacyjnego.

Przeprowadzona analiza pokazuje, że wszystkie trzy zastosowane algorytmy dają zbliżone wyniki i każdy z nich może zostać wykorzystany do uzyskania zrektyfikowanego systemu dwóch kamer. Zarówno metoda rektyfikacji zaproponowana przez Bouguet'a[Bouguet] jak i metoda Hartley'a[Hartley98] daje wyniki dobrej jakości. Nieznacznie lepiej wypada metoda, w której macierz podstawowa znajdowana jest w procesie kalibracji, a sama rektyfikacja przeprowadzana jest za pomocą algorytmu Hartley'a.

Dokładność rektyfikacji przeprowadzanej metodą Bouguet'a zależy w dużej mierze od parametrów podanych do funkcji kalibrującej. Najlepsze wyniki uzyskano gdy w procesie kalibracji pominięte zostały tylko zniekształcenia tangensowe, a wszystkie inne parametry wyliczane były zgodnie z algorytmem Zhanga[Zhang98].

Rektyfikacja przeprowadzona metodą Hartley'a jest nieco dokładniejsza niż w przypadku algorytmu Bouguet'a. Dodatkowo obrazy zrektyfikowane za pomocą tej metody (rys. 3.22) nie są tak bardzo zniekształcone jak obrazy uzyskane w przypadku rektyfikacji poprzedzonej kalibracją (rys. 3.21). Pole widzenia (ilość punktów z obrazów oryginalnych) również jest większe w przypadku rektyfikacji metodą Hartley'a. Sumarycznie metoda Hartley'a wypada lepiej w przypadku kalibracji badanego systemu. Najdokładniejsze wyniki rektyfikacji uzyskano gdy w poszukiwaniu macierzy podstawowej użyto algorytmu *8-points*. Algorytmy *RANSAC* oraz *LMedS* zwracają nieznacznie gorsze wyniki i bardzo zbliżone do siebie. Nie można więc jednoznacznie określić który z algorytmów lepiej sprawdza się w rektyfikacji badanego systemu.

Ostatnia badana metoda również korzysta z algorytmu kalibrującego. Macierz podstawowa wyliczana jest w procesie kalibracji, a sama rektyfikacja przeprowadzana jest za pomocą algorytmu Hartley'a. Najmniejszy błąd uzyskano gdy w procesie kalibracji pominięte zostały tylko zniekształcenia tangensowe. Najślabsze wyniki uzyskane zostały gdy kalibracja przeprowadzona została w całości nie pomijając żadnych parametrów wyliczanych w tym procesie. Zniekształcenia obrazów zrektyfikowanych są na poziomie zniekształceń uzyskanych algorytmem Hartley'a. Analizując wykres przedstawiający zbiorcze wyniki błędu rektyfikacji można zauważyć, że wyniki uzyskane pierwszą i trzecią metodą są do siebie podobne. Średni błąd rektyfikacji dla obu tych metod zależy w głównej mierze od parametrów podanych do funkcji kalibrującej.

Ze względu na złożoność obliczeń funkcji kalibrującej metoda Hartley'a działa znacznie szybciej niż dwie pozostałe metody. Średni błąd rektyfikacji oraz poziom zniekształcenia obrazów zrektyfikowanych również przemawiają na korzyść rektyfikacji metodą Hartley'a. Podsumowując wszystkie wyniki i dane na temat wszystkich zastosowanych algorytmów można stwierdzić, iż algorytm Hartley'a sprawdza się najlepiej dla systemu stereowizyjnego z kamerą nagrywającą w podczerwieni.

Do procesu rektyfikacji obrazów pochodzących z układu dwóch kamer działających w zakresie światła widzialnego zastosowano wydrukowaną szachownicę jako obiekt kalibrujący. Błędy w pomiarze odległości między punktami charakterystycznymi na obiekcie wynikały tylko i wyłącznie ze zniekształceń obiektów. Dodatkowo obie kamery były takie same, a więc współczynniki zniekształceń były podobne. Ze względu na zastosowanie kamery termowizyjnej w badanym układzie niemożliwe było wykorzystanie obrazu szachownicy jako obiektu kalibrującego. Stworzono więc specjalny obiekt, na którym punkty charakterystyczne zostały naniesione odręcznie. Metoda taka jest obciążona większym błędem. Dodatkowo aby stworzyć punkty charakterystyczne biorące udział w kalibracji oraz rektyfikacji należało posłużyć się

technikami przetwarzania obrazów, które również wprowadziły błąd w położeniu punktów charakterystycznych. Ze względu na różnice we właściwościach obrazów pochodzących z obu kamer zastosowane zostały techniki wstępnej kalibracji, które również mogą wpływać na dokładność rozmieszczenia punktów charakterystycznych.

Pracując na obrazach pochodzących z systemu stereowizyjnego, w którym dwie kamery różnią się w znaczny sposób właściwościami, nie da się uniknąć konieczności przeprowadzenia wstępnej kalibracji. Sama dokładność rektyfikacji może natomiast zostać zwiększona poprzez zwiększenie dokładności wykonania obiektu rektyfikującego. W przypadku badanego systemu możliwe jest przygotowanie obiektu, który dokładnością nie będzie ustępował wydrukowanemu obrazowi szachownicy. W tym celu można na przykład zaprojektować płytkę drukowaną, na której diody LED będą stanowiły punkty charakterystyczne. W procesie projektowania takiej płytki należałoby zadbać o to aby diody rozmieszczone były w sposób regularny, a odległość między każdą parą diod była taka sama. Dzięki takiemu podejściu można uniknąć konieczności stosowania technik przetwarzania obrazu do stworzenia punktów charakterystycznych. Istnieje bardzo duże prawdopodobieństwo, że zastosowanie dokładniejszego obiektu rektyfikującego poprawi jakość całej rektyfikacji.

Dodatek A

Zawartość płyty CD

W skład pracy wchodzi również płyta CD zawierająca wszystkie kody źródłowe programów przygotowanych w ramach zbadania systemu stereowizyjnego. Dodatkowo na płycie znajdują się biblioteki dynamiczne w postaci plików *.dll* potrzebne do prawidłowego działania zawartych tam programów. W skład płyty wchodzi również pliki ze zdjęciami pochodzące z badanego systemu dwukamerowego, przykładowe pliki konfiguracyjne potrzebne do poprawnego działania stworzonych programów oraz pliki *.bat* zawierające przykładowe użycie wszystkich opisanych aplikacji. Struktura płyty jest następująca:

- Folder "*code*" - zawiera kody źródłowe wszystkich programów wykorzystanych w pracy. W jego skład wchodzi foldery:
 - CameraCalibrate - zawiera program kalibrujący dla jednej kamery.
 - Dump-filter - kod źródłowy filtru *DirectShow* opisanego w sekcji (3.2.3).
 - FindChessboardCorners - kod źródłowy programu znajdującego narożniki szachownicy.
 - PreCalibration - program do wstępnej kalibracji kamer.
 - StereoCalibrate - kod źródłowy programu kalibrującego i rektyfikującego zestaw stereowizyjny.
 - UndistortAfterCalibration - program pozbawiający obrazy zniekształceń po procesie kalibracji.
 - UndistortRectify - kod źródłowy tworzący zrektyfikowane obrazy na podstawie przekształcenia rektyfikacyjnego.
 - ChoosePoints - program umożliwiający ręczny wybór punktów charakterystycznych, opisany w sekcji (3.5.1).
 - CreateImagesForChess - program znajdujący środki punktów charakterystycznych, opisany w sekcji (3.6.1)
- Folder "*dll*" - zawiera biblioteki potrzebne do poprawnego działania opisanych programów.
- Folder "*single_camera_calibration*" - zawiera programy, przykładowe pliki konfiguracyjne i zdjęcia potrzebne do przeprowadzenia kalibracji pojedynczej kamery.

- Folder ” *stereo_calibration_rectification*” - zawiera programy, przykładowe pliki konfiguracyjne oraz zdjęcia potrzebne do przeprowadzenia rektyfikacji obrazów z systemu stereowizyjnego. Dane zawarte w folderze umożliwiają rektyfikację zarówno systemu tradycyjnego jak i systemu w którym jedna z kamer jest kamera termowizyjną.

Dodatek B

Opis plików konfiguracyjnych wchodzących w skład pracy

Dodatek zawiera opisy wszystkich plików konfiguracyjnych, wraz z przykładami, potrzebnych do poprawnego działania wszystkich użytych programów.

B.1 Plik programu do znajdowania narożników szachownicy

Przykładowa postać pliku pokazana jest na listingu (listing B.1).

Listing B.1: Plik programu znajdującego narożniki szachownicy.

Inputs :

Number of Images :	1
Path to image :	image.JPG
Number of rows :	7
Number of cols :	7
Find corners subpixel :	1
Show Results :	0
Save found points :	1

Outputs :

Path to points on image :	points.txt
---------------------------	------------

- *Number of Images* - ilość zdjęć na jakich mają zostać znalezione narożniki.
- *Path to image* - ścieżka do obrazu zawierającego szachownicę.
- *Number of rows* - ilość narożników w jednym wierszu szachownicy.
- *Number of cols* - ilość narożników w jednej kolumnie szachownicy.
- *Find corners subpixel* - jeżeli "1" to narożniki będą szukane z dokładnością do jednego subpiksela.
- *Show Results* - jeżeli "1" to wyniki będą wyświetlane na ekranie w trakcie działania programu.

- *Save found points* - jeżeli "1" to informacje o lokalizacji narożników zostaną zapisane w plikach tekstowych.
- *Path to points on image* - ścieżka do miejsca gdzie mają zostać zapisane znalezione narożniki.

B.2 Plik programu do kalibracji pojedynczej kamery

Przykładowa postać pliku pokazana jest na listingu (listing B.2).

Listing B.2: Plik programu do kalibracji pojedynczej kamery.

Inputs :

Number of Images :	1
Path to points :	points0.txt
Number of rows :	7
Number of cols :	7
Image width :	640
Image height :	480
Points subpixel :	1
Save outputs :	1

Outputs :

Path to camera matrix :	cameraMatrix.xml
Path to distortion coeffs :	distCoeffs.xml

- *Number of Images, Number of rows, Number of cols, Find corners subpixel* oraz *Save outputs* - te same informacje co w dodatku (B.1).
- *Path to points* - ścieżka do pliku tekstowego zawierającego lokalizacje narożników szachownicy.
- *Image width* - szerokość obrazu w punktach.
- *Image height* - wysokość obrazu w punktach.
- *Path to camera matrix* - ścieżka do miejsca, w którym ma zostać zapisana macierz wewnętrzna kamery.
- *Path to distortion coeffs* - ścieżka do miejsca, w którym ma zostać zapisany wektor zniekształceń.

B.3 Plik programu tworzącego obrazy pozbawione zniekształceń

Przykładowa postać pliku pokazana jest na listingu (listing B.3).

Listing B.3: Plik programu tworzącego obrazy pozbawione zniekształceń.

Inputs :

Number of Images:	1
Path to images:	image.JPG
Path to camera matrix:	cameraMatrix.xml
Path to distortion coeffs:	distCoeffs.xml
Show result:	0

Outputs:

Path to output images:	ud.JPG
------------------------	--------

- *Number of Images*, *Path to image* oraz *Show Results* - te same informacje co w dodatku (B.1).
- *Path to camera matrix* - ścieżka do miejsce, w którym zapisana jest macierz podstawowa.
- *Path to distortion coeffs* - ścieżka do miejsca, w którym zapisany jest wektor zniekształceń.
- *Path to output images* - ścieżka do miejsca w którym mają zostać zapisane przekształcone obrazy.

B.4 Plik programu kalibrującego i rektyfikującego

Przykładowa postać pliku pokazana jest na listingu (listing B.4).

Listing B.4: Plik programu kalibrującego i rektyfikującego.

Inputs :

Number of Images:	1
Path to left points:	points0.txt
Path to right points:	points0.txt
Number of rows:	7
Number of cols:	7
Image width:	640
Image height:	480
Points subpixel:	1
Do calibration:	1
Do rectification:	1
Rectification algorithm:	1
Save calibration output:	1
Save rectification output:	1

Calibration Outputs:

Path to left camera matrix:	cameraMatrixLeft.xml
Path to right camera matrix:	cameraMatrixRight.xml
Path to left distortion coeffs:	distCoeffsLeft.xml
Path to right distortion coeffs:	distCoeffsRight.xml
Path to rotation matrix:	rotationMat.xml
Path to translation vector:	translationVec.xml
Path to essential matrix:	essentialMat.xml
Path to fundamental matrix:	fundamentalMat.xml

Rectification Outputs:

Path to left camera matrix rect:	cameraMatrixLeftRect.xml
Path to right camera matrix rect:	cameraMatrixRightRect.xml
Path to left distortion coeffs rect:	distCoeffsLeftRect.xml
Path to right distortion coeffs rect:	distCoeffsRightRect.xml
Path to left rotation matrix rect:	rotationMatLeftRect.xml
Path to right rotation matrix rect:	rotationMatRightRect.xml
Path to left projection matrix rect:	projectionMatLeftRect.xml
Path to right projection matrix rect:	projectionMatRightRect.xml
Path to reprojection matrix rect:	reprojectionMat.xml

-
- *Number of Images, Number of rows, Number of cols, Points subpixel, Image width* oraz *Image height* - te same informacje co w dodatku (B.2).
 - *Path to left points* i *Path to right points* to odpowiednio ścieżki do plików tekstowych zawierających lokalizację narożników na lewym i prawym obrazie.
 - *Do calibration* - jeżeli "1" to przeprowadzona zostanie kalibracja.
 - *Do rectification* - jeżeli "1" to przeprowadzona zostanie rektyfikacja.
 - *Rectification algorithm* - algorytm rektyfikacji, "1": algorytm Bouguet'a; "2": algorytm Hartley'a; "3": algorytm Hartley'a, macierz podstawowa liczona w procesie kalibracji.
 - *Save calibration output* - jeżeli "1" to wyniki kalibracji zostaną zapisane.
 - *Save rectification output* - jeżeli "1" to wyniki rektyfikacji zostaną zapisane.
 - *Path to left camera matrix* i *Path to right camera matrix* - ścieżki do plików, w których zapisane zostaną macierze wewnętrzne, odpowiednio dla lewej i prawej kamery.
 - *Path to left distortion coeffs* i *Path to right distortion coeffs* - ścieżki do plików, w których zapisane zostaną wektory zniekształceń, odpowiednio dla lewej i prawej kamery.
 - *Path to rotation matrix* i *Path to translation vector* - ścieżki gdzie zostaną zapisane odpowiednio macierz rotacji i wektor przesunięcia.
 - *Path to essential matrix* i *Path to fundamental matrix* - ścieżki gdzie zostaną zapisane odpowiednio macierz zasadnicza i podstawowa.

- *Path to left camera matrix rect* i *Path to right camera matrix rect* - ścieżki gdzie zostaną zapisane zrektyfikowane macierze wewnętrzne, odpowiednio dla lewej i prawej kamery.
- *Path to left distortion coeffs rect* i *Path to right distortion coeffs rect* - ścieżki gdzie zostaną zapisane zrektyfikowane wektory zniekształceń, odpowiednio dla lewej i prawej kamery.
- *Path to left projection matrix rect* i *Path to right projection matrix rect* - ścieżki do plików, w których zapisane zostaną macierze projekcji, odpowiednio dla lewej i prawej kamery.
- *Path to reprojection matrix rect* - ścieżka do pliku w którym zapisana zostanie macierz reprojekcji.

B.5 Plik programu dokonującego przekształcenia rektyfikacyjnego

Przykładowa postać pliku pokazana jest na listingu (listing B.5).

Listing B.5: Plik programu dokonującego przekształcenia rektyfikacyjnego.

Inputs :

Number of Images :	1
Left images :	image.JPG
Right Images :	image.JPG
Path to left camera matrix rect :	cameraMatrixLeftRect.xml
Path to right camera matrix rect :	cameraMatrixRightRect.xml
Path to left distortion coeffs rect :	distCoeffsLeftRect.xml
Path to right distortion coeffs rect :	distCoeffsRightRect.xml
Path to left rotation matrix rect :	rotationMatLeftRect.xml
Path to right rotation matrix rect :	rotationMatRightRect.xml
Path to left projection matrix rect :	projectionMatLeftRect.xml
Path to right projection matrix rect :	projectionMatRightRect.xml
Rectification Algorithm :	1
Save images with lines :	0
Show results :	0
Save rectified images :	1

Outputs :

Path to left rectified images :	lr.jpg
Path to right rectified images :	rr.jpg

- *Number of Images* oraz *Rectification Algorithm* - te same informacje co w dodatku (B.4).
- *Left images* i *Right images* - ścieżki do obrazów, które zostaną poddane przekształceniu rektyfikacyjnemu.
- Wszystkie ścieżki do macierzy zwracane przez program rektyfikujący.
- *Save images with lines* - jeżeli "1" to obrazy z liniami zostaną zapisane.

- *Show results* - jeżeli "1" to obrazy zrektyfikowane zostaną wyświetlone na ekranie podczas działania programu.
- *Save rectified images* - jeżeli "1" to zrektyfikowane obrazy zostaną zapisane na dysku twardym.
- *Path to left rectified images* i *Path to right rectified images* - ścieżki do miejsc w których zapisane zostaną zrektyfikowane obrazy, odpowiednio z lewej i prawej kamery.

B.6 Plik program znajduącego środki punktów charakterystycznych

Przykładowa postać pliku pokazana jest na listingu (listing B.6).

Listing B.6: Plik program znajduącego środki punktów charakterystycznych.

Inputs :

Number of Images :	1
Images path :	image . bmp
Width of mask :	15
Height of mask :	15
Show middle states :	0
Save results :	1

Outputs :

Path to dst images :	dstImage . bmp
----------------------	----------------

- *Number of Images* - ilość obrazów na których znajdują się punkty charakterystyczne.
- *Images path* - ścieżka do obrazów zawierających punkty.
- *Width of mask* - początkowa szerokość maski w której odbywa się poszukiwanie punktów charakterystycznych.
- *Height of mask* - początkowa wysokość maski w której odbywa się poszukiwanie punktów charakterystycznych.
- *Show middle states* - jeżeli "1" to obrazy przejściowe są pokazywane podczas działania programu.
- *Save results* - jeżeli "1" to obrazy ze zlokalizowanymi punktami charakterystycznymi są zapisywane na dysku twardym.
- *Path to dst images* - ścieżka, w której zostaną zapisane obrazy ze zlokalizowanymi punktami charakterystycznymi.

Literatura

- [AForge] Andrew Kirillov *AForge.NET Framework*
<http://www.aforgenet.com/framework>
- [AForge01] Andrew Kirillov *AForge.NET Framework Documentation*
<http://www.aforgenet.com/framework/docs/>
- [Bradski08] Gary Bradski and Adrian Kaehler, *Learning OpenCV*. O'Reilly Media, Inc., Sebastopol, 2008.
- [Brown71] Duane C. Brown, *Close-range camera calibration*. Photogrammetric Engineering 37, 1971
- [Bouguet] Jean-Yves Bouguet, *Camera calibration toolbox for Matlab*
http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.
- [Flir06] FLIR Systems *ThermaCAM Researcher - User's manual*. 2006
- [Hartley98] Richard I. Hartley, *Theory and practice of projective rectification*. "International Journal of Computer Vision" 35, 1998, 115–127
- [Hartley03] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2nd Edition, 2003.
- [Jahne02] Bernd Jahne *Digital Image Processing*. Springer-Verlag Berlin Heidelberg, Heidelberg, 5th revised and extended edition, 2002.
- [Pesce03] Mark D. Pesce, *Programming Microsoft DirectShow for Digital Video and Television*. Microsoft Press A Division of Microsoft Corporation, Redmond, 2003.
- [Wiki01] wikipedia.org, *Homogeneous coordinates*.
http://en.wikipedia.org/wiki/Homogeneous_coordinates.
- [Wiki02] wikipedia.org, *Levenberg–Marquardt algorithm*.
http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm.
- [Wiki03] wikipedia.org, *Podczerwień*.
<http://pl.wikipedia.org/wiki/Podczerwien>.
- [Wiki04] wikipedia.org, *Thermographic camera*.
http://en.wikipedia.org/wiki/Thermographic_camera.
- [Zhang98] Zhengyou Zhang, *A Flexible New Technique for Camera Calibration*. Microsoft Research, Redmond, Technical Report MSR-TR-98-71, 1998.

- [Zhang99] Zhengyou Zhang, *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*. Proceedings of the 7th International Conference on Computer Vision (pp. 666–673), Corfu, 1999.