

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY
INSTYTUT STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ

PRACA MAGISTERSKA
na kierunku AUTOMATYKA I ROBOTYKA



Piotr Opaliński
Nr albumu 197542

Rok. Akad. 2008/2009
Warszawa, 15.11.2008 r.

Usuwanie przeplotu z sekwencji obrazów

Zakres pracy:

1. *Wstęp*
2. *Przegląd istniejących algorytmów służących do usuwania przeplotu z sekwencji obrazów*
3. *Omówienie Implementacji wybranych algorytmów służących do usuwania przeplotu*
4. *Porównanie zaimplementowanych algorytmów pod względem jakości przetworzonego obrazu oraz czasu działania algorytmów*
5. *Wnioski i podsumowanie*

Podpis i pieczętka

Kierownika Zakładu Dydaktycznego

Opiekun naukowy:
Dr inż. Witold Czajewski

Termin wykonania:
Praca wykonana i zaliczona pozostaje
własnością Instytutu i nie będzie
zwrócona wykonawcy

Usuwanie przeplotu w sekwencjach obrazów

Streszczenie

Celem pracy jest porównanie poprzez analizę oraz przedstawienie wyników badań istniejących metod usuwania przeplotu w oparciu o kryterium wydajności oraz skuteczności oraz dowiedzenie, że metoda zaproponowana przez autora opracowania w określonych warunkach jest najodpowiedniejsza.

Rozdział pierwszy charakteryzuje cel pracy, a także pokazuje rys historyczny powstania techniki przeplotu oraz opowiada o jego szerokim zastosowaniu. W rozdziale jest również poruszona kwestia usuwania przeplotu, kontynuowana w rozdziale następnym.

Rozdział drugi opisuje i ilustruje istniejące algorytmy usuwania przeplotu oraz zgrubnie charakteryzuje ich zalety oraz wady.

W rozdziale trzecim zawarty jest sposób podejścia do implementacji algorytmów usuwania przeplotu, zastosowane przez autora opracowania optymalizacje oraz uproszczenia. W tym rozdziale omówiona jest dokładnie metoda proponowana przez autora, jej geneza, założenia oraz implementacja.

W czwartym rozdziale autor przedstawia szczegółowe wyniki badań, w podziale na trzy kategorie. Pierwsza kategoria, zakłada porównanie wydajnościowe algorytmów usuwania przeplotu. W drugiej kategorii, te same metody poddane są badaniu jakościowemu, zarówno subiektywnemu, okiem badającego, jak i substatywnemu – w postaci porównania właściwości jakościowych algorytmów na podstawie skuteczności odnajdywania markerów w obrazie. Ostatnia, trzecia kategoria, weryfikuje przydatność metod usuwania przeplotu ze względu na dokładność umiejscowienia markera w przestrzeni euklidesowej.

Piąty rozdział jest podsumowaniem całej pracy, bazując na wynikach przeprowadzonych badań. Wskazuje on także potencjalne kierunki badawcze, które otwiera niniejsze opracowanie. Na koniec, odpowiada także na pytanie, dlaczego usuwanie przeplotu jest zagadnieniem nadal aktualnym i jakie perspektywy otwiera przyszłość.

Image sequence deinterlacing

Abstract

The goal of this thesis is a comparison drawing upon analysis and reporting results of the research on existing deinterlacing techniques based on effectiveness and efficiency criteria and proving that the method proposed by the case study by the author is most appropriate within certain conditions.

Chapter one characterises the goal of this thesis and shows the historical background on foundation of the interlacing technique and it's broad use. This chapter also brings up the matter of deinterlacing which is continued in chapter two.

Chapter two describes and illustrates existing deinterlacing algorithms and roughly characterises their pros and cons.

The third chapter contains methods of the implementation approach for deinterlacing algorithms and also describes optimisations and simplifications used by the elaboration's author during their coding. This chapter precisely describes the method proposed by the author, it's source, foundations and implementation.

In the fourth chapter, the author portrays the detailed research results, split into three categories. The first category assumes an efficiency comparison of deinterlacing algorithms. In the second category, the same methods are subject to a quality analysis, both subjective, by the eye of the author, and substantive, as a quantitative algorithm comparison of the quality characteristics based on the fiduciary marker detection effectiveness in the image sequence. The last, third category, verifies the usability of deinterlacing methods based on the precision of marker placement determination in the Euclidean space.

The fifth chapter wraps up the whole thesis basing on the results of the carried out research. It also indicates possible research directions that have been opened by this elaboration. Finally, it answers the question why deinterlacing is still a valid problem and what is the outlook on the future for deinterlacing.

Warszawa, dnia roku.

Politechnika Warszawska
Wydział Elektryczny

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa magisterska pt. „USUWANIE PRZEPLOTU Z SEKWENCJI OBRAZÓW”

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

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej.

Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

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

Imię i Nazwisko dyplomanta: Piotr Opaliński

Podpis dyplomanta:

Spis treści

Wstęp	1
I.1 Wprowadzenie	1
I.2 Definicja przeplotu	2
I.3 Zastosowanie i geneza przeplotu	4
I.4 Przeplot współcześnie	5
I.5 Usuwanie przeplotu	6
Przegląd istniejących algorytmów usuwania przeplotu	7
II.1 Metody wykorzystujące łączenie pól	7
II.2 Metody wykorzystujące rozszerzanie pól	10
II.3 Metody mieszane	13
II.4 Metody inteligentne	14
Omówienie i implementacja wybranych algorytmów usuwania przeplotu	18
III.1 Metoda weaving	22
III.3 Metoda blending	23
III.4 Metoda selective blending	24
III.5 Metody discard	25
III.6 Metody doubling	26
III.7 Metody quarter	27
III.8 Progressive scan	28
III.9 Metoda autorska	30
Porównanie metod usuwania przeplotu	33
IV.1 Porównanie wydajnościowe	33
IV.2 Porównanie jakościowe	49
IV.3 Porównanie wykrywania pozycji przestrzennej wskaźników	79
Wnioski i podsumowanie	95
V.1 Wnioski z badań	95
V.2 Możliwości rozwoju	96
V.3 Podsumowanie	97
Bibliografia	99

Wstęp

I.1 Wprowadzenie

Niniejsza praca traktuje o problemie z pogranicza techniki analogowej oraz cyfrowej – usuwaniu przeplotu z sekwencji obrazów.

Dwa pierwsze rozdziały są rozważaniem o teoretycznych aspektach przeplotu oraz jego usuwania, zaś kolejne przedstawiają implementację oraz wyniki analizy wybranych metod usuwania przeplotu, w tym metody autorskiej. Metoda autorska, opisana w rozdziałach 3 – 4, została opracowana w oparciu o istniejące algorytmy graficzne, tak aby spełniała kryteria optymalnej wydajności oraz skuteczności w określonych warunkach. Celem pracy jest porównanie poprzez analizę oraz przedstawienie wyników badań istniejących metod usuwania przeplotu w oparciu o kryterium wydajności oraz skuteczności oraz dowiedzenie, że metoda autorska w określonych warunkach, jest najbardziej odpowiedniejsza.

Pierwszy rozdział niniejszego opracowania przybliży istotę techniki, jaką jest przeplot poprzez omówienie teoretycznych aspektów przeplotu:

- definicji,
- zastosowania,
- przyczyn powstania techniki przeplotu,
- roli we współczesnych systemach wizyjnych,
- usuwania przeplotu z sekwencji obrazów.

Kolejny – drugi rozdział – jest przeglądem istniejących podstawowych algorytmów usuwania przeplotu ze strumieni wideo. Analiza została przeprowadzona w podziale na cztery podgrupy algorytmów:

- metody wykorzystujące łączenie pól obrazu (algorytmy weaving, blending, selective blending oraz inverse telecine),
- metody wykorzystujące rozszerzanie pól obrazu (half-sizing, discard, resize down oraz line doubling),

- metody mieszane (selective blending),
- metody śledzące ruch w sekwencji obrazu (motion compensation).

W trzecim rozdziale opisane zostały sposoby implementacji wybranych algorytmów usuwania przeplotu z sekwencji obrazów w części praktycznej niniejszego opracowania wraz z pseudokodem przedstawiającym realizację poszczególnych algorytmów.

Pierwsza część czwartego rozdziału przedstawia opis techniki testowania wydajności poszczególnych metod usuwania przeplotu wybranych do implementacji. W tej części niniejszego opracowania zostały również zawarte wyniki badań metod w oparciu o kryterium wydajności opatrzone komentarzem autora.

Druga część rozdziału czwartego jest logiczno-strukturalną kontynuacją analizy przeprowadzonej pierwszej jego części. Element różniący stanowi zastosowane kryterium, którym w przypadku części drugiej jest kryterium jakościowe usuwania przeplotu, rozumiane jako subiektywna ocena przetworzonego obrazu w stosunku do obrazu rzeczywistego oraz, już obiektywne, porównanie algorytmów w odniesieniu do liczbowego wykrywania znacznika w obrazie.

Trzecia, ostatnia, część czwartego rozdziału traktuje o skuteczności poszczególnych metod usuwania przeplotu w obrazie w przypadku procesów wykrywania znaczników w obrazie – pod względem jakości ustalania położenia znacznika w przestrzeni trójwymiarowej.

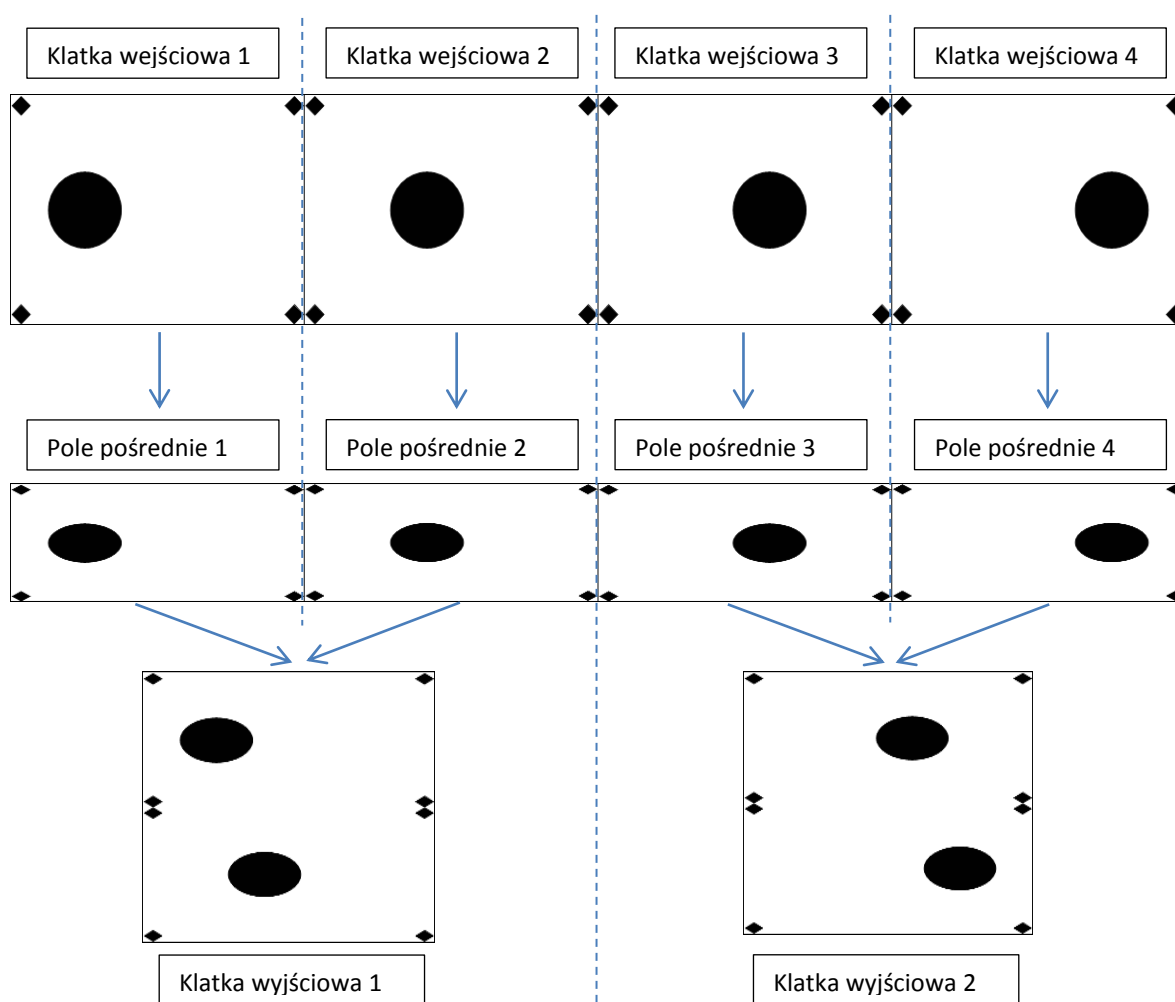
Rozdział piąty zawiera wnioski wynikające z niniejszego opracowania oraz podsumowanie wcześniejszych rozdziałów.

Niniejsze opracowanie zostało przygotowane w oparciu o studium literaturowe oraz badanie empiryczne, opatrzone komentarzami oraz ilustracjami graficznymi.

1.2 Definicja przeplotu

Przeplotem nazywamy technikę wyświetlania obrazu polegającą na wyświetlaniu kolejnych linii obu pól składowych, nieparzystych i parzystych, na przemian [1].

Każda ramka obrazu składa się z dwóch półobrazów, zarejestrowanych w różnym czasie. Półobraz (także nazywany polem obrazu) górny pochodzi z chwili czasowej t_1 , zaś półobraz dolny pochodzi z chwili czasowej t_2 , przy czym każda para półobrazów jest rejestrowana w takim samym odstępie czasu ($t_2 - t_1$). Proces ten został zilustrowany na rysunku (Rysunek 1). W pierwszym rzędzie znajdują się cztery kolejne zarejestrowane obrazy. Rząd poniżej ukazuje cztery kolejne zarejestrowane półobrazy; nieparzyste półobrazy powstały wyłącznie z nieparzystych linii, zaś parzyste – wyłącznie z parzystych linii odpowiednich obrazów. Na dole ukazane są wynikowe ramki obrazu powstałe z pionowego złączenia dwóch półobrazów.



Rysunek 1.1: Ilustracja powstawania obrazu z przeplotem (źródło: opracowanie własne)

1.3 Zastosowanie i geneza przeplotu

Technika ta pozwala na zmniejszenie pasma przenoszenia przesyłanego sygnału lub zwiększenia pozornej rozdzielności obrazu [1].

Przeplot jest szeroko stosowany w technice telewizyjnej – w celu wyeliminowania efektu migotania obrazu oraz w grafice komputerowej – w celu umożliwienia szybkiego dostępu do treści graficznych (patrz: rozdział 1.4).

Początki stosowania przeplotu sięgają lat 20. XX wieku, wraz z pojawieniem się odbiorników telewizyjnych. Wyświetlanie obrazu z częstotliwością stosowaną w kinach (24 klatki na sekundę) na odbiornikach telewizyjnych dawało efekt migotania obrazu [2]. W celu zniwelowania tego efektu, postanowiono dostosować częstotliwość wyświetlania obrazu do częstotliwości napięcia zasilającego (co również wiązało się z uproszczeniem budowy poszczególnych elementów odpowiedzialnych za synchronizację obrazu), co miało pozwolić wyświetlać obraz dwukrotnie częściej (powielając jedną klatkę, aby zachować tempo wyświetlania). Dwukrotny wzrost częstotliwości wiązał się ze zwiększeniem częstotliwości odchylenia poziomego obrazu oraz ze zwiększeniem pasma sygnału nośnego, co przekraczało możliwości ówczesnej techniki. Zdecydowano, aby zamiast wyświetlać całą ramkę dwa razy, w tym samym czasie wyświetlić dwa półobrazy.

Przeplot stosowany jest w technologii telewizyjnej po dziś dzień, ponieważ w mocy była zasada, że każda nowa technologia nie może ograniczać obioru na odbiornikach już istniejących na rynku. Wprowadzany obecnie na całym świecie nowy standard telewizji cyfrowej nie zakłada możliwości wyświetlania obrazu z przeplotem.

Przeplot do niedawna był stosowany we wszystkich standardach nadawania obrazu telewizyjnego (PAL, SECAM, NTSC) oraz w niektórych trybach graficznych starszych monitorów. Standardy PAL oraz SECAM zakładają częstotliwość wyświetlania półobrazów z częstotliwością 50 Hz, zaś NTSC z częstotliwością 60 Hz.

I.4 Przeplot współcześnie

Współcześnie, mogłoby się wydawać, duży postęp technologiczny, skutkujący cyfryzacją praktycznie wszystkich dziedzin związanych z rejestracją obrazu, powinien pozostawić konieczność stosowania przeplotu daleko w przyszłości. Jednakże, mimo upływu lat, technologia ta jest nadal szeroko stosowana.

I.4.1 Przemysł filmowy

Przemysł filmowy jest w dalszym ciągu mocno związany z przeplotem, co jest wynikiem wykorzystania techniki przeplotu w formatach wideo SDTV – 576i (regiony PAL oraz SECAM), 480i (regiony NTSC) oraz HDTV – 720i, 1080i [3] [4]. Wspomniane formaty SDTV (576i, 480i) najczęściej są spotykane na nośnikach VCD, DVD, zaś format HDTV (720i, 1080i) zdobywa swoją popularność na płytach Blu-ray. Obecność przeplotu w formacie SDTV wynikało z tradycji telewizji analogowej – w chwili tworzenia 576i oraz 480i były najbardziej popularnymi trybami wideo.

W przypadku formatu HDTV, popularność 1080i wynika z przewagi tego trybu nad 1080p (przy częstotliwości dekodowania 25 lub 30 klatek na sekundę) – oferuje większą płynność obrazu (możliwość wyświetlania 50 lub 60 klatek na sekundę), zaś obniżenie rozdzielczości pionowej obrazu niwelowane jest dzięki zastosowaniu filtrów usuwających przeplot w odtwarzaczach nośników. W chwili obecnej dopiero tworzony jest standard wideo, w którym tryb 1080i zostanie zastąpiony trybem 1080p (przy częstotliwości kodowania/dekodowania 50 lub 60 klatek na sekundę) [5].

I.4.2 Telewizja przemysłowa

Oddzielną gałęzią rynku, w którym dominują urządzenia analogowe wykorzystujące technikę przeplotu to tzw. telewizja przemysłowa (CCTV - Closed-circuit television), czyli wszelkiego rodzaju systemy podglądu i monitoringu. Tego typu systemy są bardzo powszechne – od monitoringu budynków mieszkalnych i supermarketów aż po duże zakłady oraz ulice miasta; większość urządzeń służących do rejestrowania obrazu jest analogowa, dopiero od niedawna prowadzone są prace nad popularyzacją podobnych systemów z kamerami cyfrowymi [6]. W dodatku, mimo znacznej przewagi urządzeń

cyfrowych nad analogowymi, najczęściej decydującym czynnikiem wyboru jest koszt rozwiązania – według opracowania firmy Axis, dopiero przy instalacji 32 cyfrowych rejestratorów, koszt wdrożenia zrównuje się z rozwiązaniem analogowym [7]. Co więcej, z tego samego powodu, przy cyfryzacji istniejących rozwiązań, często rezygnuje się z wymiany kamer analogowych i modernizuje wyłącznie dalsze etapy infrastruktury.

I.4.3 Grafika komputerowa

Przeplot posiada również zastosowanie w technice komputerowej, a ściślej w niektórych popularnych formatach graficznych, m. in. w formatach GIF oraz PNG. Wykorzystanie metod jest różne pomiędzy formatami, ale idea jest ta sama – umożliwienie szybkiego dostępu do obrazu przy przesyłaniu go w ograniczonym paśmie. Obraz przesyłany jest partiami – np. w formacie GIF, przesyłane jest kolejno – 1/8 linii, 1/8 linii, 2/8 linii oraz w ostatniej partii 4/8 linii. Rozwiązanie to pozwala uzyskać mniej dokładny obraz wcześniej, który z czasem uzupełniany jest o dalsze detale.

I.4.4 Wobulacja

Wobulacja, czyli technologia pozornego zwiększania rozdzielczości obrazu bez konieczności zwiększania liczby pikseli w obrazie, jest opracowana na zasadzie przeplatania się (tylko nie w pionie, ale w poziomie) dwóch półobrazów. Pozwala to na ograniczenie liczbie mikroluster w odbiornikach o połowę, efektywnie redukując koszty produkcji urządzeń. Technologia ta znana jest pod rynkową nazwą „Smooth Picture”, stosowaną przez Samsunga, Toshiba, Panasonic oraz HP – pomysłodawcę rozwiązania.

I.5 Usuwanie przeplotu

Usuwanie przeplotu jest warunkiem koniecznym uzyskania pożądanego obrazu, który wcześniej był poddany działaniu techniki przeplotu. O ile analogowe odbiorniki obrazu doskonale radziły sobie z wyświetlaniem przeplatane obrazu bez widocznych dla ludzkiego oka artefaktów, tak cyfrowe wyświetlacze wymagały opracowania szeregu metod usuwania przeplotu. W obecnej chwili istnieje wiele takich metod – począwszy od prostych, dostępnych praktycznie w każdym urządzeniu (weaving, blending, progressive scan lub line doubling), a skończywszy na skomplikowanych metodach dostępnych na sprzęcie wysokiej klasy do obróbki obrazu (motion compensation).

Przegląd istniejących algorytmów usuwania przeplotu

II.1 Metody wykorzystujące łączenie pól

Usuwanie przeplotu przy pomocy łączenia pól polega na wykorzystaniu informacji z obu pól składowych do stworzenia klatki wynikowej.

II.1.1 Weaving

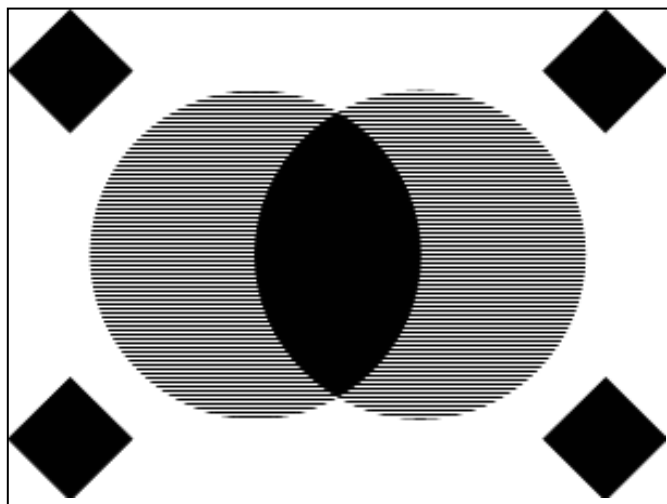
Jest to podstawowa metoda usuwania przeplotu z sekwencji obrazu. Polega ona na naprzemiennym wyświetlaniu kolejnych linii z obu pól składowych. Weaving jest domyślną metodą stosowaną przy zapisie obrazu cyfrowego przy konwersji ze źródła analogowego.

Zalety metody [8]:

- prostota,
- brak potrzeby ingerencji w źródło obrazu w znakomitej większości przypadków,
- brak utraty jakości obrazu w scenach, w których nie występuje ruch lub zmiany w obrazie następują z bardzo małą prędkością.

Wady metody [9]:

- pojawienie się w obrazie artefaktów w postaci tzw. grzebienia, gdy pomiędzy dwiema ramkami w obrazie pojawi się jakikolwiek ruch (patrz: rysunek II.1).



Rysunek II.1: Ilustracja przykładowej ramki wynikowej przetworzonej metodą weaving (źródło: opracowanie własne)

II.1.2 Blending

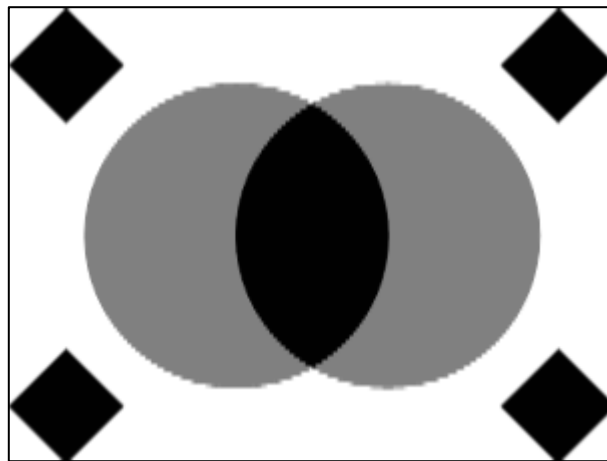
Blending polega na uśrednianiu poszczególnych linii pól nieparzystych i parzystych, tworząc jedną ramkę obrazu. Każda pozioma linia jest uśredniania z sąsiadującymi liniami – poprzedzającą oraz kolejno następującą, z odpowiednimi wagami – 0,5 dla linii bieżącej oraz 0,25 dla dwóch pozostałych.

Zalety metody [9]:

- prostota.

Wady metody [9]:

- znaczna utrata szczegółowości obrazu,
- powstawanie efektu smużenia (patrz: rysunek II.2), w szczególności w dynamicznych scenach; efekt ten występuje ze względu na łączenie informacji z dwóch różnych momentów czasowych.



Rysunek II.2: Ilustracja przykładowej ramki wynikowej przetworzonej metodą blending (źródło: opracowanie własne)

II.1.3 Selective blending

Selective blending jest kombinacją metod weaving oraz blending. W ramach tej metody obszary, które nie zmieniły się od ostatniej ramki, są wyświetlane za pomocą metody weaving. Obszary, które uległy zmianie od ostatniej ramki, są uśredniane za pomocą

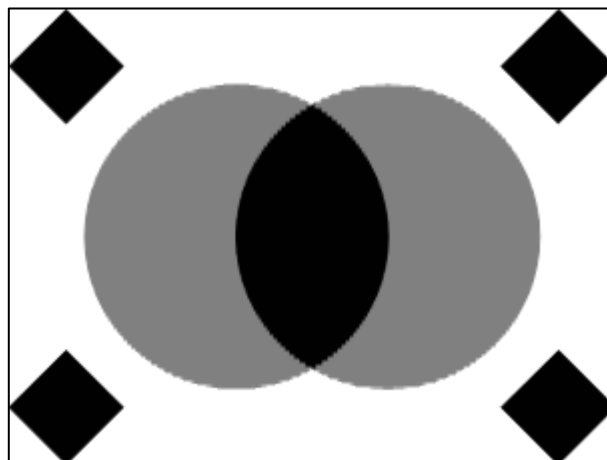
metody blending. Pozwala to zachować pełną rozdzielczość pionową obrazu, połowę rozdzielczości czasowej oraz powodują mniej zniekształceń niż poszczególne metody.

Zalety metody [9]:

- brak utraty jakości obrazu w mało dynamicznych scenach.

Wady metody [9]:

- utrata szczegółowości obrazu, ale tylko w zmieniających się pomiędzy kolejnymi ramkami obszarach obrazu,
- powstawanie efektu smużenia, w szczególności w dynamicznych scenach.



Rysunek II.3: Ilustracja przykładowej ramki obrazu przetworzonej metodą selective blending (źródło: opracowanie własne)

II.1.4 Inverse telecine

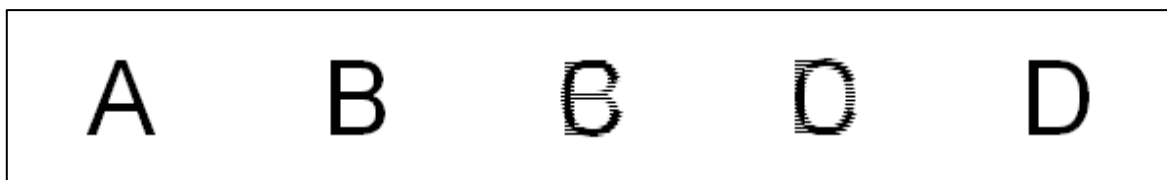
Metoda ta nie jest stricte metodą usuwania przeplotu. Metodę tą można zastosować w momencie, gdy obraz został przekonwertowany z progresywnego w standardzie PAL na obraz z przeplotem w standardzie NTSC (tzw. telecine) oraz gdy znane są parametry pierwotnej konwersji. Metoda pozwala na uzyskać obraz w 100% zgodny z oryginałem.

Zalety metody [10]:

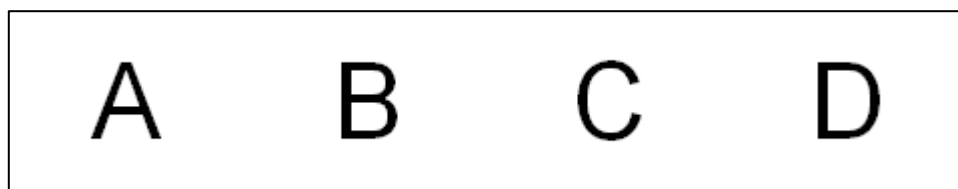
- pozwala uzyskać obraz zgodny w 100% z oryginałem.

Wady metody [10]:

- wymagana jest znajomość parametrów pierwotnej konwersji,
- można ją stosować tylko w przypadku obrazów przekonwertowanych ze standardu PAL do NTSC.



Rysunek II.4: Przykładowa sekwencja pięciu kolejnych klatek obrazu przetworzonego metodą telecine (źródło: opracowanie własne)



Rysunek II.5: Przykładowa sekwencja czterech ramek obrazu odtworzonego po konwersji PAL – NTSC metodą inverse telecine (źródło: opracowanie własne)

II.2 Metody wykorzystujące rozszerzanie pól

Usuwanie przeplotu metodami wykorzystującymi rozszerzanie pól polega w głównej mierze na manipulacji rozmiarem poszczególnych pól składowych ramki. Obraz wynikowy może składać się z jednej lub dwóch ramek, co w drugim przypadku oznacza dwukrotne zwiększenie rozdzielczości czasowej sekwencji obrazów.

II.2.1 Discard

Discard (lub też half-sizing) - metoda polega na wyświetlaniu każdego pola ramki oddzielnie bez rozszerzania do rozmiaru oryginalnego obrazu. Obraz wynikowy zachowuje pełną rozdzielczość pionową oraz czasową oryginału, jednak nie nadaje się do zwyczajnego oglądania ze względu na błędny stosunek rozmiarów.

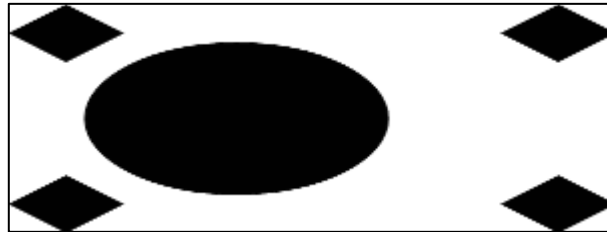
Zalety metody [9]:

- prostota,

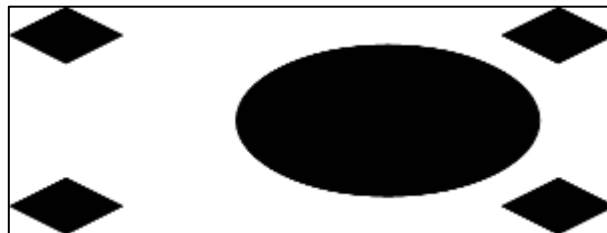
- wysoka rozdzielczość czasowa obrazu wynikowego.

Wady metody [9]:

- obraz wynikowy ma stosunek rozmiarów różny od oryginalnego.



Rysunek II.6: Ramka obrazu przetworzonego metodą discard, usunięte pole dolne (źródło: opracowanie własne)



Rysunek II.7: Ramka obrazu przetworzonego metodą discard, usunięte pole górne (źródło: opracowanie własne)

II.2.2 Double

Algorytm double występuje w dwóch wersjach.

Pierwsza, polega na usunięciu jednego pola z ramki obrazu przy jednoczesnym rozszerzeniu tego pola poprzez zdublowanie każdej jego linii. W zależności od usuwanego pola, przetwarzane są pola górne lub dolne ramki.

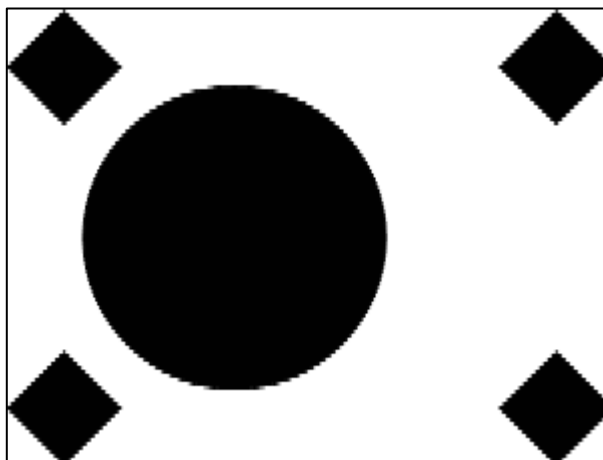
W drugiej, nie jest usuwane żadne pole; pola są rozdzielane i dalej rozszerzane do rozmiaru oryginalnej ramki.

Zalety metody [9]:

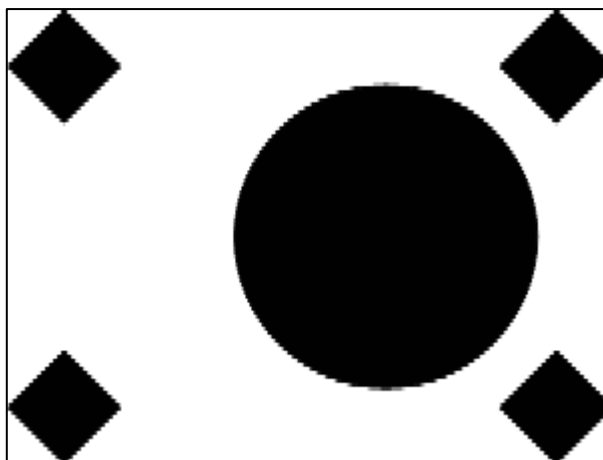
- prostota,
- w przypadku wariantu drugiego – dwukrotnie wyższa rozdzielczość czasowa obrazu wynikowego.

Wady metody [9]:

- połowę niższą rozdzielczość pionowa obrazu wynikowego.



Rysunek II.8: Ramka obrazu przetworzonego metodą double, dublowane linie pola górnego (źródło: opracowanie własne)



Rysunek II.9: Ramka obrazu przetworzonego metodą double, dublowane linie pola dolnego (źródło: opracowanie własne)

II.2.4 Quarter

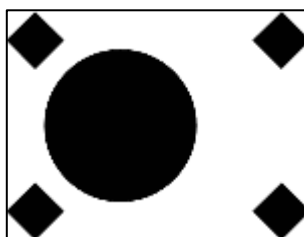
Quarter, lub też resize down, polega na pomniejszeniu całej ramki obrazu dwukrotnie. W zależności od kolejności pól w obrazie mogą powstać dwie wersje ramek wynikowych, linie półobrazu rozpoczynającego ramkę zostają przeniesione do ramki wynikowej.

Zalety metody [9]:

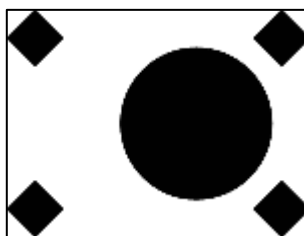
- prostota,
- dwukrotnie wyższa rozdzielczość czasowa (w przypadku użycia obu klatek wynikowych w sekwencji końcowej).

Wady metody [9]:

- połowę niższa rozdzielczość pionowa i pozioma każdej ramki obrazu wynikowego.



Rysunek II.10: Ramka uzyskana z obrazu poprzez zastosowanie metody quarter, z zachowaniem linii pola górnego (źródło: opracowanie własne)



Rysunek II.11: Ramka uzyskana z obrazu poprzez zastosowanie metody quarter, z zachowaniem linii pola dolnego (źródło: opracowanie własne)

II.3 Metody mieszane

Metody mieszane integrują cechy metod łączących i rozszerzających pola, co pozwala na połączenie ich zalet i zniwelowanie wpływu wad obrazu poddanego obróbce.

II.3.1 Progressive scan

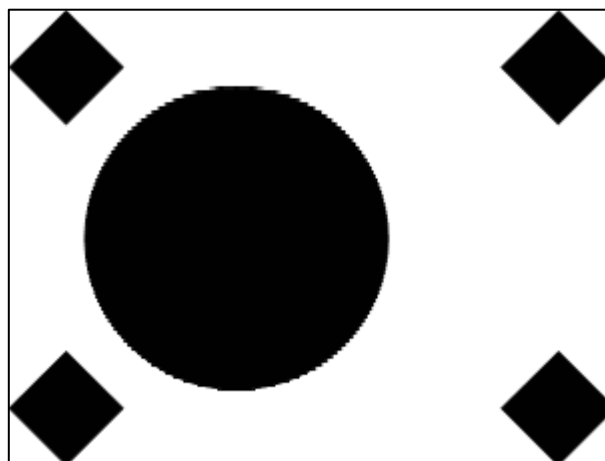
Progressive scan jest kombinacją metod weaving oraz line doubling. W ramach tej metody obszary, które nie zmieniły się od ostatniej ramki, są wyświetlane za pomocą metody weaving. Obszary, które uległy zmianie od ostatniej ramki, są uśredniane za pomocą metody line doubling. Od metody selective blending różni się tym, że z każdego pola jest tworzona oddzielna ramka, a więc rozdzielczość czasowa jest dwa razy większa.

Zalety metody [9]:

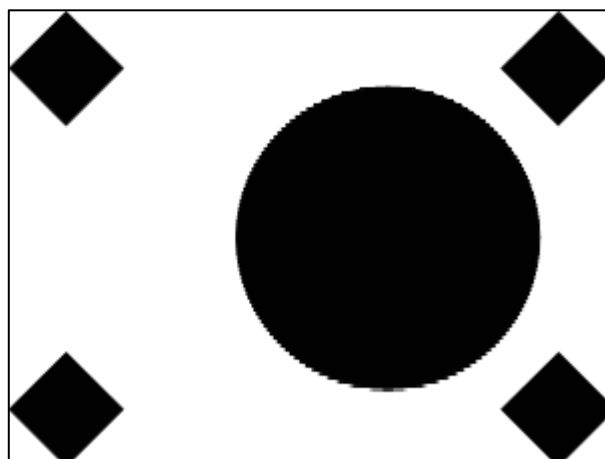
- wysoka rozdzielczość czasowa obrazu wynikowego,
- brak utraty jakości obrazu w mało dynamicznych scenach.

Wady metody [9]:

- utrata szczegółowości obrazu, ale tylko w zmieniających się obszarach obrazu.



Rysunek II.12: Pierwsza ramka obrazu przetworzona metodą progressive-scan (źródło: opracowanie własne)



Rysunek II.13: Druga ramka obrazu przetworzona metodą progressive-scan (źródło: opracowanie własne)

II.4 Metody inteligentne

Metody inteligentne potrafią adaptować się w zależności od przetwarzanego obrazu, dzięki czemu, dla danej sekwencji obrazu, rezultat jest możliwie najlepszy. Dzieje się to dzięki dokładnej analizie ruchu w obrazie, zarówno w przestrzeni, jak i w czasie.

II.4.1 Motion compensation

Motion compensation jest techniką, która nie tylko wykorzystuje wszystkie powyższe metody do usuwania przeplotu, ale także poddaje obraz analizie w dziedzinie czasu. Technika ta pozwala osiągać dużo lepsze efekty, ponieważ wykorzystuje informacje z wielu ramek obrazu, zamiast jednej lub dwóch. Wymaga ona także zastosowania odpowiednich algorytmów wykrywania zmiany sceny w obrazie – w przeciwnym przypadku algorytm będzie wyszukiwał ruch pomiędzy dwoma odrębnymi scenami, w rezultacie powodując błędy w obrazie.

Na przestrzeni ostatnich lat powstał szereg rozwiązań implementujących założenia zasad motion compensation, których wyniki były prezentowane w różnych opracowaniach i periodykach:

- motion adaptive deinterlacing via edge pattern recognition [10],
- rough set approach [11],
- application of bayesian belief network in reliable analysis [12],
- spatial-temporal content-adaptive algorithm [13],
- artifact detection for motion compensation [14],
- fuzzy filter for noise reduction method [15],
- MRF-based deinterlacing algorithm with exemplar-based refinement [16],
- fuzzy motion-adaptive interpolation with picture repetition detection [17],
- spatial-temporal line-warping based deinterlacing method [18],
- fine edge-preserving deinterlacing algorithm [19],
- edge direction from analysis of the DCT coefficient distribution [20].

Niestety, bardzo trudne lub wręcz niemożliwe jest przybliżenie czytelnikowi dokładnego wyniku działania powyższych metod, zważywszy na fakt, że autorowi niniejszego opracowania nie udało się pozyskać kodu źródłowego bądź działającej implementacji tych algorytmów. Co więcej, same metody przedstawione w przytoczonych opracowaniach są opisane w sposób uniemożliwiający samodzielne odtworzenie poszczególnych algorytmów

lub są one (algorytmy) niekompletne, a w najlepszym przypadku, ich kluczowe parametry nie są określone w publikacjach.

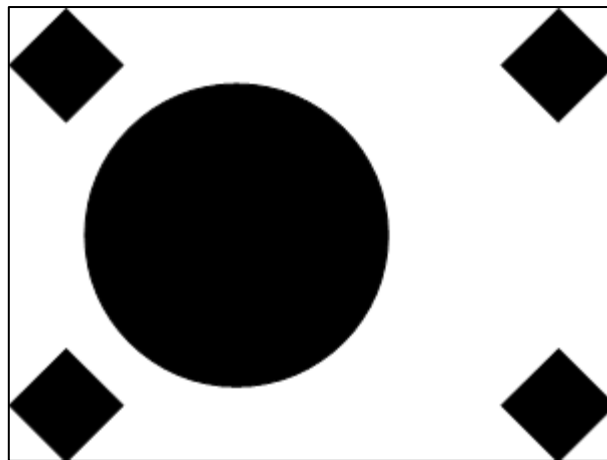
Zaprezentowany w opisie metod efekt ich działania, z mniejszym lub większym powodzeniem naśladuje proces odtwarzania obrazu oryginalnego, dając wyniki podobne do zobrazowanych rysunkami II.14 oraz II.15 względem nieprzetworzonego obrazu z przeplotem.

Zalety metod [9]:

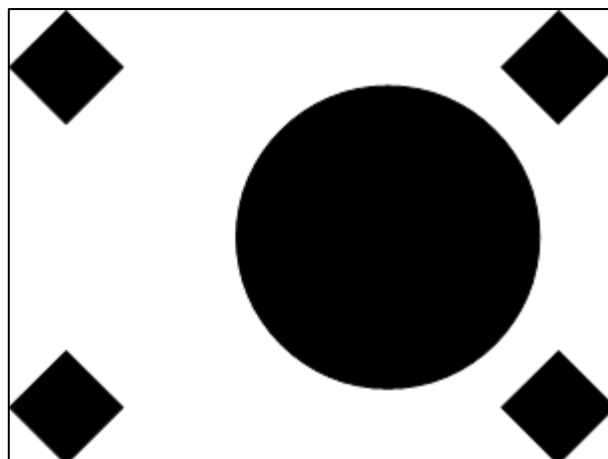
- pozwala uzyskać obraz bardzo zbliżony (a w niektórych przypadkach wizualnie 100% zgodny) do oryginału,
- wysoka rozdzielczość czasowa obrazu wynikowego.

Wady metod [9]:

- czasochłonność, metody te wymagają znacznych nakładów obliczeniowych,
- wysokie koszty obróbki (w przypadku implementacji sprzętowych),
- brak szeroko dostępnych rozwiązań.



**Rysunek II.14: Ramka pierwsza obrazu przetworzonego metodą motion compensation
(źródło: opracowanie własne)**



Rysunek II.15: Ramka druga obrazu przetworzonego metodą motion compensation

Omówienie i implementacja wybranych algorytmów usuwania przeplotu

Rozdział ten jest poświęcony opisowi sposobu implementacji programowej poszczególnych algorytmów wybranych przez autora niniejszego opracowania do badań.

Rozdział ten nie pokazuje dosłownego kodu programu, a tylko tzw. pseudokod, mający na celu objaśnienie mechanizmów działania poszczególnych algorytmów.

Do rzeczywistej implementacji kodu badający posłużył się językiem C, zaś środowiskiem programistycznym było Visual C++ Studio Express 2008 oraz Visual C++ Studio Express 2010.

Język C został wybrany, ponieważ jest on dobrze znany autorowi i pozwala na daleko idące optymalizacje kodu pod względem wydajności [21] [22].

Wybór środowiska programistycznego Visual C++ Studio Express był podyktowany wieloma czynnikami:

- jest to środowisko darmowe do zastosowań niekomercyjnych oraz ogólnodostępne,
- projekty do VC++S były dołączone do biblioteki ArToolkit, wykorzystanej do wyszukiwania znaczników w obrazie,
- środowisko to jest znane badającemu.

Do przedstawienia pseudokodu, autor niniejszego opracowania posłużył się następującymi oznaczeniami oraz uproszczeniami:

- y_p - oznacza liczbę równą połowie liczby pionowych pikseli pełnej ramki,
- y_r - oznacza liczbę równą liczby pionowych pikseli pełnej ramki,
- x_r - oznacza liczbę równą liczby poziomych pikseli pełnej ramki,
- $P'_{j,2i}$ - oznacza piksel ramki wynikowej o współrzędnych przypisie dolnym,
- $P_{j,i}$ - oznacza piksel ramki wejściowej o współrzędnych przypisie dolnym,
- $P^{-1}_{j,i}$ - oznacza poprzednią ramkę wejściową w sekwencji obrazów,

- $P'_{1,j,i}$ - oznacza piksel pierwszej ramki wynikowej o współrzędnych w przypisie dolnym,
- $P'_{2,j,i}$ - oznacza piksel drugiej ramki wynikowej o współrzędnych w przypisie dolnym,
- $P'_{j,2i} = P_{j,i}$ - jest uproszczeniem, gdyż przypisanie naprawdę odbywa się na poziomie każdego koloru składowego, a nie tylko na poziomie całego piksela.

Kod rzeczywisty programu może różnić się od opisanych algorytmów, wynika to z optymalizacji kodu przeprowadzonej przez autora opracowania; efekt końcowy jest jednak zawsze identyczny, zgodny z podanymi algorytmami.

Badający, bazując na swojej wiedzy programistycznej oraz literaturze fachowej [22], dokonał następujących optymalizacji względem oryginalnych wersji algorytmów (w tym również zmodyfikowanej wersji algorytmu Gunnara Thalina):

- przypisanie do zmiennej powtarzających się obliczeń arytmetycznych – pozwala to na wykonanie danej operacji arytmetycznej jeden raz zamiast kilku razy, np.

```
// przypisanie wielokrotnie używanych sum do zmiennych
temp=y+x+c;
temp2=temp-cxs;

// w tym przypadku pozwala zastąpić osiem operacji
// arytmetycznych tylko dwiema
if (lastFrame[temp]!=baseFrame[temp])
{
    finalFrame[temp]=finalFrame[temp2];
}

if (lastFrame[temp2]!=baseFrame[temp2])
{
    finalFrame2[temp2]=finalFrame2[temp];
}
```

- przeprowadzanie operacji dzielenia i mnożenia za pomocą operatorów przesunięcia bitowego; przesunięcie bitowe na większości mikroprocesorów zajmuje jeden cykl zaś operacje mnożenia i dzielenia odpowiednio między 3 a 11 cykli oraz od 40 do 80 cykli procesora, za [22], np.:


```
// zastosowanie przesunięcia bitowego
// w prawo o 1 zamiast dzielenia przez 2
baseFrame[tempb]=((iB0+iB2)>>1);
baseFrame[tempg]=((iG0+iG2)>>1);
baseFrame[tempr]=((iR0+iR2)>>1);
```

- grupowanie operacji arytmetycznych w kolejności, aby uniknąć powielania tych samych operacji, np.:

```
// użycie jednej operacji mnożenia przez 2
// i jednej operacji dzielenia przez 2
finalFrame[temp]=((baseFrame[temp-cxs]+
                    (baseFrame[temp]<<1)+
                    baseFrame[temp+cxs])>>2);
```

zamiast

```
// użycie trzech operacji dzielenia
finalFrame[temp]=(baseFrame[temp-cxs]>>2)+
                  (baseFrame[temp]>>1)+
                  (baseFrame[temp+cxs]>>2);
```

- użycie funkcji *memcpy* zamiast zagnieżdżonych pętli *for* do kopiowania linii lub całych struktur obrazu, np.:

```
memcpy(finalFrame, baseFrame, depth*xsize*yssize * sizeof (ARUint8));
```

zamiast

```
for (int c=0; c<depth_; c++)
{
    for (int x=0; x<cxs; x+=depth)
    {
        for (int y=cxs; y<cys; y+=cxs)
        {
            temp=y+x+c;
            finalFrame[temp]=baseFrame[temp];
        }
    }
}
```

- ustawianie możliwych wariantów wyników porównań instrukcji warunkowych w kolejności zgodnie z ich prawdopodobieństwem wystąpienia, co pozwala oszczędzić

czas sprawdzania prawdziwości mniej prawdopodobnych wyników (iteracja kończy się na pierwszym spełnionym warunku), np.:

```
// w przypadku mało dynamicznych sekwencji, prawdopodobieństwo
// stwierdzenia braku ruchu pomiędzy kolejnymi ramkami jest bardziej
// prawdopodobne niż znalezienie zmiany obrazu
if (lastFrame[temp]==baseFrame[temp])
{
    finalFrame[temp]=baseFrame[temp];
}
else
{
    finalFrame[temp]=((baseFrame[temp]+baseFrame[temp+cxs])>>1);
}
```

- „odwijanie” pętli, czyli dwukrotne zmniejszanie powtórzeń pętli w celu uniknięcia niepotrzebnych instrukcji warunkowych oraz grupowanie podobnych wyrażeń istotnych dla części iteracji pętli, np.:

```
// w tym przypadku mamy n wystąpień pętli wewnętrznej, piksele
// są przypisywane dla linii i oraz i+1 jednocześnie
for (int c=0; c<depth_; c++)
{
    for (int x=0; x<cxs; x+=depth)
    {
        for (int y=cxs; y<cys; y+=cxs<<1)
        {
            temp=y+x+c;
            finalFrame[temp]=baseFrame[temp];
            finalFrame2[temp]=baseFrame[temp+cxs];
        }
    }
}
```

zamiast

```
// w tym przypadku mamy 2n wystąpień pętli wewnętrznej, piksele
// są przypisywane dla linii i oraz i+1 oddzielnie, a co więcej,
// dopiero po sprawdzeniu warunku wystąpienia odpowiedniej linii
for (int c=0; c<depth_; c++)
{
    for (int x=0; x<cxs; x+=depth)
    {
        for (int y=cxs; y<cys; y+=cxs)
        {
            temp=y+x+c;

            if (y % cxs<<1 == 0)
            {
                finalFrame[temp]=baseFrame[temp];
            }
        }
    }
}
```

```

    }
    if (y % cxs << 1 == cxs)
    {
        finalFrame2[temp]=baseFrame[temp];
    }
}
}
}

```

Wybór algorytmów do implementacji związany jest ich powszechnością, a także łatwością w kodowaniu, wynikającej z szerokiej dostępności materiałów na ich temat.

Algorytmy wymienione w punkcie IV.4.1 nie są przedmiotem tego opracowania ze względu na brak ogólnodostępnych źródeł wspomnianych metod. Algorytmy te są bardzo złożone i nie jest możliwe dokładne określenie przebiegu algorytmu, co również wynika z bardzo ogólnego sposobu opisu w tekstach źródłowych, oraz, co gorsze, z najczęściej pominiętych istotnych fragmentów objaśnień metod. Nie są dostępne również kody programowe oraz wersje działające programów, co dalej wyklucza jakąkolwiek analizę porównawczą.

III.1 Metoda weaving

III.1.1 Opis

Metoda weaving jest najprostszą metodą w implementacji – wykonanie polega na przyporządkowaniu kolejnym liniom ramki wyjściowej linii pół ramki wejściowej (górnych i dolnych naprzemiennie). W zależności od sterowników urządzenia przechwytyjącego strumień wejściowy obrazu, obraz cyfrowy może przyjmować dwie formy – z ramką podzieloną na dwa oddzielne pola lub z ramką poddaną działaniu metody weaving na wyjściu. Poniższy pseudokod obrazuje konieczne kroki do uzyskania ramki wyjściowej w przypadku pierwszym; drugi przypadek nie wymaga żadnej obróbki.

Dla uproszczenia opisu i kodu pozostałych metod, zostanie przyjęte, że ramka wejściowa tychże metod pochodzi z przypadku drugiego.

III.2.2 Pseudokod

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do yp-1:
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xr-1:
        // Linia 2i
         $P'_{j,2i} = P_{j,i}$ 
        // Linia 2i+1
         $P'_{j,2i+1} = P_{j,yp+i}$ 
```

III.3 Metoda blending

III.3.1 Opis

Implementacja metody blending polega na uśrednianiu kolejnych linii ramki wejściowej, z wagami:

- 0,25 dla linii i-1 oraz i+1,
- 0,5 dla linii i.

W przypadku pierwszej i ostatniej linii ramki, linie bieżące oraz przyległe (linia następną dla początku ramki oraz linia poprzedzająca dla ostatniej ramki) traktowane są jako równoważne z wagami 0,5.

III.3.2 Pseudokod

```
// Linia 0
 $P'_{j,0} = 0,5 \cdot P_{j,0} + 0,5 \cdot P_{j,1}$ 
// Linie od 1 do yr-2
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 1 do yr-2:
    // Pętla wewnętrzna, liczba powtórzeń
    // równa szerokości ramki w pikselach
    dla j od 0 do xr-1:
        // Linia i
         $P'_{j,i} = 0,25 \cdot P_{j,i-1} + 0,5 \cdot P_{j,i} + 0,25 \cdot P_{j,i+1}$ 
// Linia yr-1
 $P'_{j,yp-1} = 0,5 \cdot P_{j,yp-2} + 0,5 \cdot P_{j,yp-1}$ 
```

III.4 Metoda selective blending

III.4.1 Opis

Implementacja algorytmu selective blending jest połączeniem implementacji metod weaving oraz blending. Każdy punkt ramki wejściowej jest porównywany z odpowiadającym mu punktem poprzedniej ramki wejściowej. Gdy wartości reprezentowane przez te punkty są takie same – algorytm przetwarza punkt metodą weaving; gdy wartości są od siebie różne – algorytm wykorzystuje metodę blending.

III.4.2 Pseudokod

```
// Linia 0
// Pętla wewnętrzna, liczba powtórzeń równa
// szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Warunek dla przetwarzania typu blending
    jeśli  $P_{j,i}^{-1} \neq P_{j,i}$ :
         $P'_{j,i} = 0.5 \cdot P_{j,i} + 0.5 \cdot P_{j,i+1}$ 
    // Warunek dla przetwarzania typu weaving
    inaczej:
         $P'_{j,i} = P_{j,i}$ 
// Linie od 1 do yr-2
// Pętla zewnętrzna, liczba powtórzeń
// równa wysokości ramki w pikselach
dla i od 1 do yr-2:
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xp-1:
        // Linia i
        // Warunek dla przetwarzania typu blending
        jeśli  $P_{j,i}^{-1} \neq P_{j,i}$ :
            // Uśrednianie zgodne z algorytmem blending
             $P'_{j,i} = 0.25 \cdot P_{j,i-1} + 0.5 \cdot P_{j,i} + 0.25 \cdot P_{j,i+1}$ 
        // Warunek dla przetwarzania typu weaving
        inaczej:
            // Linia nie jest zmieniana
             $P'_{j,i} = P_{j,i}$ 
// Linia yr-1
// Pętla wewnętrzna, liczba powtórzeń równa
// szerokości ramki w pikselach
dla j od 0 do  $x_w-1$ :
    // Warunek dla przetwarzania typu blending
```

```

jeśli  $P_{j,i}^{-1} \neq P_{j,i}$ :
    // Uśrednianie zgodne z algorytmem blending
     $P'_{j,yr-1} = 0.5 \cdot P_{j,yr-2} + 0.5 \cdot P_{j,yr-1}$ 
// Warunek dla przetwarzania typu weaving
inaczej:
    // Linia nie jest zmieniana
     $P'_{j,yr-1} = P_{j,yr-1}$ 

```

III.5 Metody discard

III.5.1 Opis

W przypadku rodziny metod discard możemy mówić o trzech metodach: discard lower, discard upper oraz discard. W przypadku discard lower, algorytm usuwa wszystkie linie ramki należące do dolnego pola obrazu. Podobnie się dzieje w przypadku algorytmu discard upper, z tą różnicą, że usuwanymi liniami są linie ramki należące do pole górnego. Ostatnia z metod, discard, nie usuwa żadnych linii, raczej rozdziela je od siebie, tworząc dwie oddzielne ramki wynikowe, dając dodatkowo dwukrotnie większą od bazowej liczbę klatek na sekundę. Wszystkie algorytmy z powyższych produkują obraz mniejszy pionowo dwukrotnie, z innym niż oryginalnie, stosunkiem szerokości do wysokości ramki.

III.5.2 Pseudokod metody discard lower

```

// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń
    // równa szerokości ramki w pikselach
    dla j od 0 do  $x_w-1$ :
        // Linia i
         $P'_{j,i} = P_{j,2i}$ 

```

III.5.3 Pseudokod metody discard upper

```

// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń równa szerokości ramki w
    // pikselach
    dla j od 0 do  $x_w-1$ :
        // Linia i
         $P'_{j,i} = P_{j,2i+1}$ 

```

III.5.4 Pseudokod metody discard (wersja dwuklatkowa)

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do  $x_w-1$ :
        // Linia i
        // Ramka A
         $P'_{j,i} = P_{j,2i}$ 
        // Ramka B
         $P'_{j,i} = P_{j,2i+1}$ 
```

III.6 Metody doubling

III.6.1 Opis

Doubling jest rodziną metod analogiczną do rodziny discard, z jedną podstawową różnicą – każda ramka wyjściowa, będąca wynikiem przetwarzania metod doubling, zachowuje pełną rozdzielczość pionową. Aby taki efekt uzyskać, należy zgodnie z algorytmem, podwajać odpowiednie linie pół. Dla double upper, podwajana jest każda nieparzysta linia ramki wejściowej, zastępując kolejne parzyste linie. Dla double lower, to linie parzyste są podwajane, nadpisując linie pochodzące pola górnego. Ostatni z algorytmów, line doubling, pozwala na połączenie dwóch powyższych metod – na koniec dając dwie ramki o pełnej wysokości, zgodne z oryginalnymi stosunkami rozmiarów.

III.6.3 Pseudokod metody double lower

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do  $x_w-1$ :
        // Linia i
         $P'_{j,2i} = P_{j,2i}$ 
        // Linia i+1
         $P'_{j,2i+1} = P_{j,2i}$ 
```

III.6.3 Pseudokod metody double lower

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń
    // równa szerokości ramki w pikselach
    dla j od 0 do  $x_w-1$ :
        // Linia i
         $P'_{j,2i} = P_{j,2i+1}$ 
        // Linia i+1
         $P'_{j,2i+1} = P_{j,2i+1}$ 
```

III.6.4 Pseudokod metody line doubling

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do  $y_p-1$ :
    // Pętla wewnętrzna, liczba powtórzeń
    // równa szerokości ramki w pikselach
    dla j od 0 do  $x_r-1$ :
        // Linia i
        // Pixel przetwarzany metodą double upper
         $P_{1j,2i} = P_{j,2i}$ 
        // Pixel przetwarzany metodą double lower
         $P_{2j,2i} = P_{j,2i+1}$ 
        // Linia i+1
        // Pixel przetwarzany metodą double upper
         $P_{1j,2i+1} = P_{j,2i}$ 
        // Pixel przetwarzany metodą double lower
         $P_{2j,2i+1} = P_{j,2i+1}$ 
```

III.7 Metody quarter

III.7.1 Opis

Metody quarter charakteryzują się przede wszystkim dwukrotnie mniejszą rozdzielczością ramki wynikowej, zarówno w pionie, jak i w poziomie. Sama metoda polega na przenoszeniu tylko jednej czwartej pikseli ramki początkowej, czyli tylko połowie pola górnego lub dolnego, dla zachowania odpowiednich rozmiarów klatki końcowej. Podobnie, jak w przypadku metod discard oraz double, quarter może posiadać trzy wersje obrazu wynikowego: jedna wersja dla ramki utworzonej z pola górnego, druga wersja dla ramki utworzonej z pola dolnego lub też trzecia możliwość – może składać się z obu wymienionych ramek jednocześnie.

III.7.2 Pseudokod metody quarter, przetwarzane jest pole górne

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do yp-1:
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xr/2-1:
        // Linia i
         $P'_{j,i} = P_{2j,2i}$ 
```

III.7.3 Pseudokod metody quarter, przetwarzane jest pole dolne

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do yp-1:
    // Pętla wewnętrzna, liczba powtórzeń
    // równa szerokości ramki w pikselach
    dla j od 0 do xr/2-1:
        // Linia i
         $P'_{j,i} = P_{2j,2i+1}$ 
```

III.7.4 Pseudokod metody quarter - dwuklatkowej

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do yp-1:
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xr/2-1:
        // Linia i
         $P'_{1j,i} = P_{2j,2i}$ 
         $P'_{2j,i} = P_{2j,2i+1}$ 
```

III.8 Progressive scan

III.8.1 Opis

Wynikiem przetwarzania ramki zgodnie z algorytmem progressive scan są dwie ramki wynikowe. Każda z nich składa się z obszarów nieprzetworzonych (algorytm weaving) oraz obszarów przetworzonych algorytmami double upper (dla ramki pierwszej) oraz double lower (dla ramki drugiej). Obszary nieprzetworzone występują w przypadku nie wykrycia zmiany pomiędzy dwiema kolejnymi ramkami sekwencji (ramka bieżąca oraz ramka poprzednia); jeśli pomiędzy ramkami wystąpił ruch w obrazie, dany obszar jest procesowany.

III.8.2 Pseudokod

```
// Pętla zewnętrzna, liczba powtórzeń równa
// połowie wysokości ramki w pikselach
dla i od 0 do yp-1:
    // Pętla wewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xr-1:
        // Linia i
        // Przypadek, kiedy dany punkt ramki bieżącej oraz
        // ten sam punkt ramki poprzedniej są takie same
        jeśli  $P_{j,2i} = P_{j,2i}^{-1}$ :
            // Punkty nie są zmieniane, tożsame
            // z przetworzeniem metodą weaving
             $P'_{1,j,2i} = P_{j,2i}$ 
             $P'_{2,j,2i} = P_{j,2i}$ 
        // Przypadek, kiedy dany punkt ramki bieżącej
        // oraz ten sam punkt ramki poprzedniej są różne
        inaczej:
            // Punkt przetworzony zgodnie z algorytmem
            // discard lower
             $P'_{1,j,2i} = P_{j,2i}$ 
            // Punkt przetworzony zgodnie z algorytmem
            // discard upper
             $P'_{2,j,2i} = P_{j,2i+1}$ 
        // Linia i+1
        // Przypadek, kiedy dany punkt ramki bieżącej
        // oraz ten sam punkt ramki poprzedniej są takie same
        jeśli  $P_{j,2i+1} = P_{j,2i+1}^{-1}$ :
            // Punkty nie są zmieniane
             $P'_{1,j,2i+1} = P_{j,2i+1}$ 
             $P'_{2,j,2i+1} = P_{j,2i+1}$ 
        // Przypadek, kiedy dany punkt ramki bieżącej
        // oraz ten sam punkt ramki poprzedniej są różne
        inaczej:
            // Punkt przetworzony zgodnie z algorytmem
            // discard lower
             $P'_{1,j,2i+1} = P_{j,2i}$ 
            // Punkt przetworzony zgodnie z algorytmem
            // discard upper
             $P'_{2,j,2i+1} = P_{j,2i+1}$ 
```

III.9 Metoda autorska

III.9.1 Opis

Metoda, opracowana przez autora tej pracy na jej potrzeby, jest niespokrewniona z metodami poprzednimi. Korzysta ona zupełnie z innego podejścia do wykrywania efektu przeplotu. Idea tego rozwiązania została zaczerpnięta z filtrów „area based deinterlace” oraz „smooth deinterlace” autorstwa Gunnara Thalina, gruntownie zmieniona na potrzeby związane z odnajdywaniem wskaźników w sekwencjach obrazów z przeplotem [29]. Polega ona w głównej mierze na przypisaniu porównaniu trzech kolejnych linii obrazu pod kątem występowania zmian pomiędzy liniami. Oczywistym problemem jest fakt, że różnice pomiędzy liniami mogą wynikać nie tylko w efekcie obecności przeplotu w obrazie, ale także z powodu występowania tam krawędzi zarejestrowanych obiektów. W tym celu punktom kolejnych linii zostały przypisane wartości jasności, z odpowiednimi wagami dla poszczególnych kolorów składowych [24]. Następnym krokiem było zniwelowanie wpływu występowania rzeczywistych krawędzi na wykrywanie przeplotu przez algorytm. Zostało to osiągnięte poprzez porównanie trzech par punktów: punktów linii i oraz $i+1$, linii $i+1$ oraz $i+2$, a także linii i oraz $i+2$ – jeśli suma różnic poszczególnych wartości z przypisanymi wagami przekroczy próg istotności – dany punkt jest interpolowany na podstawie kolejnych linii należących do jednego pola.

Metoda może mieć dwa warianty – wariant pierwszy, badamy i interpolujemy tylko jedno pole składowe – wtedy sekwencja wynikowa zachowa liczbę klatek na sekundę obrazu oryginalnego oraz wariant drugi, gdzie badane i przetwarzane są oba pola składowe ramki – w tym przypadku, ta sama sekwencja będzie posiadała dwukrotnie większą rozdzielczość czasową.

III.9.2 Pseudokod wariantu jednoklatkowego

```
// Linia 0
// Pętla zewnętrzna, liczba powtórzeń równa
// szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Linia pozostaje niezmienniona
    P'_{j,0} = P_{j,0}
// Linie od 1 do yr-2
// Pętla zewnętrzna, liczba powtórzeń równa
// szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Pętla wewnętrzna, liczba powtórzeń równa yr-2
    dla i od 0 do yr-2:
        // Obliczanie wartości progowej dla linii i+1
        wp_{i+1} = 2,99 \cdot ((R^{-1}_{i,j} - R^0_{i,j}) (R^1_{i,j} - R^0_{i,j})
            - wk \cdot (R^{-1}_{i,j} - R^1_{i,j})^2)
            + 5,87 \cdot ((G^{-1}_{i,j} - G^0_{i,j}) (G^1_{i,j} - G^0_{i,j})
            - wk \cdot (G^{-1}_{i,j} - G^1_{i,j})^2)
            + 1,14 \cdot ((B^{-1}_{i,j} - B^0_{i,j}) (B^1_{i,j} - B^0_{i,j})
            - wk \cdot (B^{-1}_{i,j} - B^1_{i,j})^2)

        // Porównanie ważonej wartości progowej dla trzech
        // kolejnych punktów z progiem istotności. Wartość
        // progę istotności ustalana jest eksperymentalnie,
        // optymalne wartości progę są różne w zależności
        // od sceny obrazu.
        jeżeli wp_{i-1} + 2wp_i + wp_{i+1} > prog:
            // Linia jest interpolowana tylko dla
            // linii parzystych (odpowiadającym polu górnemu)
            jeżeli i modulo 2 = 0:
                // Właściwa interpolacja punktu
                P'_{i,j} = (P_{i,j-1} + P_{i,j+1})/2
            // Jeżeli linia jest nieparzysta
            inaczej
                // Linia pozostaje niezmienniona
                P'_{i,j} = P_{i,j}
            // Jeżeli wartość progowa nie została przekroczona
            inaczej
                // Linia pozostaje niezmienniona
                P'_{i,j} = P_{i,j}

        // Przesuwanie obliczonych wartości progowych wraz z przesuwaniem
        // się przetwarzanych punktów obrazu. Wartości progowej
        // linii i-1 przypisywana jest analogiczna wartość linii i,
        // której z kolei przypisywana jest wartość linii i+1. Wartość
        // linii i+1 jest rewaluowana w kolejnej iteracji.
        wp_{i-1} = wp_i
        wp_i = wp_{i+1}
    // Linia yr-1
    // Pętla zewnętrzna, liczba powtórzeń równa
    // szerokości ramki w pikselach
    dla j od 0 do xr-1:
        // Linia pozostaje niezmienniona
        P'_{j,yr-1} = P_{j,yr-1}
```

III.9.3 Pseudokod wariantu dwukłatkowego

```
// Linia 0
// Pętla zewnętrzna, liczba powtórzeń równa szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Linia pozostają niezmiennione
    P1'j,0 = Pj,0
    P2'j,0 = Pj,0
// Linie od 1 do yr-2
// Pętla zewnętrzna, liczba powtórzeń równa szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Pętla wewnętrzna, liczba powtórzeń równa yr-2
    dla i od 0 do yr-2:
        // Obliczanie wartości progowej dla linii i+1
        wpi+1 = 2,99 · ((R-1i,j - R0i,j) (R1i,j - R0i,j)
                    - wk · (R-1i,j - R1i,j)2)
                + 5,87 · ((G-1i,j - G0i,j) (G1i,j - G0i,j)
                    - wk · (G-1i,j - G1i,j)2)
                + 1,14 · ((B-1i,j - B0i,j) (B1i,j - B0i,j)
                    - wk · (B-1i,j - B1i,j)2)
        // Porównanie ważonej wartości progowej
        jeśli wpi-1 + 2wpi + wpi+1 > prog:
            // Interpolacja tylko dla linii parzystych
            jeśli i modulo 2 = 0:
                // Właściwa interpolacja punktu
                P1'i,j = (Pi,j-1 + Pi,j+1) / 2
                // Linia pozostaje niezmienniona
                P2'i,j = Pi,j
            // Jeśli linia jest nieparzysta
            inaczej:
                // Linia pozostaje niezmienniona
                P1'i,j = Pi,j
                // Właściwa interpolacja punktu
                P2'i,j = (Pi,j-1 + Pi,j+1) / 2
        // Jeśli wartość progowa nie została przekroczona
        inaczej:
            // Linie pozostają niezmiennione
            P1'i,j = Pi,j
            P2'i,j = Pi,j

// Przesuwanie obliczonych wartości progowych
wpi-1 = wpi
wpi = wpi+1
// Linia yr-1
// Pętla zewnętrzna, liczba powtórzeń równa szerokości ramki w pikselach
dla j od 0 do xr-1:
    // Linie pozostają niezmiennione
    P1'j,yr-1 = Pj,yr-1
    P2'j,yr-1 = Pj,yr-1
```

Porównanie metod usuwania przeplotu

IV.1 Porównanie wydajnościowe

Porównanie wydajnościowe poszczególnych implementacji algorytmów usuwania przeplotu jest czysto syntetycznym wyznacznikiem obciążenia obliczeniowego komputera. Porównanie to polega na pomiarze liczby klatek w obrazie, pomiarze ilości czasu potrzebnego na przetworzenie wszystkich klatek w sekwencji, a następnie – wynikiem będzie średni czas, jaki potrzebuje procesor do obróbki pojedynczej klatki. Czas przetwarzania pojedynczej klatki jest mierzony jako różnica czasu procesora przed oraz po zakończeniu procesu obróbki. Dla zapewnienia jak najbardziej wiernego rzeczywistemu pomiaru czasu przetwarzania, na czas testu wyłączone zostają wszystkie dodatkowe funkcje programu, takie jak odnajdywanie znaczników w obrazie oraz wyświetlanie wynikowego obrazu na ekranie - funkcje te uruchomione są jako oddzielny wątek programu i mogą znacząco wpływać na wynik pomiaru. Wybór wydajności, jako jednego z kryteriów porównywania metod, ma wiele wspólnego z ich zastosowaniem w robotyce – urządzenia czy roboty mobilne bardzo często są ograniczone wydajnościowo, przez co dużym atutem jest fakt niewielkiego obciążania mocy obliczeniowej jednostki sprzężonej z robotem.

Porównanie zostało wykonane na przygotowanej wcześniej w warunkach laboratoryjnych sekwencji obrazów o rozdzielczości 768 na 576 pikseli (było to podyktowane rozdzielczością wyjściową kamery rejestrującej obraz). Obraz został nagrany z prędkością 25 klatek na sekundę i zawierał 3543 klatki (kamera została wcześniej odpowiednio skalibrowana przy pomocy materiałów dołączonych do biblioteki ArToolkit). Każde badanie, poza metodą weaving (patrz: punkt IV.1.1), zostało wykonane dziesięć razy w celu uzyskania wiarygodnej średniej pomiarów, niezależnej od chwilowych spadków wydajności procesu, etc. Nie były odrzucane skrajne wyniki pomiarów, jako że w żadnej z serii prób nie wystąpiły zniekształcające anomalie. Zarejestrowana sekwencja pozwala

ocenić wielorakie aspekty związane z badaniem usuwania przeplotu oraz znajdowania wskaźników w obrazie:

- posiada zarówno segmenty mało i bardzo dynamiczne,
- zawiera trzy wskaźniki o różnym stopniu skomplikowania,
- zakres ruchu markerów w obrazie obejmuje przybliżanie, oddalanie, ruchy poziome oraz pionowe, nachylenia wertykalne oraz horyzontalne, skręcenia w osi płaszczyzny znaczników,
- prezentuje markery na planie bliskim i dalekim.



Rysunek IV.1: Ilustracja scen statycznych (brak smużenia), dynamicznych (duże smużenie)



Rysunek IV.2: Zakres bliskości oraz oddalania w zarejestrowanej sekwencji



Rysunek IV.3: Ilustracja różnych kątów nachylenia znaczników względem kamery

Badanie było przeprowadzane zawsze na tej samej sekwencji wsadowej, posiadającej 3543 klatek, ale ze względu na wymagania czasowe badań, autor przetwarzał sekwencję w czasie w rzeczywistym, tzn. wynikowa liczba klatek to ta, która została przetworzona w czasie trwania oryginalnego obrazu (np. 3543 klatek obrazu wsadowego to około 2:21s odtwarzane z prędkością 25 klatek na sekundę. Wynik liczby przetworzonych klatek, 1111, zgodnie z tabelą IV.2, należy interpretować, że w czasie 141s trwania sekwencji wsadowej, zostało przetworzone 1111 klatek, równomiernie rozłożonych w czasie; dla platformy drugiej czas odtwarzania sekwencji uległ wydłużeniu, dlatego ogólny czas próby przekracza 141s – należy jednak zaznaczyć, że w żaden sposób nie miało to wpływu na wynik). Zabieg ten podyktowany potrzebą ograniczenia czasu prób, gdyż niektóre próby, teoretycznie mogłyby trwać ponad godzinę, a potrzeba wykonania 10 prób na dwóch platformach znacznie rozciągnęłaby proces w czasie. Zabieg ten został ograniczony tylko do porównania wydajnościowego, dwa następne badania przeprowadzane były zawsze na pełnej ilości klatek.

Do testów porównawczych autor wykorzystał dwie, różniące się od siebie parametrami, platformy testowe.

Platforma pierwsza, bardzo wydajna; oparta na procesorze Pentium 4 Prescott z technologią HyperThreading, taktowanym częstotliwością 3 GHz. Posiada zamontowane 2 GB pamięci ram, taktowane efektywną częstotliwością 400 MHz. Dysk twardy to Samsung Spinpoint F3 1000 GB, o prędkości 7200 obrotów/minutę. Całość pracuje pod nadzorem systemu Windows XP Professional SP3.

Platforma druga, platforma mobilna; oparta na procesorze Intel Atom taktowany częstotliwością 1.6 GHz. Posiada zamontowane 2 GB pamięci ram, taktowane efektywną częstotliwością 333 MHz. Dysk twardy to Samsung Spinpoint M6 500 GB, o prędkości 5400 obrotów/minutę. Całość pracuje pod nadzorem systemu Windows XP Home SP3.

Poniżej zaprezentowane są wyniki badań opatrzone komentarzem autora.

IV.1.1 Metoda weaving

Ponieważ zdecydowana większość sekwencji obrazu, pochodząca ze źródeł analogowych, przechowywana jest w sposób identyczny, co efekt końcowy metody weaving - dlatego czas przetwarzania pojedynczej klatki przyjęty został jako zerowy, gdyż nie zachodzi potrzeba wykonywania żadnych obliczeń [31].

Weaving						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	0	-	-	0	-	-
AVG	0	-	-	0	-	-

Tabela IV.1: Tabela pomiarów wydajnościowych metody weaving (źródło: opracowanie własne)

IV.1.2 Metoda blending

Wyniki testów wydajnościowych metody blending przedstawiają ten algorytm jako jeden z mniej wydajnych, trzecie miejsce od końca, zarówno na platformie testowej pierwszej oraz platformie testowej numer dwa; uśrednianie każdego punktu każdej linii jest zajmuje relatywnie bardzo dużo czasu procesora.

Blending						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	1111	126,005	8,817	341	198,879	1,715
2	1100	125,576	8,76	346	197,572	1,751
3	1113	126,67	8,787	340	197,562	1,721
4	1101	124,512	8,843	344	197,521	1,742
5	1099	125,452	8,76	346	199,32	1,736
6	1112	125,643	8,85	336	198,83	1,69
7	1124	125,065	8,987	336	199,624	1,683
8	1121	125,212	8,953	342	198,048	1,727
9	1106	127,401	8,681	346	197,604	1,751
10	1113	126,795	8,778	344	197,109	1,745
AVG	1110	125,833	8,821	336	198,99	1,689

Tabela IV.2: Tabela pomiarów wydajnościowych metody blending (źródło: opracowanie własne)

IV.1.3 Metoda selective blending

W przypadku metody selective blending, możliwa do uzyskania prędkość obróbki obrazu wyrażona w klatkach na sekundę jest dla mocniejszej platformy najmniejsza ze wszystkich, zaś dla platformy słabszej, przedostatnia. Już samo porównywanie każdego punktu ramki bieżącej z analogicznym punktem ramki poprzedzającej w celu znalezienia ruchu mocno rzutuje na prędkość obróbki, jeśli więc dodać konieczność przetworzenia obszarów w których wystąpił ruch algorytmem blending – otrzymujemy najmniej optymalną wydajnościową metodę.

Selective blending						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	589	124,473	4,732	494	384,454	1,285
2	590	124,332	4,745	489	387,327	1,262
3	586	125,414	4,673	501	382,003	1,312
4	597	123,704	4,826	493	385,102	1,28
5	591	123,916	4,769	493	383,586	1,285
6	590	123,922	4,761	488	387,598	1,259
7	586	124,723	4,698	496	383,595	1,293
8	591	125,07	4,725	496	382,452	1,297
9	583	123,952	4,703	491	380,88	1,289
10	595	123,686	4,811	494	387,429	1,275
AVG	589,8	124,319	4,744	499	384,762	1,297

Tabela IV.3: Tabela pomiarów wydajnościowych metody selective blending (źródło: opracowanie własne)

IV.1.4 Metody discard

Metody discard przyniosły niejako zaskoczenie – wydawać by się mogło, że prędkość obróbki tej metody może ustępować tylko algorytmowi quarter, ale finalnie, metoda jest dopiero czwarta pod względem ogólnej klasyfikacji wydajności (wersja jednoklatkowa) oraz szósta (wersja dwuklatkowa, o czym również należy pamiętać w ocenie wydajności tej metody). Stało się tak ze względu na konieczność tworzenia nowej struktury dla ramki wynikowej, jako że jest ona o połowę niższa od ramki wejściowej i to najprawdopodobniej zaważyło na słabszym wyniku.

Discard (wersja jednoklatkowa)						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	51,409	68,918	1437	297,542	4,83
2	3543	51,085	69,355	1451	296,996	4,886
3	3543	51,146	69,272	1440	298,881	4,818
4	3543	51,025	69,437	1423	296,052	4,807
5	3543	50,547	70,093	1457	298,191	4,886
6	3543	51,344	69,005	1431	294,531	4,859
7	3543	51,348	69	1426	294,685	4,839
8	3543	50,844	69,684	1445	294,894	4,9
9	3543	50,907	69,598	1442	297,181	4,852
10	3543	50,654	69,945	1432	297,79	4,809
AVG	3543	51,031	69,428	1451	298,62	4,859

Tabela IV.4: Tabela pomiarów wydajnościowych metody discard, wersja jednoklatkowa (źródło: opracowanie własne)

Discard (wersja dwuklatkowa)						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	103,324	34,29	715	298,568	2,395
2	3543	103,039	34,385	721	297,12	2,427
3	3543	104,386	33,941	718	298,728	2,404
4	3543	102,667	34,51	716	297,727	2,405
5	3543	102,962	34,411	723	299,683	2,413
6	3543	104,468	33,915	721	296,683	2,43
7	3543	102,932	34,421	725	298,543	2,428
8	3543	104,196	34,003	706	300,778	2,347
9	3543	102,693	34,501	713	295,988	2,409
10	3543	104,183	34,007	707	299,869	2,358
AVG	3543	103,485	34,237	710	301,118	2,358

Tabela IV.5: Tabela pomiarów wydajnościowych metody discard, wersja dwuklatkowa (źródło: opracowanie własne)

IV.1.5 Metody double

Wersje tej metody zajęły odpowiednio drugie miejsce w klasyfikacji wydajnościowej (wersja jednoklatkowa) oraz siódme (w przypadku wersji dwuklatkowej). W przypadku tej pierwszej, miejsce jest w pełni zasłużone – algorytm jest bardzo prosty i może wykorzystać istniejącą już strukturę przechowującą ramkę. Inaczej jest w przypadku wersji dwuklatkowej – uzyskana prędkość obróbki w klatkach na sekundę spada o ponad 80%, ze względu na potrzebę przeprowadzenia podwójnej ilości obliczeń oraz konieczność utworzenia struktury drugiej ramki wynikowej.

Double (wersja jednoklatkowa)						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	40,413	87,67	1570	306,335	5,125
2	3543	40,64	87,18	1582	303,804	5,207
3	3543	41,171	86,056	1560	303,706	5,137
4	3543	40,907	86,611	1579	305,189	5,174
5	3543	41,097	86,211	1587	303,992	5,221
6	3543	41,296	85,795	1579	307,602	5,133
7	3543	40,81	86,817	1553	306,214	5,072
8	3543	41,012	86,389	1575	308,242	5,11
9	3543	41,164	86,07	1554	304,37	5,106
10	3543	40,536	87,404	1587	309,169	5,133
AVG	3543	40,905	86,615	1553	306,68	5,064

Tabela IV.6: Tabela pomiarów wydajnościowych metody double, wersja jednoklatkowa (źródło: opracowanie własne)

Double, wersja dwuklatkowa						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	1760	108,788	16,178	362	195,316	1,853
2	1765	108,256	16,304	357	194,136	1,839
3	1776	109,233	16,259	357	195,903	1,822
4	1738	108,214	16,061	357	196,459	1,817
5	1757	107,863	16,289	362	193,876	1,867
6	1781	108,7	16,385	362	194,755	1,859
7	1743	107,838	16,163	358	195,277	1,833
8	1782	108,024	16,496	363	196,563	1,847
9	1763	109,358	16,121	365	197,348	1,85
10	1763	108,542	16,243	360	195,411	1,842
AVG	1762,8	108,482	16,25	366	195,35	1,874

Tabela IV.7: Tabela pomiarów wydajnościowych metody double, wersja dwuklatkowa (źródło: opracowanie własne)

IV.1.6 Metody quarter

Quarter, czy też resize-down, jest metodą obiektywnie najszybszą – pomimo potrzeby tworzenia nowej struktury obrazu wynikowego (ze względu na zmienione rozmiary ramki wynikowej – co w przypadku metody discard zaważyło na miejscu w rankingu wydajnościowym), liczba punktów obrazu do obróbki jest na tyle mała, że algorytm jest ponad dwukrotnie szybszy od następnego.

W wersji dwuklatkowej możemy zaobserwować dwukrotne zmniejszenie się liczby klatek na sekundę przy przetwarzaniu algorytmem quarter – zarówno na pierwszej oraz drugiej platformie testowej.

Quarter, wersja jednoklatkowa						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	16,968	208,805	1034	140,028	7,384
2	3543	17,458	202,944	1019	138,57	7,354
3	3543	16,572	213,794	1025	139,535	7,346
4	3543	17,05	207,801	1023	139,675	7,324
5	3543	16,989	208,547	1025	139,469	7,349
6	3543	16,815	210,705	1026	138,96	7,383
7	3543	17,181	206,216	1029	139,595	7,371
8	3543	17,056	207,727	1036	138,806	7,464
9	3543	17,49	202,573	1021	139,646	7,311
10	3543	16,547	214,117	1023	139,775	7,319
AVG	3543	17,013	208,253	1042	139,54	7,467

Tabela IV.8: Tabela pomiarów wydajnościowych metody quarter, wersja jednoklatkowa (źródło: opracowanie własne)

Quarter, wersja dwuklatkowa						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	34,523	102,627	1034	285,028	3,628
2	3543	34,841	101,691	1037	284,207	3,649
3	3543	35,063	101,047	1042	284,255	3,666
4	3543	34,804	101,799	1046	282,652	3,701
5	3543	35,418	100,034	1040	283,256	3,672
6	3543	34,563	102,508	1045	286,66	3,645
7	3543	35,008	101,205	1041	285,764	3,643
8	3543	34,646	102,263	1029	286,079	3,597
9	3543	34,584	102,446	1035	287,113	3,605
10	3543	34,697	102,113	1049	285,74	3,671
AVG	3543	34,815	101,766	1040	286,609	3,629

Tabela IV.9: Tabela pomiarów wydajnościowych metody quarter, wersja dwuklatkowa (źródło: opracowanie własne)

IV.1.7 Metoda progressive scan

Wydajnościowo, progressive scan, jest bardzo słabą metodą. Zajmuje dziewiąte miejsce w klasyfikacji (dla platformy pierwszej) oraz ostatnie miejsce (dla platformy drugiej). W tym przypadku, o wyniku zaważyła liczba porównań koniecznych do realizacji algorytmu oraz konieczność tworzenia dodatkowej struktury obrazu dla ramki drugiej.

Progressive scan						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	1029	116,777	8,812	380	389,183	0,976
2	1030	116,181	8,865	383	388,131	0,987
3	1037	116,06	8,935	383	389,014	0,985
4	1040	117,103	8,881	375	390,688	0,96
5	1032	117,249	8,802	376	387,188	0,971
6	1039	116,918	8,887	385	391,232	0,984
7	1020	117,016	8,717	381	389,576	0,978
8	1029	116,186	8,856	375	390,065	0,961
9	1024	116,108	8,819	377	389,438	0,968
10	1040	116,24	8,947	383	386,778	0,99
AVG	1032	116,584	8,852	383	390,998	0,98

Tabela IV.9: Tabela pomiarów wydajnościowych metody progressive scan (źródło: opracowanie własne)

IV.1.8 Metoda autorska

Wyniki pomiarów dla metody autorskiej pozwoliły jej zająć miejsce trzecie (w przypadku wersji jednoklatkowej) oraz miejsce piąte (w przypadku wersji dwuklatkowej). Stosunkowo niewielka liczba porównań i ograniczona ilość obliczeń potrzebnych do realizacji tej metody sprawiają, że jest to bardzo szybka metoda (w szczególności druga wersja algorytmu – nie tylko wyprzedziła wszystkie metody z wersjami dwuklatkowymi, ale także kilka metod jednoklatkowych).

Metoda autorska, wersja jednoklatkowa						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	48,771	72,646	678	160	4,238
2	3543	48,73	72,707	677	159,425	4,247
3	3543	48,953	72,376	676	159,96	4,226
4	3543	49,245	71,946	676	158,951	4,253
5	3543	48,925	72,417	680	161,056	4,222
6	3543	48,51	73,036	688	160,32	4,291
7	3543	48,543	72,987	671	161,226	4,162
8	3543	49,237	71,958	685	161,303	4,247
9	3543	48,704	72,746	676	160,033	4,224
10	3543	48,601	72,9	683	159,874	4,272
AVG	3543	48,822	72,57	669	159,856	4,185

Tabela IV.10: Tabela pomiarów wydajnościowych metody autorskiej, wariant jednoklatkowy (źródło: opracowanie własne)

Metoda autorska, wersja dwuklatkowa						
Próba	Platforma testowa 1			Platforma testowa 2		
	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę	Liczba przetworzonych klatek	Czas przetwarzania sekwencji w sekundach	Liczba klatek na sekundę
1	3543	74,836	47,344	575	179,736	3,199
2	3543	74,505	47,554	580	179,803	3,226
3	3543	74,852	47,333	575	180,495	3,186
4	3543	74,511	47,55	572	178,889	3,198
5	3543	74,865	47,325	573	180,06	3,182
6	3543	74,982	47,251	576	180,35	3,194
7	3543	75,115	47,168	567	179,335	3,162
8	3543	74,631	47,474	576	180,209	3,196
9	3543	74,547	47,527	574	179,773	3,193
10	3543	75,438	46,966	572	178,212	3,21
AVG	3543	74,828	47,349	568	180,511	3,147

Tabela IV.11: Tabela pomiarów wydajnościowych metody autorskiej, wariant dwuklatkowy (źródło: opracowanie własne)

IV.1.9 Metoda Gunnara Thalina

Odniesienie pomiaru wydajności metody Gunnara Thalina do metod pozostałych jest niemożliwe, gdyż przeniesienie oryginalnego kodu metody wiąże się ze zmianą sposobu reprezentacji obrazu w filtrach programu VirtualDub i tym samym znaczne zniekształcenie samego algorytmu – w efekcie dałoby to wyniki niemiarodajne.

Próba porównania wydajności poszczególnych algorytmów w samym programie VirtualDub względem metody Gunnara Thalina również nie przyniosła rezultatów (poprzez porównanie czasów przetwarzania sekwencji poszczególnymi metodami), ze względu na brak informacji, jaki narzut czasowy dokładany jest do samego przetwarzania ramki (czas potrzebny na zapisanie klatki, czasy związane z wbudowanymi funkcjami programu VirtualDub, etc.).

Można domniemywać, że algorytm autorski wypadłby w tym porównaniu lepiej, jako że bazuje na uproszczonej wersji tej metody (szczególnie w przypadku wersji dwuklatkowej).

Jedynymi danymi, które badający może tu przytoczyć, jest różnica przetwarzania wersji jednoklatkowej oraz wersji dwuklatkowej algorytmu Gunnara Thalina:

- dla wersji jednoklatkowej, czas przetwarzania wyniósł 241s,
- dla wersji dwuklatkowej, czas przetwarzania wyniósł 540s, czyli około 2,241 razy wolniej.

Odnosząc to do wcześniejszych wyników badań, można wykazać, że różnica pomiędzy wersją jednoklatkową a dwuklatkową metody autorskiej jest mniejsza niż powyższy współczynnik, gdyż wynosi on tylko 1,5327 – różnica czasu wykonania pomiędzy dwoma wariantami metody autorskiej jest prawie półtora razy mniejsza.

IV.2 Porównanie jakościowe

Porównanie jakościowe algorytmów usuwania przeplotu składa się z dwóch części.

Cześć pierwsza jest subiektywną oceną obrazu wynikowego dokonaną przez autora niniejszego opracowania. Oceniający zwracał uwagę w szczególności na następujące elementy:

- rozdzielczość pionowa i pozioma klatki (lub klatek) wynikowej w stosunku do klatki oryginalnej,
- zachowanie rozmiarów / formatu obrazu ramki końcowej względem ramki początkowej,
- istnienie artefaktów, czyli widocznych zniekształceń obrazu na skutek działania algorytmów,
- płynność sekwencji wynikowej.

Druga część jest reprezentacją porównania jakościowego wykrywania znacznika w obrazie.

Do tego celu autor wykorzystał dwa wskaźniki:

- stosunek liczby klatek z wykrytym znacznikiem do ogółu klatek w sekwencji, wartość procentowa,
- arytmetyczną średnią pewności wykrycia znacznika w obrazie liczoną na podstawie wszystkich klatek, w których znacznik został wykryty. Pewność wykrycia jest podawana przez algorytm wyszukujący znaczniki i oznacza stopień prawdopodobieństwa, że w danym obrazie faktycznie został wykryty dany znacznik.

Fakt wyboru odnajdywania znaczników w obrazie, jako elementu związanego z porównaniem wyników poszczególnych algorytmów usuwania przeplotu, związany jest z ich szerokim zastosowaniem w robotyce. Zastosowań jest cała gama – od oznaczania elementów otoczenia, po śledzenie ruchów robotów. Z punktu widzenia większości tych zastosowań, wykrycie znacznika oraz pewność jego wykrycia są bardzo istotnymi

parametrami związanym z posługiwaniem się znacznikami, gdyż od tego może zależeć kluczowa decyzja, jaką podejmie dalszy algorytm programu w danym przypadku – dlatego dużą wartością dodaną jest wykrywanie jak najpełniejszej liczby klatek w obrazie oraz jak największa pewność wykrycia [26] [27] [28] [29].

Wyszukiwanie markerów jest realizowane przez bezpłatną bibliotekę ARToolkit pozwalającą na stosowanie zaawansowanych technik śledzenia znaczników w obrazie. Dla potrzeb niniejszego opracowania wykorzystane zostały tylko metody wykrywające znaczniki w obrazie oraz kilka funkcji pomocniczych, takich jak pomiar czasu działania oraz zwracanie liczby odpowiadającej prawdopodobieństwu wykrycia wskaźnika w obrazie.

Sekwencjami wsadowymi były poszczególne sekwencje obrazów w pełni przetworzone w analogicznych punktach podrozdziału IV.1.

W ramach badań, autor posługiwał się trzema różnymi znacznikami (patrz: rysunek IV.4), o różnym stopniu skomplikowania. Celem było zbadanie, jaki wpływ na poszczególne algorytmy usuwania przeplotu mają wskaźniki, czy niektóre metody będą lepiej działały dla pewnych wskaźników, etc.



Rysunek IV.4: Wskaźniki użyte podczas badań nad algorytmami (źródło: opracowanie własne)

Dodatkowym kryterium pomiarowym będzie użycie funkcji historii śledzenia znaczników w bibliotece ARToolkit. Kiedy ta funkcja jest włączona, algorytm wyszukujący znaczniki w obrazie pamięta ostatnie pozycje znalezionego wskaźnika, powodując znaczny wzrost poprawności wyszukiwania. W przypadku wyłączenia tej funkcji, algorytm znajdowania markerów nie korzysta z historii wyszukiwań, algorytm przez to działa szybciej, ale dzieje się to kosztem skuteczności algorytmu.

Badanie związane z faktem wykrywania znaczników będących tzw. „fałszywymi pozytywami” nie zostały ujęte w niniejszym opracowaniu, ze względu na to, że stanowiły one bardzo sporadyczne przypadki (kilka do kilkunastu takich wykryć na 3000+ klatek przy metodach weaving oraz blending, które bardzo zniekształcają widziany obraz oraz jednostkowe przypadki dla takiej samej skali dla pozostałych metod). Tak mała skala tego problemu wynika z samej charakterystyki markerów – bardzo wyróżniają się od otoczenia i bardzo trudno pomylić z obiektami w tle.

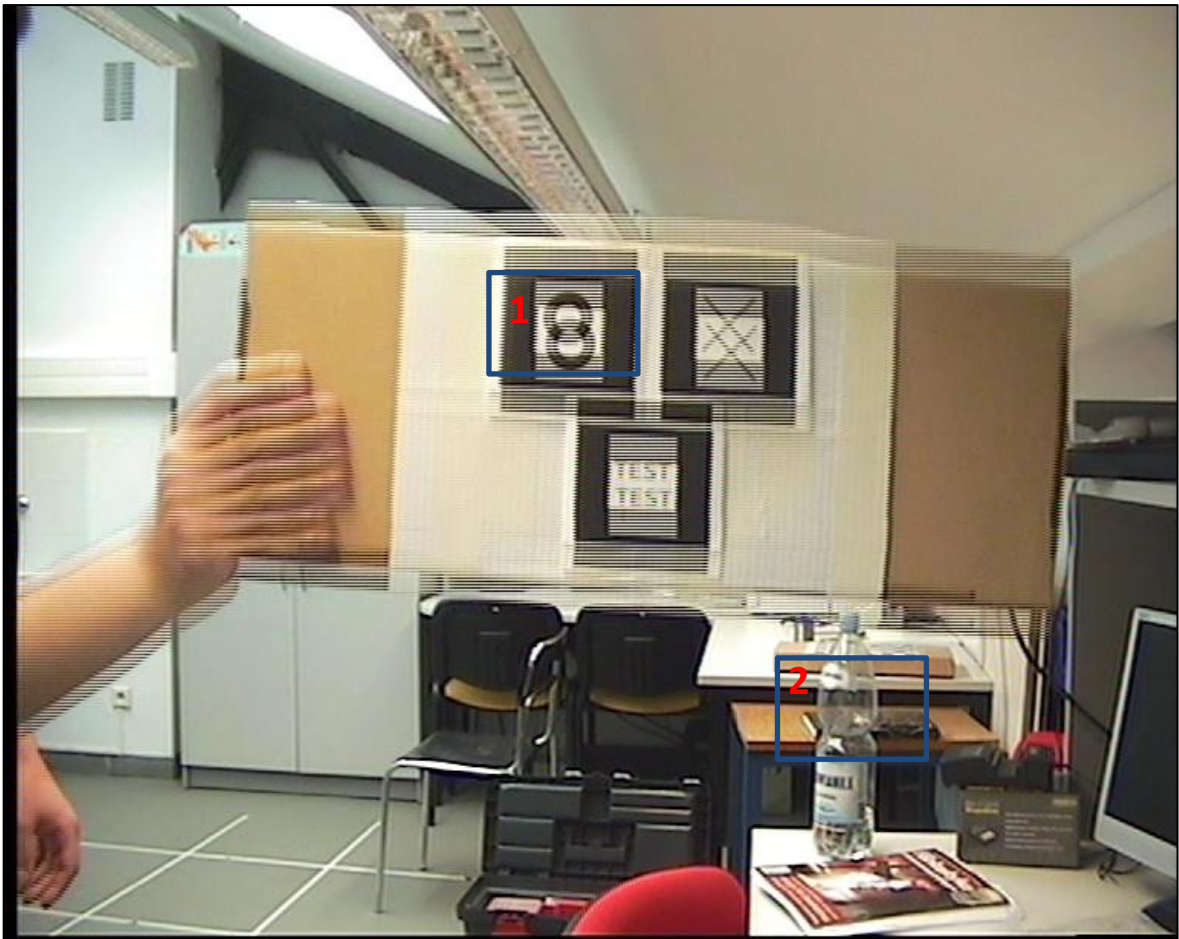
IV.2.1 Metoda weaving

Empirycznie, dla obrazu przetworzonego metodą weaving można wyróżnić dwie jakościowe obserwacje (patrz: rysunek IV.5):

- 1) ramka wynikowa zawiera artefakty w postaci tzw. „grzebienia”, znacznie zaburzającego percepcję jakościową obrazu, szczególnie mocno widoczne na etapach dynamicznych sceny. Jest również dużym problemem w momentach, kiedy znacznik nie znajduje się na wprost kamery, perspektywa w połączeniu z efektem przeplotu daje bardzo złe efekty¹
- 2) obraz ma pełną rozdzielczość pionową, co jest szczególnie dostrzegalne w miejscach, gdzie ruch nie występuje.

Jeśli chodzi o pomiar wykrywalności znaczników w obrazie oraz pewności wykrycia z tym związanej, metoda weaving cechuje się najgorszą ze wszystkich metod wykrywalnością (zmierzonej na poziomie 29,9%; według obserwacji autora, maksymalna wykrywalność dla użytej sekwencji obrazów mogłaby teoretycznie wynieść około 73-75%, wartość nieosiągnięta przez żadną z metod) oraz przeciętną pewnością wykrywania markerów (patrz: tabela IV.12).

¹ 1) lub (1) w tekście odwołują się do odpowiadających im oznaczeń prostokątów w rysunkach ilustrujących efekt końcowy danego algorytmu



Rysunek IV.5: Ramka wynikowa algorytmu weaving (źródło: opracowanie własne)

Weaving												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	36,12	0,9218	31,82	0,8514	36,14	0,7669	28,67	0,8738	20,67	0,7313	25,61	0,6743
2	36,36	0,9216	31,47	0,8521	36,45	0,7662	28,9	0,8746	20,63	0,7311	25,89	0,6737
3	35,99	0,9212	31,79	0,8515	35,41	0,7664	28,86	0,8731	20,58	0,7315	25,77	0,6741
4	36,2	0,9209	31,57	0,8513	35,89	0,7657	28,24	0,8735	20,53	0,7319	25,64	0,6743
5	35,83	0,9213	31,58	0,8515	35,42	0,766	28,54	0,8737	20,68	0,732	25,58	0,6748
6	36,02	0,922	31,63	0,8522	36,49	0,7657	28,69	0,8739	20,79	0,7308	25,73	0,6749
7	36,01	0,9223	31,91	0,8515	36,27	0,7664	28,39	0,8746	20,43	0,7311	25,75	0,6742
8	35,87	0,922	31,5	0,8516	36,43	0,7664	28,51	0,8732	20,71	0,7307	25,64	0,6737
9	36,45	0,9222	31,72	0,8521	36,15	0,7661	28,83	0,8732	20,52	0,7319	25,65	0,6747
10	36,25	0,9223	32,01	0,8506	36,35	0,7654	28,86	0,8746	20,83	0,7318	25,62	0,6741
AVG	36,3	0,9224	32	0,8515	36,13	0,7661	28,71	0,8733	20,64	0,7309	25,76	0,6742
AVG	Wykrywalność [%]		29,92				Pewność		0,8031			

Tabela IV.12: Tabela pomiarów algorytmu weaving (źródło: opracowanie własne)

Objaśnienia tabeli:

O oznacza próby przeprowadzone ze wskaźnikiem z kołem w środku

X - oznacza próby przeprowadzone ze wskaźnikiem ze znakiem X w środku

TEST - oznacza próby przeprowadzone ze wskaźnikiem ze napisem TEST w środku

Wyk. [%] - oznacza wykrywalność, liczoną jako procentowy udział liczby znalezionych znaczników do całkowitej liczbie badanych klatek

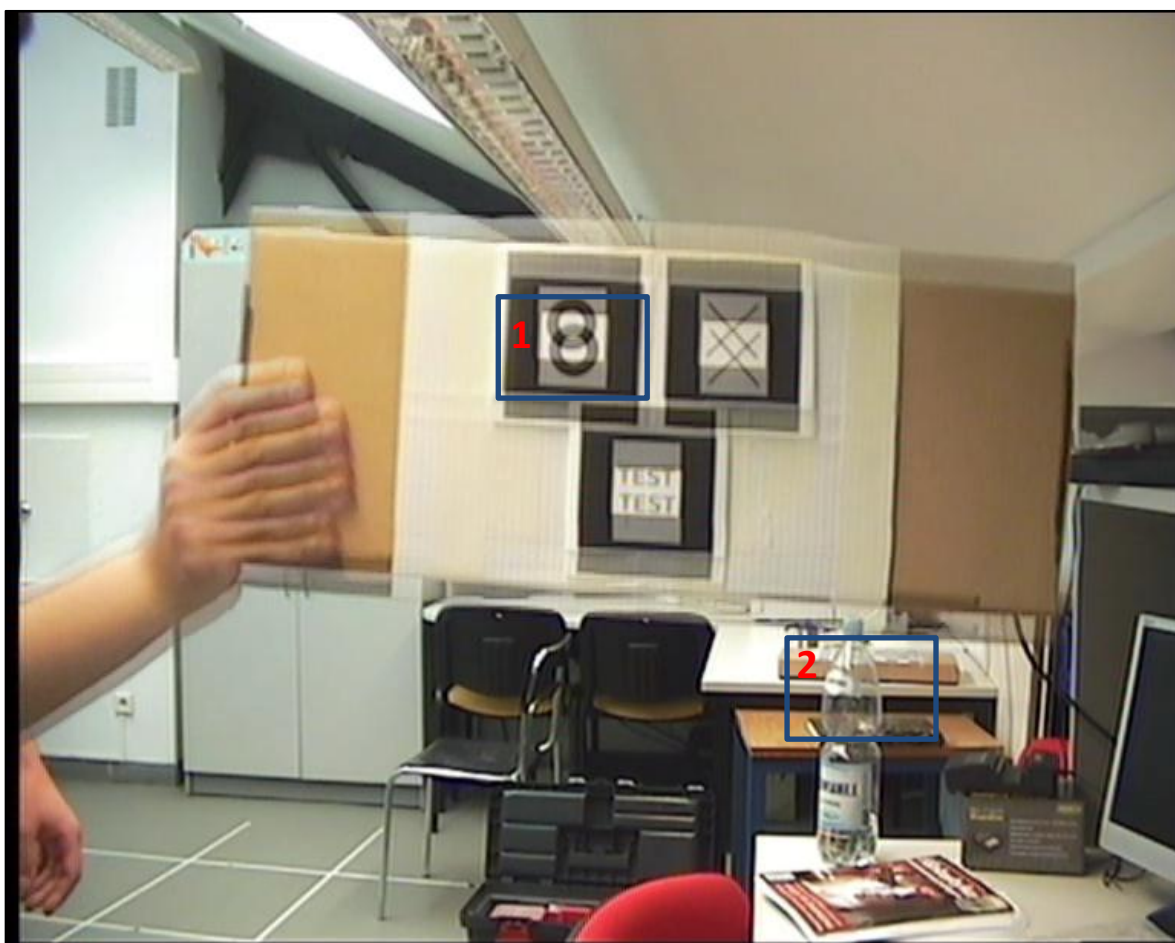
Pwn. - oznacza średnią pewność wykrycia wskaźnika w ramce obrazu, reprezentuje prawdopodobieństwo, że dany wskaźnik jest faktycznie został znaleziony poprawnie w obrazie

IV.2.2 Metoda blending

Efekty działania algorytmu blending pozwalają zauważyć wyraźne pogorszenie jakości obrazu (2), czego przyczyną jest przeprowadzone uśrednianie kolejnych linii; detale są gorzej widoczne, niezależnie od tego czy w danym miejscu występował ruch, czy nie. Dodatkowo, w obrazie końcowym, ramach obszarów, w których nastąpił ruch, występuje bardzo silne smużenie (1), znacznie wpływające na percepcję obrazu – poruszające się przedmioty są praktycznie częściowo kopiowane w pewnej odległości od siebie z połowiczną transparentnością; odległość ta jest zależna od siły i kierunku ruchu. Smużenie

jest szczególnie problematyczne w przypadku, gdy znaczniki nie są ustawione na wprost kamery; w takim przypadku, wykrycie znacznika jest bardzo utrudnione.

Podobnie jak metoda weaving, blending wprowadza bardzo dużo widocznych zniekształceń do obrazu końcowego, co jest przyczyną bardzo słabej wykrywalności znaczników (37,87%, drugi wynik od końca) oraz przeciętnej pewności wykrywania markera (patrz: tabela IV.13).



Rysunek IV.6: Ramka wynikowa algorytmu blending (źródło: opracowanie własne)

Blending												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	45,47	0,9175	40,35	0,8118	46,21	0,7728	37,82	0,8603	24,82	0,726	32,05	0,672
2	45,93	0,9175	40,59	0,8115	46,13	0,7717	38,4	0,8643	24,95	0,7258	32,09	0,6684
3	45,93	0,9208	40,4	0,816	46,05	0,7727	38,25	0,8581	24,68	0,7304	32,23	0,6712
4	45,87	0,9207	40,35	0,8103	46,19	0,7712	37,81	0,8634	24,78	0,7275	32,18	0,6707
5	45,85	0,9181	40,37	0,81	46,32	0,7698	38,18	0,8647	24,83	0,7274	32,04	0,6711
6	45,78	0,9204	40,18	0,811	46,09	0,7764	38,39	0,8651	24,72	0,727	32,33	0,6712
7	45,81	0,9241	40,36	0,8094	45,84	0,7755	38,39	0,8598	24,66	0,7278	32,31	0,6727
8	45,69	0,9212	40,09	0,8137	46,23	0,7717	37,85	0,8612	24,82	0,7311	32,16	0,6701
9	45,19	0,9193	40,38	0,8117	46,42	0,7721	37,97	0,8583	24,72	0,7264	31,96	0,6733
10	45,77	0,92	40,26	0,811	45,95	0,7732	38,06	0,8642	24,78	0,7284	32,18	0,6731
AVG	45,73	0,92	40,33	0,8116	46,14	0,7727	38,11	0,8619	24,78	0,7278	32,15	0,6714
AVG	Wykrywalność [%]			37,87			Pewność			0,7942		

Tabela IV.13: Tabela pomiarów algorytmu blending (źródło: opracowanie własne)

IV.2.3 Metoda selective blending

Efekt wizualny działania algorytmu selective blending jest nieznacznie lepszy od działania metody blending, z tą różnicą, że w razie braku ruchu w sekwencji, obraz nie jest procesowany i dzięki temu nie traci ostrości (2). Niestety, gdy pojawi się w kolejnych ramkach jakikolwiek ruch, „załącza” się algorytm blending i wprowadza znaczne, łatwo dostrzegalne zmiany (1) (patrz: rysunek IV.7). Ten typ zmian bardzo wpływa na wykrywanie markerów, a w szczególności w momentach, gdy wskaźnik nie jest zwrócony prostopadle to kamery.

Mimo teoretycznie bardziej korzystnego algorytmu, selective blending, nie wypada wiele lepiej niż sam algorytm blending – zmiana jest wręcz minimalna, plasując tę metodę na

trzecim od końca miejscu, jeśli chodzi o wykrywalność markerów oraz również na jednej z ostatnich pozycji w przypadku pewności wykrywania (patrz: tabela IV.14).



Rysunek IV.7: Tabela pomiarów algorytmu selective blending (źródło: opracowanie własne)

Selective blending												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	47,81	0,9294	43,17	0,8238	44,61	0,7736	35,42	0,6895	26,68	0,747	35,41	0,6936
2	47,84	0,925	43,24	0,8257	45,08	0,7745	35,13	0,6905	26,64	0,7519	35,37	0,6939
3	47,33	0,9215	43,17	0,822	44,81	0,7735	35,26	0,6899	26,96	0,7496	34,83	0,6877
4	47,99	0,928	43,18	0,8203	44,47	0,7764	34,93	0,6928	26,69	0,7506	34,86	0,6911
5	47,9	0,9264	43,59	0,8223	44,57	0,7769	35,39	0,6929	26,65	0,7526	35,36	0,6886
6	47,1	0,9279	43,22	0,8259	44,58	0,7784	35,23	0,6878	26,68	0,7459	34,92	0,6894
7	47,83	0,9237	43,67	0,8226	44,61	0,7798	34,94	0,6938	26,61	0,7517	35,33	0,6934
8	47,74	0,9295	43,14	0,8214	45,09	0,7782	35,49	0,6901	26,97	0,7516	35,11	0,6909
9	47,64	0,9264	43,43	0,8188	44,46	0,78	35,3	0,6883	26,95	0,7527	35,32	0,6921
10	47,1	0,9273	42,99	0,8194	44,5	0,7769	35,07	0,6883	26,64	0,7472	35,48	0,6942
AVG	47,63	0,9265	43,28	0,8222	44,68	0,7768	35,22	0,6904	26,75	0,7501	35,2	0,6915
AVG	Wykrywalność [%]			38,79			Pewność			0,7763		

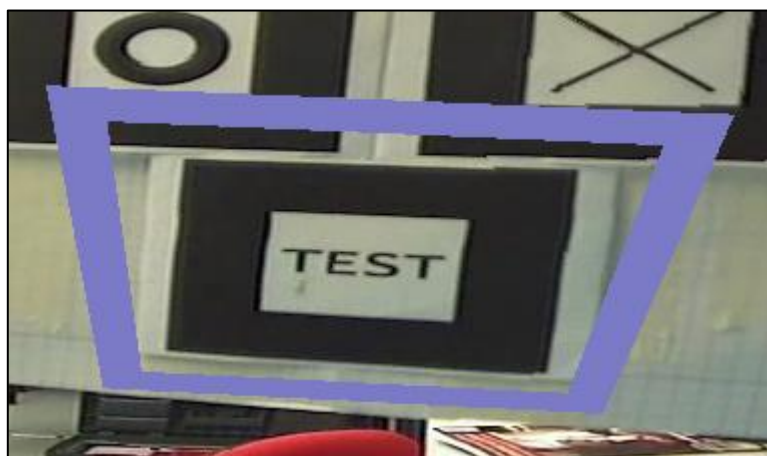
Tabela IV.14: Tabela pomiarów algorytmu selective blending (źródło: opracowanie własne)

IV.2.4 Metody discard

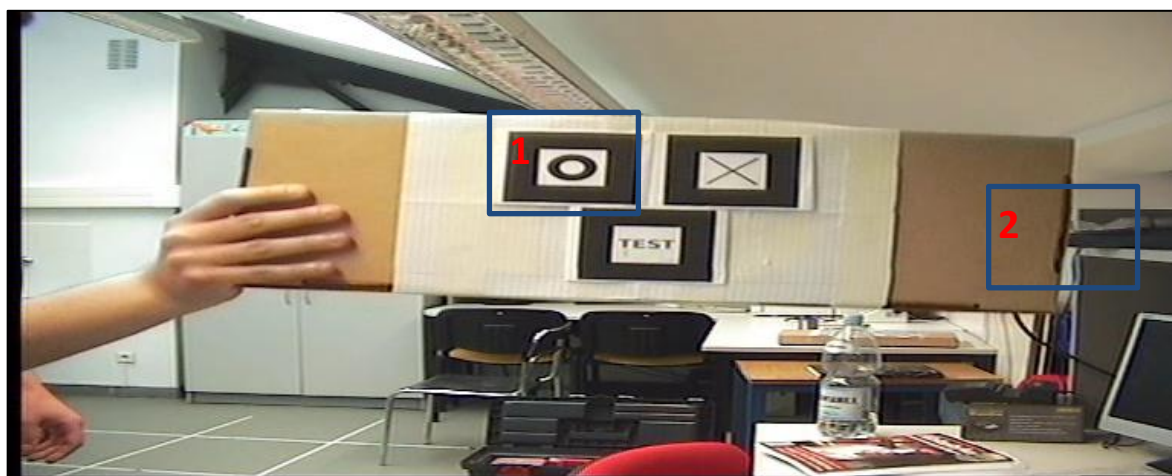
Rodzina metod discard jest bardzo charakterystyczna jeśli chodzi o efekt końcowy obróbki obrazu – ramka (lub też ramki wynikowe) są zmniejszone w pionie o połowę, dając nienaturalny stosunek szerokości do wysokości obrazu (1). Ostrość klatki jest dobra, gdyż algorytm sam w sobie nie wprowadza żadnych zmian do poszczególnych linii. Dodatkowo, jeśli wykorzystamy w końcowej sekwencji oba pola uzyskane tą metodą, dostaniemy obraz o bardzo dobrej płynności, dwukrotnie większej niż normalnie postrzegana (2) oraz (3).

Zarówno jednoklatkowa, jak i dwuklatkowa² wersja metody cechuje się bardzo dobrą wykrywalnością znaczników w obrazie, ale też najmniejszą spośród zarejestrowanych pewność wykrywania markera (patrz: tabela IV.15).

Co więcej, zarejestrowane znaczniki, z racji nieprawidłowych wymiarów, cechują się dużą dowolnością dopasowania, co dobitnie przedstawia rysunek IV.8. Efekt ten jest nieznacznie niwelowany, gdy wskaźniki nie znajdują się na wprost kamery, tylko pod pewnym kątem.

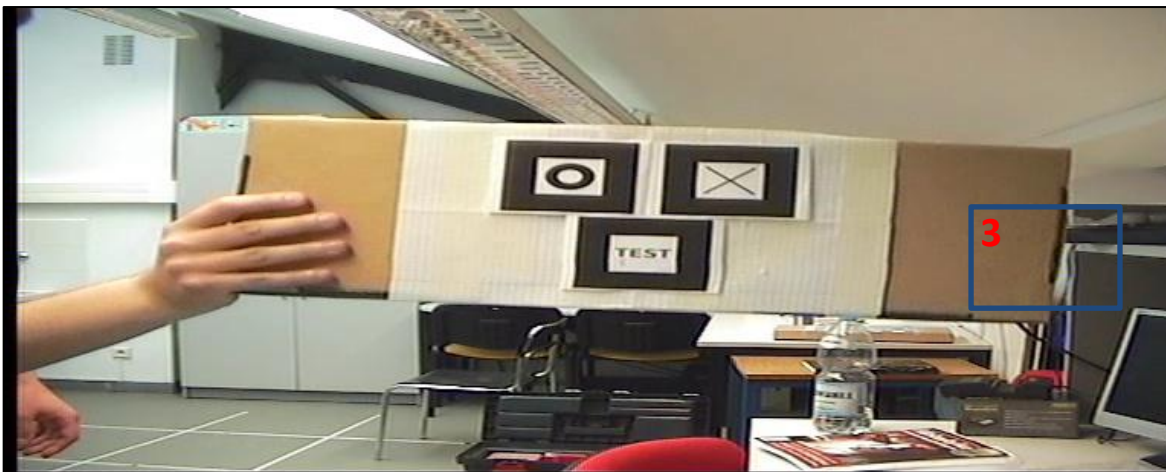


Rysunek IV.8: Błędne dopasowanie znaczników dla rodziny metod discard (źródło: opracowanie własne)



Rysunek IV.9: Ramka wynikowa algorytmu discard, po usunięciu pola dolnego (źródło: opracowanie własne)

² Autor posługuje się pojęciem metody jednoklatkowej oraz metody dwuklatkowej; metoda jednoklatkowa charakteryzuje się tym, że w wyniku daje jedną klatkę końcową, zaś metoda dwuklatkowa daje dwie klatki wynikowe (w obu przypadkach przy jednej klatce wsadowej)



Rysunek IV.10: Ramka wynikowa algorytmu discard, po usunięciu pola górnego (źródło: opracowanie własne)

Discard, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	73,25	0,9401	73,14	0,8215	75,07	0,7946	61,73	0,6913	52,81	0,705	61,87	0,6919
2	73,9	0,9392	71,91	0,8222	75,33	0,7968	62,24	0,6933	52,91	0,7075	61,75	0,6906
3	72,78	0,9367	72,85	0,8216	75,67	0,7908	62,64	0,6917	52,33	0,7054	61,74	0,6934
4	72,57	0,9325	72,92	0,8227	75,32	0,7981	61,68	0,6966	52,48	0,705	61,7	0,6954
5	73,23	0,933	73,15	0,8227	75,39	0,7946	62,39	0,6954	52,92	0,7067	61,73	0,6905
6	72,74	0,9396	71,79	0,822	75,89	0,7963	62,57	0,6916	52,42	0,7067	61,66	0,691
7	73,36	0,9331	71,93	0,8234	75,23	0,7976	62,5	0,6942	52,32	0,7053	61,66	0,6929
8	73,76	0,9355	71,73	0,8264	76,17	0,7945	62,7	0,6913	52,69	0,7042	62,3	0,6951
9	73,85	0,9405	72,38	0,8275	76,42	0,793	61,87	0,6945	52,93	0,7092	62,35	0,693
10	73,92	0,9377	72,98	0,8272	75,6	0,7905	62,59	0,6913	52,55	0,7076	62,56	0,6923
AVG	73,34	0,9368	72,48	0,8237	75,61	0,7947	62,29	0,6931	52,64	0,7063	61,93	0,6926
AVG	Wykrywalność [%]			66,38			Pewność			0,7745		

Tabela IV.15: Tabela pomiarów algorytmu discard, wersja jednoklatkowa (źródło: opracowanie własne)

Discard, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	74,78	0,9373	73,25	0,8273	76,39	0,7942	61,11	0,6908	52,82	0,709	61,66	0,6897
2	74,51	0,9317	73,48	0,8278	76,9	0,7913	61,81	0,6877	53,29	0,7119	61,36	0,6924
3	74,91	0,9327	73,8	0,8278	76,26	0,7973	61,56	0,6903	53,28	0,7117	61,63	0,6869
4	73,64	0,934	73,85	0,8234	77,07	0,792	60,81	0,6898	53,09	0,7098	60,64	0,6891
5	73,49	0,9345	73,37	0,8286	76,84	0,7952	61,32	0,6913	53,44	0,7133	60,87	0,6916
6	73,65	0,9394	73,59	0,828	77,02	0,7907	61,47	0,6904	53,07	0,7116	61,29	0,6891
7	73,79	0,9403	73,2	0,8287	76,25	0,7981	60,94	0,6922	53,44	0,707	60,93	0,6898
8	74,67	0,9382	74,02	0,8248	76,65	0,7924	61,07	0,6935	53,54	0,709	60,61	0,692
9	74,04	0,937	73,09	0,8293	75,66	0,7957	60,73	0,691	52,92	0,7102	61,04	0,6927
10	74,41	0,9355	73,42	0,8295	75,9	0,7951	60,98	0,6882	53,4	0,7075	61,55	0,6886
AVG	74,19	0,9361	73,51	0,8275	76,49	0,7942	61,18	0,6905	53,23	0,7101	61,16	0,6902
AVG	Wykrywalność [%]		66,63				Pewność		0,7739			

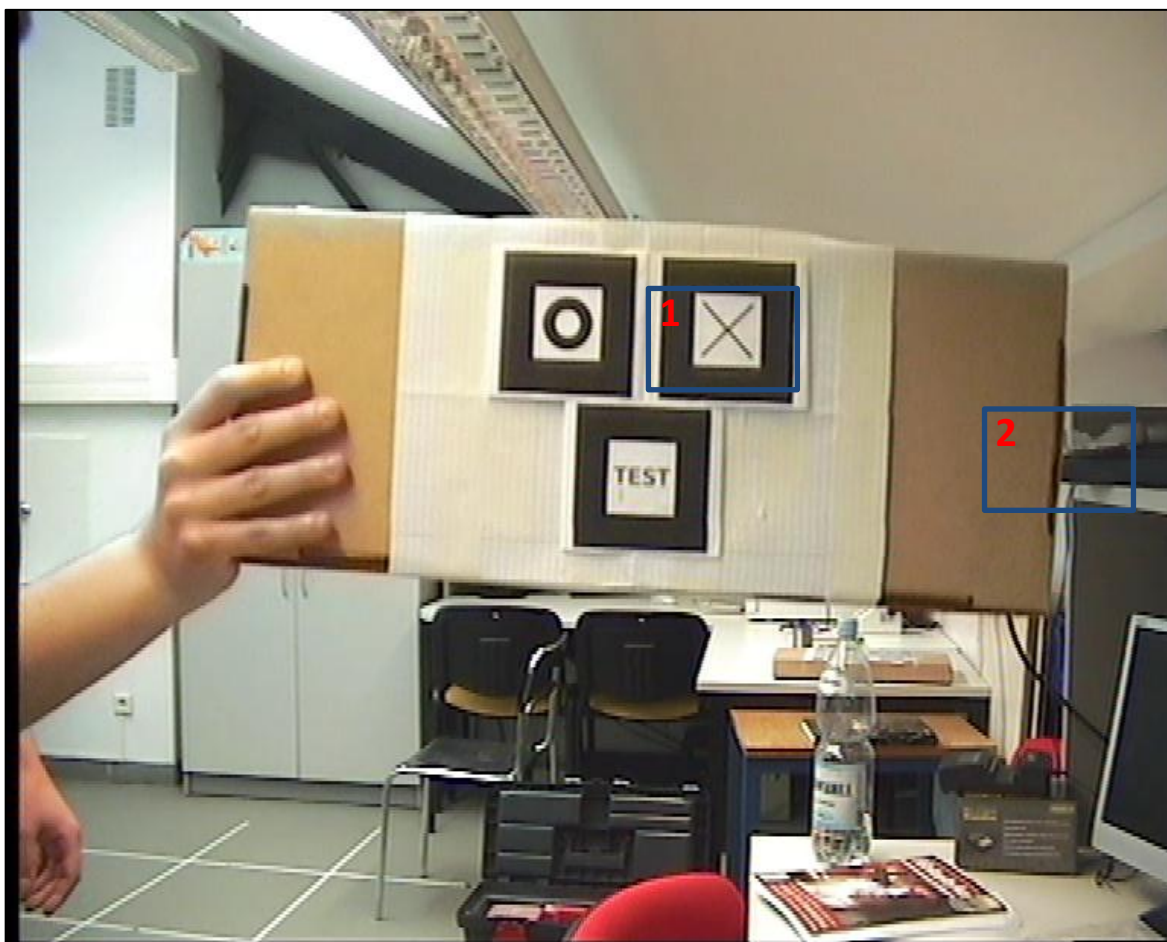
Tabela IV.16: Tabela pomiarów algorytmu discard, wersja dwuklatkowa (źródło: opracowanie własne)

IV.2.5 Metody double

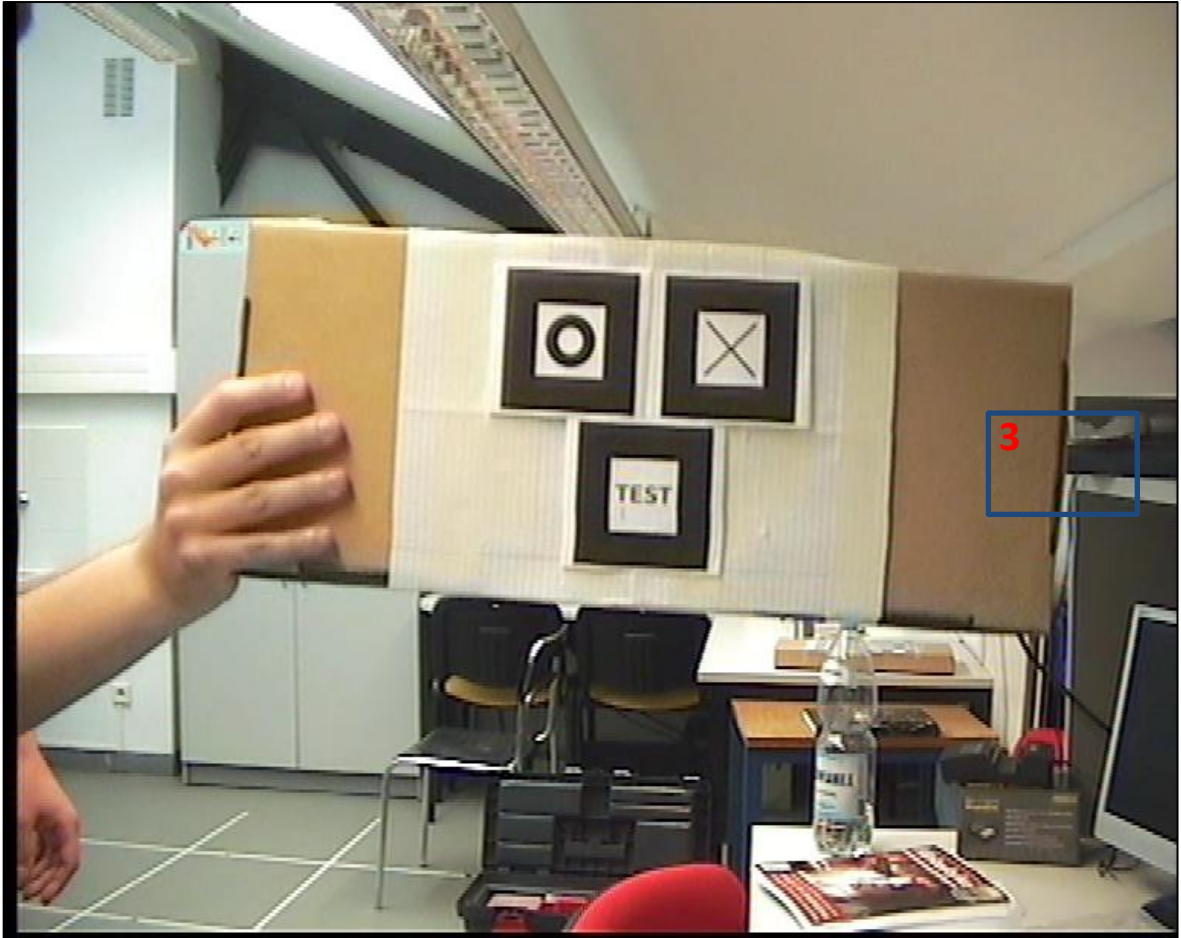
Wizualnie, rodzina algorytmów double prezentuje się przeciętnie, jeśli chodzi o rozdzielczość klatek wynikowych, ponieważ i wersje jednoklatkowe, jak i wersje dwuklatkowe metod są sztucznie powiększane przez dublowanie poszczególnych linii pól. W efekcie, obraz mimo zgodnej z oryginalną wielkości (1) nie jest w stanie oddać szczegółowości obrazu wsadowego. Utrata szczegółowości obrazu staje się szczególnie widoczna w scenach, gdy znacznik jest oddalony od kamery lub znajduje się pod znacznym kątem nachylenia względem niej. Algorytm jest bardzo stały, jeśli chodzi o jakość obrazu – jest ona niezmienną między scenami statycznymi i dynamicznymi, jest w dużej mierze niezależna od kąta nachylenia wskaźnika względem kamery.

Dwuklatkowa wersja algorytmu pozwala na uzyskanie sekwencji obrazu w dwóch kolejnych chwilach w czasie (2) (3), co przekłada się na uzyskanie bardzo dobrej płynności obrazu. Niestety, wprowadza ona (wersja dwuklatkowa) także widoczne migotanie w obrazie, będące wynikiem podwajania linii kolejnych pól – przy zmianie rozszerzonego pola górnego na podwójne pole dolne – wizualnie obraz przeskakuje raz o jeden piksel do góry, raz o jeden piksel w dół. Ma to również wpływ na postrzeganie kąta nachylenia znaczników względem kamery – i w efekcie na nieznaczne obciążenie pomiaru tego nachylenia dodatkową niepewnością.

Pomimo niedostatków jakościowych, obie wersje metod radzą sobie bardzo dobrze w wykrywaniu znaczników (odpowiednio 7 i 6 miejsce, z nieznaczną stratą do najlepszego wyniku) oraz w pewności odnajdywania markerów (7 i 5 lokata wśród wszystkich wyników, patrz: tabele IV.17 oraz IV.18).



Rysunek IV.11: Ramka wynikowa algorytmu double upper (źródło: opracowanie własne)



Rysunek IV.12: Ramka wynikowa algorytmu double lower (źródło: opracowanie własne)

Double, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	72,16	0,9626	73,87	0,8633	74,29	0,8076	61,65	0,91	64,31	0,7228	62,65	0,6835
2	71,06	0,9663	73,91	0,8626	73,92	0,8099	61,46	0,9119	63,48	0,721	62,23	0,6841
3	70,97	0,9626	73,58	0,8649	73,7	0,8107	61,96	0,9131	64,03	0,7171	62,97	0,6816
4	71,17	0,9596	74,68	0,862	74,06	0,8076	62,46	0,9167	64,19	0,7159	63,2	0,6816
5	71,77	0,9628	73,64	0,8695	74,07	0,806	62,36	0,9099	64,14	0,7228	62,65	0,6859
6	71,04	0,9577	74,31	0,8672	73,45	0,8093	62,48	0,9141	64,01	0,7213	62,08	0,6827
7	72,13	0,965	74,07	0,8673	74,47	0,8057	61,67	0,9156	63,25	0,7165	62,54	0,6811
8	71,27	0,9606	73,9	0,8629	74,79	0,8095	62,3	0,9131	63,33	0,7211	62,28	0,6823
9	72,15	0,9656	73,75	0,8656	74,18	0,8125	61,79	0,9163	63,54	0,7179	62,56	0,6836
10	71,66	0,9578	73,62	0,8662	74,3	0,8131	61,78	0,9124	63,27	0,7163	62,05	0,6815
AVG	71,54	0,9621	73,93	0,8652	74,12	0,8092	61,99	0,9133	63,76	0,7193	62,52	0,6828
AVG	Wykrywalność [%]			67,98			Pewność			0,8253		

Tabela IV.17: Tabela pomiarów algorytmu double, wersja jednoklatkowa (źródło: opracowanie własne)

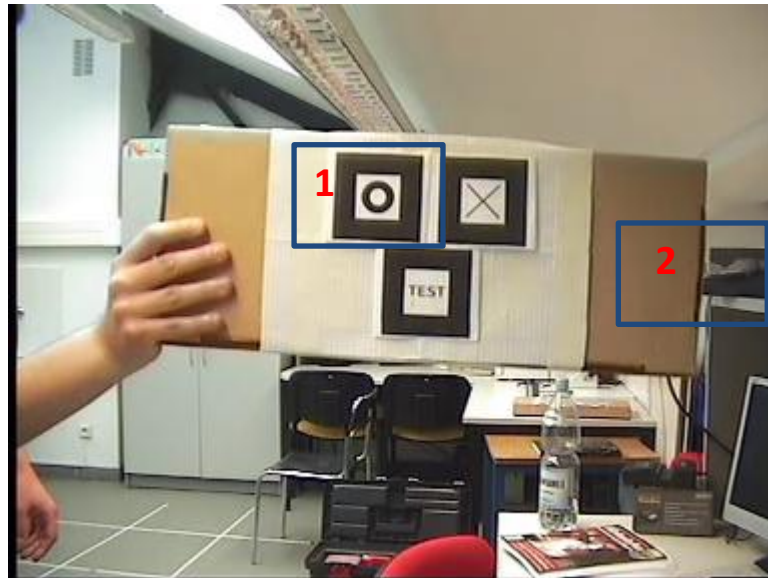
Double, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	72,51	0,9485	74,54	0,857	76,2	0,8054	63,31	0,9163	63,53	0,9445	63,09	0,6775
2	72,57	0,9462	74,38	0,8608	76,63	0,8012	62,9	0,9087	64,17	0,9493	63,61	0,6777
3	72,12	0,9464	74,71	0,8603	76,29	0,7997	63,67	0,9159	63,18	0,9473	62,88	0,6834
4	73,05	0,9463	75,23	0,8551	75,37	0,8008	63,77	0,9093	63,63	0,9485	62,79	0,681
5	72,62	0,9514	74,53	0,8615	76,56	0,8058	63,4	0,9097	63,07	0,9451	63,63	0,6792
6	72,81	0,9466	74,5	0,8626	76,73	0,8031	62,88	0,9123	63,27	0,9468	63,75	0,6788
7	72,11	0,9466	75,39	0,8614	76,53	0,8074	63,14	0,9158	63,72	0,9461	62,94	0,6776
8	72,15	0,9471	74,94	0,8611	75,39	0,8047	63,22	0,9144	64,03	0,9486	63,67	0,6779
9	72,59	0,9537	74,46	0,8598	75,76	0,8017	62,86	0,9141	63,09	0,9532	63,41	0,6813
10	71,98	0,9465	74,34	0,855	75,97	0,8	63,35	0,9081	63,07	0,9461	63,04	0,6778
AVG	72,45	0,9479	74,7	0,8595	76,14	0,803	63,25	0,9125	63,48	0,9476	63,28	0,6792
AVG	Wykrywalność [%]		68,88				Pewność		0,8583			

Tabela IV.18: Tabela pomiarów algorytmu double, wersja dwuklatkowa (źródło: opracowanie własne)

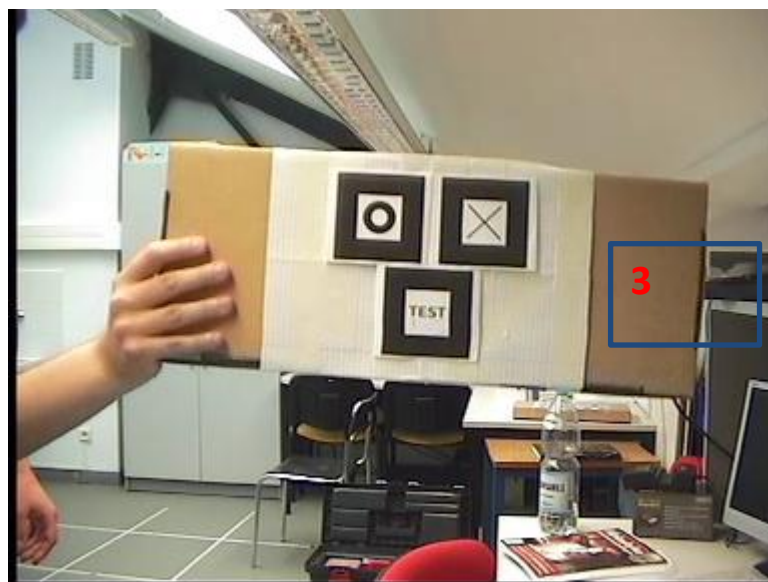
IV.2.6 Metody quarter

Rodzina metod quarter jest równie charakterystyczna, co rodzina metod discard – ze względu na dwukrotnie mniejszą rozdzielczość obrazu końcowego, pionową i poziomą (1). Obraz wydaje się być ostry, ale to jedynie efekt zmniejszenia rozdzielczości; w rzeczywistości ramka wynikowa bardzo traci na szczegółowości. Efekt ten jest powielany w sytuacji, gdy znacznik dodatkowo jest oddalony od kamery i znacznie utrudnia to jego wykrycie. W przypadku wariantu dwuklatkowego algorytmu, sekwencja końcowa zyskuje na płynności, gdyż rozdzielczość czasowa jest dwa razy większa (2) (3).

Wszystkie warianty metody quarter wykazują się raczej przeciętną wykrywalnością znaczników oraz posiadają najniższy spośród innych metod współczynnik pewności wykrywania markera (patrz: tabela IV.19 oraz IV.20).



Rysunek IV.13: Ramka wynikowa algorytmu quarter pochodząca z pola górnego (źródło: opracowanie własne)



Rysunek IV.14: Ramka wynikowa algorytmu quarter pochodząca z pola dolnego (źródło: opracowanie własne)

Quarter, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	72,56	0,9212	70,51	0,8012	74,77	0,7818	62,09	0,8136	37,79	0,6774	50,46	0,6297
2	72,99	0,9209	70,75	0,7993	75,37	0,7824	61,46	0,8188	37,76	0,6792	50,54	0,6271
3	72,43	0,9224	70,9	0,8016	74,9	0,7813	62,2	0,8137	37,93	0,6775	50,94	0,6249
4	72,14	0,919	70,68	0,8002	74,95	0,7887	61,53	0,8174	37,76	0,682	51,46	0,6272
5	72,79	0,9168	70,91	0,7967	75,5	0,7847	61,58	0,8169	37,79	0,6791	50,88	0,63
6	72,13	0,9234	70,86	0,8021	75,34	0,7853	62,07	0,817	37,72	0,6818	51,06	0,6304
7	72,5	0,9191	70,55	0,7995	75	0,7876	62,04	0,813	38,02	0,6771	51,08	0,6277
8	72,29	0,9171	71,37	0,801	75,26	0,7874	62,21	0,8192	38,17	0,6788	50,56	0,6256
9	72,79	0,9156	71,14	0,7992	75,27	0,7841	61,53	0,8123	37,77	0,6813	50,67	0,6255
10	72,01	0,9195	70,8	0,796	75,17	0,7864	62,23	0,815	38,19	0,6777	50,67	0,6302
AVG	72,46	0,9195	70,85	0,7997	75,15	0,785	61,89	0,8157	37,89	0,6792	50,83	0,6278
AVG	Wykrywalność [%]			61,51			Pewność			0,7712		

Tabela IV.19: Tabela pomiarów algorytmu double, wersja jednoklatkowa (źródło: opracowanie własne)

Quarter, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	75,51	0,9311	72,65	0,8104	76,93	0,788	62,26	0,8192	37,65	0,6827	50,92	0,6273
2	75,37	0,9257	72,81	0,8063	77,42	0,7919	62,2	0,8145	37,73	0,6802	51	0,625
3	75,27	0,9265	72,78	0,8111	77,59	0,7904	62,35	0,817	38,23	0,679	50,89	0,6265
4	75,19	0,9307	73,22	0,8068	77,95	0,792	62,03	0,8138	37,85	0,6822	51,28	0,6283
5	75,23	0,931	73,09	0,8073	77,82	0,7936	61,65	0,8187	37,96	0,6796	51,34	0,6246
6	74,95	0,925	73,49	0,8062	77,83	0,7902	62,36	0,8156	37,5	0,6793	50,58	0,6272
7	75,66	0,9302	72,63	0,8075	77,68	0,7937	61,49	0,8144	37,52	0,678	51,09	0,6299
8	75,17	0,9292	73,18	0,8083	77,47	0,7901	61,37	0,8139	37,9	0,6787	50,9	0,627
9	75,5	0,9274	72,15	0,8079	77,47	0,7902	62,05	0,815	37,73	0,6773	50,65	0,6288
10	74,56	0,929	73,49	0,8069	77,93	0,7901	62,35	0,8165	37,65	0,681	50,84	0,626
AVG	75,24	0,9286	72,95	0,8079	77,61	0,791	62,01	0,8159	37,77	0,6798	50,95	0,6271
AVG	Wykrywalność [%]		62,76				Pewność		0,7751			

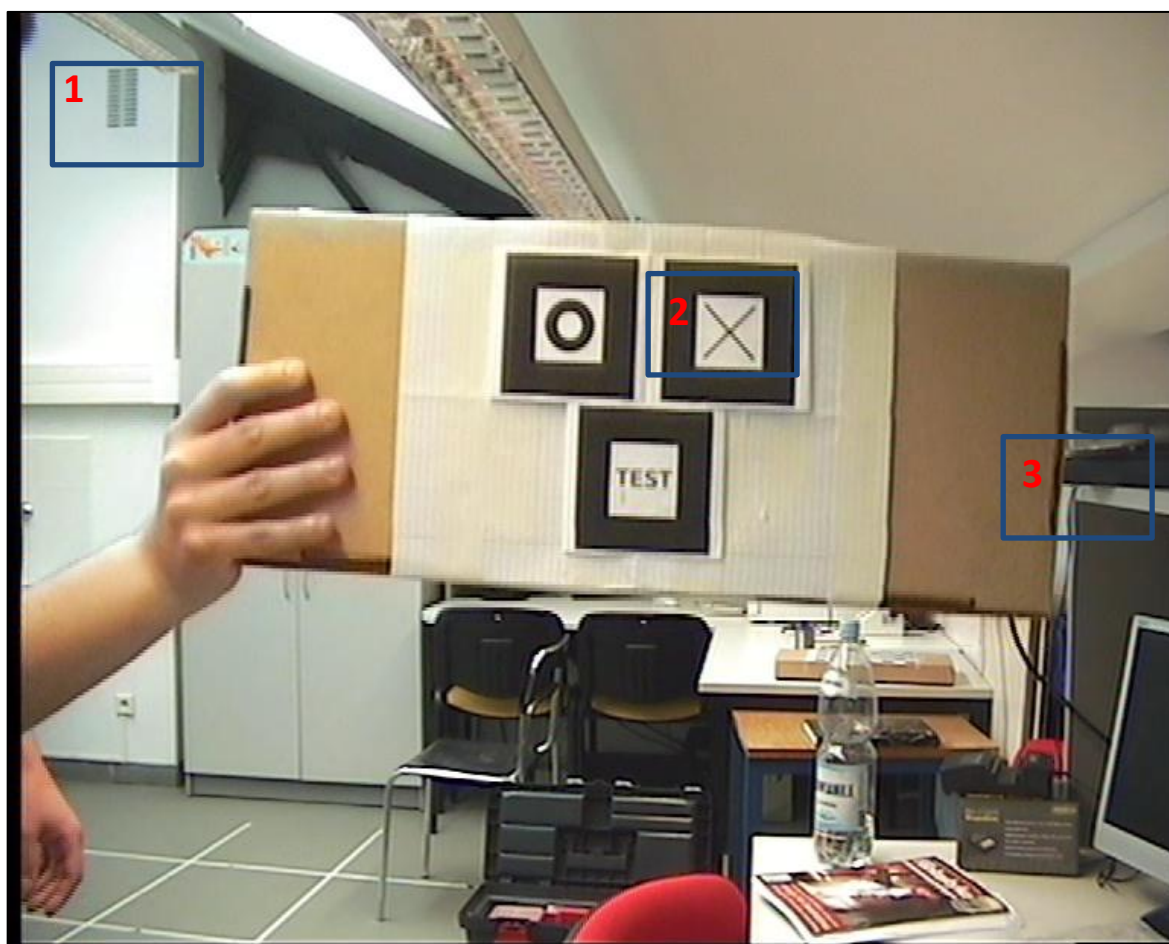
Tabela IV.20: Tabela pomiarów algorytmu double, wersja jednoklatkowa (źródło: opracowanie własne)

IV.2.7 Metoda progressive scan

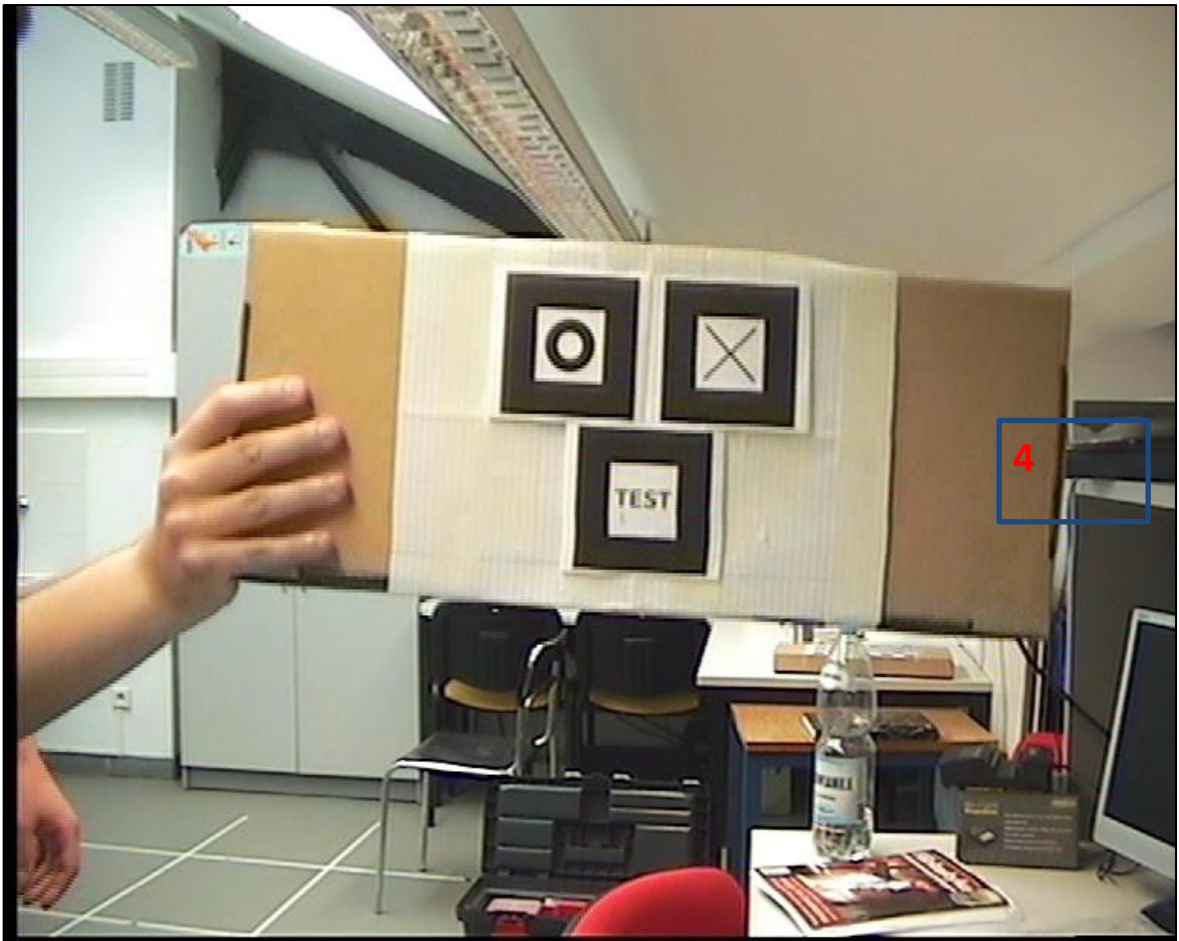
Wizualnie, metoda progressive scan, jest krok do przodu przed metodą double w wersji dwuklatkowej – posiada wszystkie jej zalety oraz częściowo eliminuje jej poważną wadę – utratę rozdzielczości pionowej ramki. Obraz zachowuje prawidłowy stosunek rozmiarów (2) oraz jest dalej bardzo płynny (3) (4). Co więcej, dzięki stosowaniu przetwarzania typu weaving w spokojnych obszarach, scena zachowuje pełną rozdzielczość pionową klatki wejściowej, tam, gdzie ruch nie wystąpił (1). Algorytm ten nie eliminuje wszystkich niedostatków metody doubling; dynamiczne miejsca w obrazie są w gorszej rozdzielczości (2) oraz wykazują tendencję do migotania.

Pod względem wykrywania markerów, algorytm sprawuje się bardzo dobrze, zajmując piątą lokatę w ogólnym zestawieniu, osiągając bardzo zbliżony wynik do najlepszego. Jeśli

chodzi o pewność wykrycia znacznika, metoda plasuje się na drugim w kolejności miejscu, o ułamek punktu procentowego gorzej niż wariant metody autoskiej (patrz: tabela IV.21).



Rysunek IV.15: Ramka wynikowa algorytmu progressive scan powstała na podstawie pola górnego (źródło: opracowanie własne)



Rysunek IV.16: Ramka wynikowa algorytmu progressive scan powstała na podstawie pola dolnego (źródło: opracowanie własne)

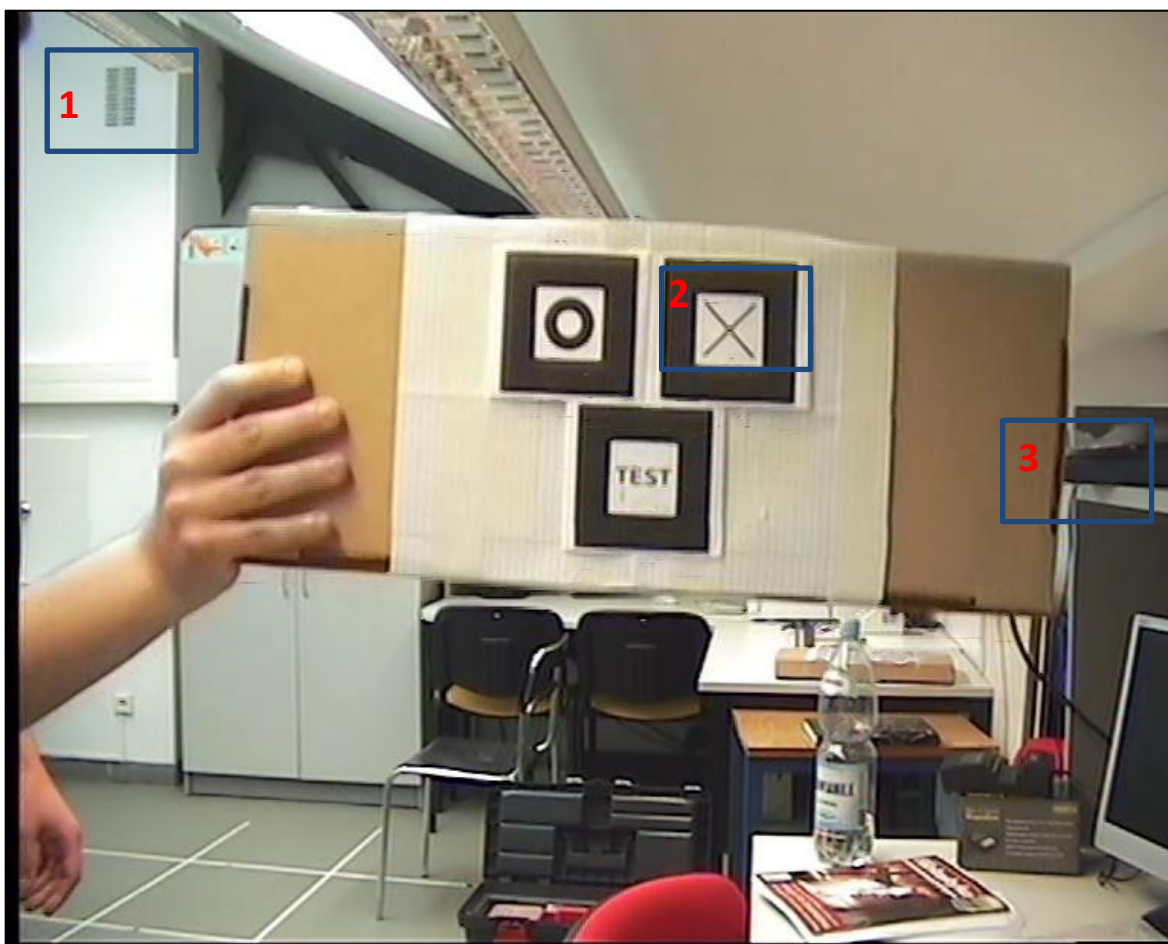
Double, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	73,71	0,9602	76,7	0,8583	76,03	0,8094	63,26	0,9092	63,43	0,9566	62,99	0,6823
2	72,73	0,9573	75,94	0,8596	76,04	0,8055	62,67	0,9077	63,99	0,961	62,65	0,6788
3	72,29	0,9623	75,36	0,8572	75,38	0,8072	63,28	0,9114	62,84	0,9606	63,37	0,6832
4	72,69	0,9617	75,88	0,857	75,53	0,81	63,09	0,9068	63	0,9572	62,74	0,6808
5	73,19	0,9638	75,81	0,8603	76,01	0,8026	62,46	0,9105	62,87	0,9642	62,9	0,6814
6	73,3	0,9588	75,48	0,8541	75,85	0,8068	63,56	0,9121	63,85	0,9573	62,7	0,6805
7	73,08	0,956	75,59	0,8621	75,43	0,8074	62,56	0,9107	63,03	0,9566	63,55	0,6809
8	73,17	0,9567	75,91	0,8545	75,92	0,8047	63,13	0,9135	63,54	0,9566	62,43	0,6791
9	72,73	0,9647	75,9	0,8583	75,88	0,8085	63,37	0,9066	62,94	0,962	63,48	0,6844
10	73,12	0,96	75,72	0,857	75,39	0,81	62,71	0,9127	63,99	0,9555	62,61	0,6804
AVG	73	0,9602	75,83	0,8578	75,75	0,8072	63,01	0,9101	63,35	0,9588	62,94	0,6812
AVG	Wykrywalność [%]			68,98			Pewność			0,8626		

Tabela IV.21: Tabela pomiarów algorytmu double, wersja jednoklatkowa (źródło: opracowanie własne)

IV.2.8 Metoda autorska

Metoda autorska, pod względem graficznym, prezentuje się bardzo dobrze: ma zachowaną rozdzielczość pionową ramki wsadowej, przez co obraz wydaje się być ostry i dobrze czytelny (1) oraz ma prawidłowe stosunki rozmiarów krawędzi. Nie podziela też wad algorytmów opisanych wcześniej – obszary będące w ruchu nie tracą na jakości (2) oraz nie migocą (brak migotania dotyczy wersji dwuklatkowej, w wersji jednoklatkowej taki problem nie występuje w ogóle). W przypadku, gdy na wyjściu algorytmu otrzymujemy dwie klatki – efektywnie zwiększa się rozdzielczość czasowa całej sekwencji (3) (4). Metoda bardzo dobrze sobie radzi niezależnie od oddalenia markera do kamery oraz jego nachylenia względem niej.

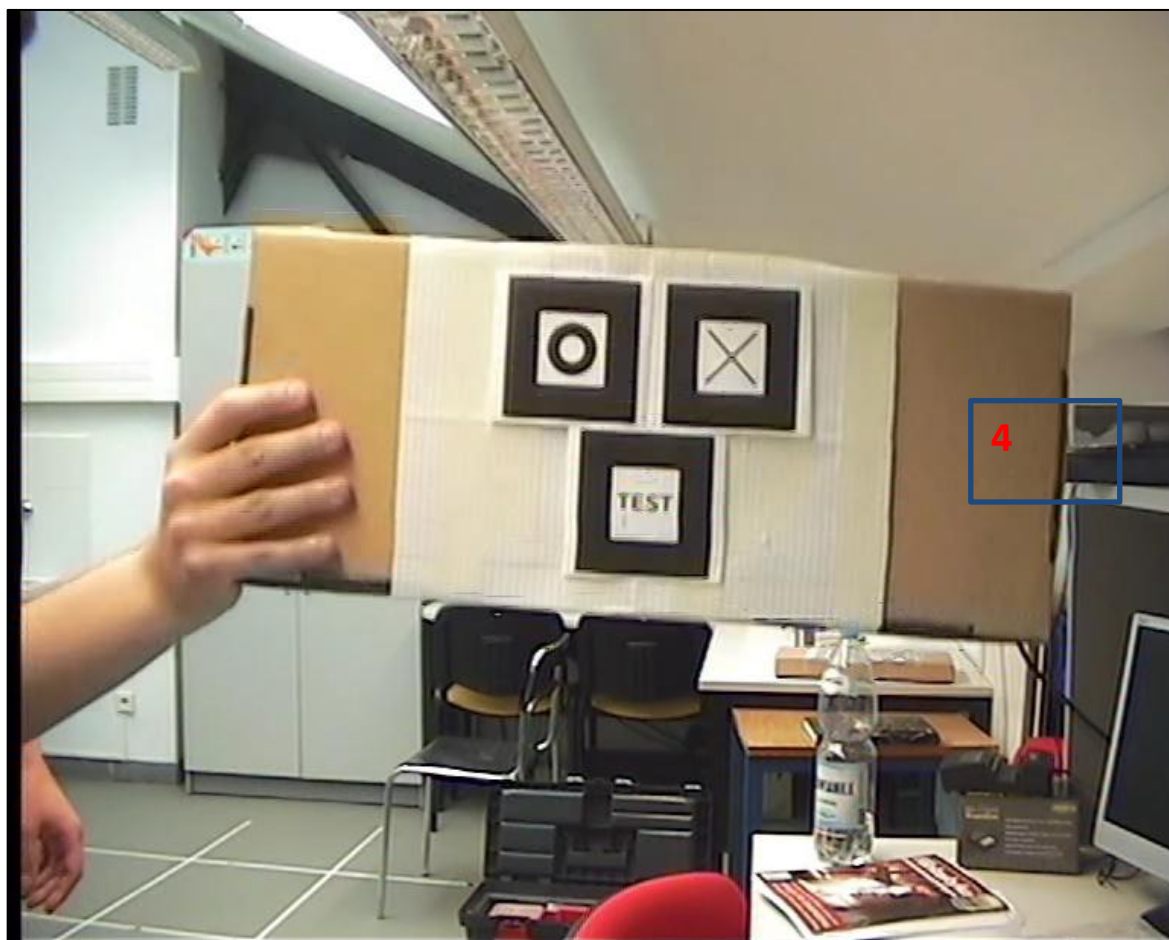
Oba warianty metody wykazują ponadprzeciętną wykrywalność znaczników (drugie miejsce dla wersji jednoklatkowej – pierwsze wśród metod niezmienionej liczbie klatek na sekundę oraz najlepszy wynik ogólnie dla wariantu dwuklatkowego). W przypadku pewności wyszukiwania markera w obrazie, obie metody zajmują bardzo dobre miejsca w rankingu z wynikiem średnim ponad 0,86 (patrz: tabele IV.22 oraz IV.23).



Rysunek IV.17: Ramka wynikowa algorytmu autorskiego powstała na bazie pola górnego (źródło: opracowanie własne)

Należy także nadmienić, że metoda autorska (a wraz z nią metoda Gunnara Thalina) miała najlepszą wykrywalność znaczników w dali – wraz z malejącą liczbą pikseli, metody typu weaving, blending, doubling, discard, a w szczególności quarter miały bardzo małą wykrywalność, kiedy plansza ze wskaźnikami była najbardziej oddalona od kamery. Z racji braku utraty rozdzielczości pionowej, algorytm proponowany przez autora bardzo dobrze

sobie poradził z wykrywaniem optycznie małych wskaźników – co znacznie rozszerza stosowalność tej metody względem pozostałych. Obserwacja ta nie wynika wprost z przedstawionych w poszczególnych tabelach wyników, ale została empirycznie potwierdzona przez badającego w czasie poszczególnych prób jakościowych.



Rysunek IV.18: Ramka wynikowa algorytmu autorskiego powstała na bazie pola dolnego (źródło: opracowanie własne)

Metoda autorska, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	73,95	0,956	77,05	0,8458	77,43	0,7951	63,07	0,9162	60,92	0,9524	62,84	0,6899
2	73,57	0,9556	76,68	0,8415	78,46	0,7939	63,98	0,9112	61,39	0,9549	63,36	0,6903
3	73,28	0,9536	76,09	0,8408	77,73	0,7904	63,74	0,9091	61,37	0,9605	62,91	0,6843
4	72,68	0,953	76,01	0,8418	77,55	0,7905	63,87	0,9097	61,39	0,952	62,79	0,6846
5	73,54	0,9551	77,04	0,8455	78,53	0,7968	63,11	0,9089	61,8	0,9542	63,08	0,6861
6	73,34	0,9575	77,23	0,8486	77,82	0,7975	63,8	0,91	61,38	0,9563	63,67	0,6905
7	72,87	0,9515	76,61	0,8418	77,49	0,7975	63,57	0,9127	60,78	0,9549	62,99	0,6847
8	73,89	0,9563	76,72	0,8422	77,83	0,7933	63,54	0,9104	61,4	0,9597	63,65	0,6839
9	73,54	0,9579	76,91	0,8451	77,84	0,7913	63,53	0,9143	61,66	0,9552	63,68	0,6855
10	72,78	0,9525	76,41	0,8419	78,26	0,7955	63	0,9112	60,94	0,9554	62,87	0,6861
AVG	73,34	0,9549	76,68	0,8435	77,89	0,7942	63,52	0,9114	61,3	0,9556	63,18	0,6866
AVG	Wykrywalność [%]			69,32			Pewność			0,8577		

Tabela IV.22: Tabela pomiarów metody autorskiej, wersja jednoklatkowa (źródło: opracowanie własne)

Metoda autorska, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	73,86	0,9647	76,72	0,8668	77,45	0,8119	63,1	0,9123	61,82	0,958	63,51	0,6867
2	74,18	0,9578	76,74	0,8632	78,63	0,8132	63,12	0,9139	61,43	0,9615	63,2	0,6813
3	73,57	0,9625	77,45	0,8667	78,33	0,813	63,98	0,9111	61,15	0,9609	63,25	0,6834
4	73,17	0,9585	76,98	0,862	78,74	0,8111	62,86	0,9122	61,85	0,9623	63,5	0,6818
5	73,23	0,9587	76,9	0,8691	78,63	0,8123	62,77	0,9101	61,82	0,9667	63,24	0,6838
6	73,23	0,965	76,34	0,8637	78,08	0,8084	62,84	0,9168	61,02	0,9586	63,94	0,6861
7	72,79	0,961	77	0,8645	78,74	0,8129	63,09	0,9147	61,81	0,9628	63,21	0,6862
8	73,41	0,9657	77,04	0,8659	77,53	0,8068	63,8	0,9138	60,99	0,9606	63,03	0,6815
9	74,23	0,9665	77,56	0,8611	77,45	0,8122	63,26	0,9081	61,72	0,959	63,73	0,6816
10	74,02	0,9572	77,27	0,8646	77,41	0,8121	63,02	0,9105	60,92	0,959	63,53	0,6844
AVG	73,57	0,9618	77	0,8648	78,1	0,8114	63,18	0,9124	61,45	0,9609	63,41	0,6837
AVG	Wykrywalność [%]			69,45			Pewność			0,8658		

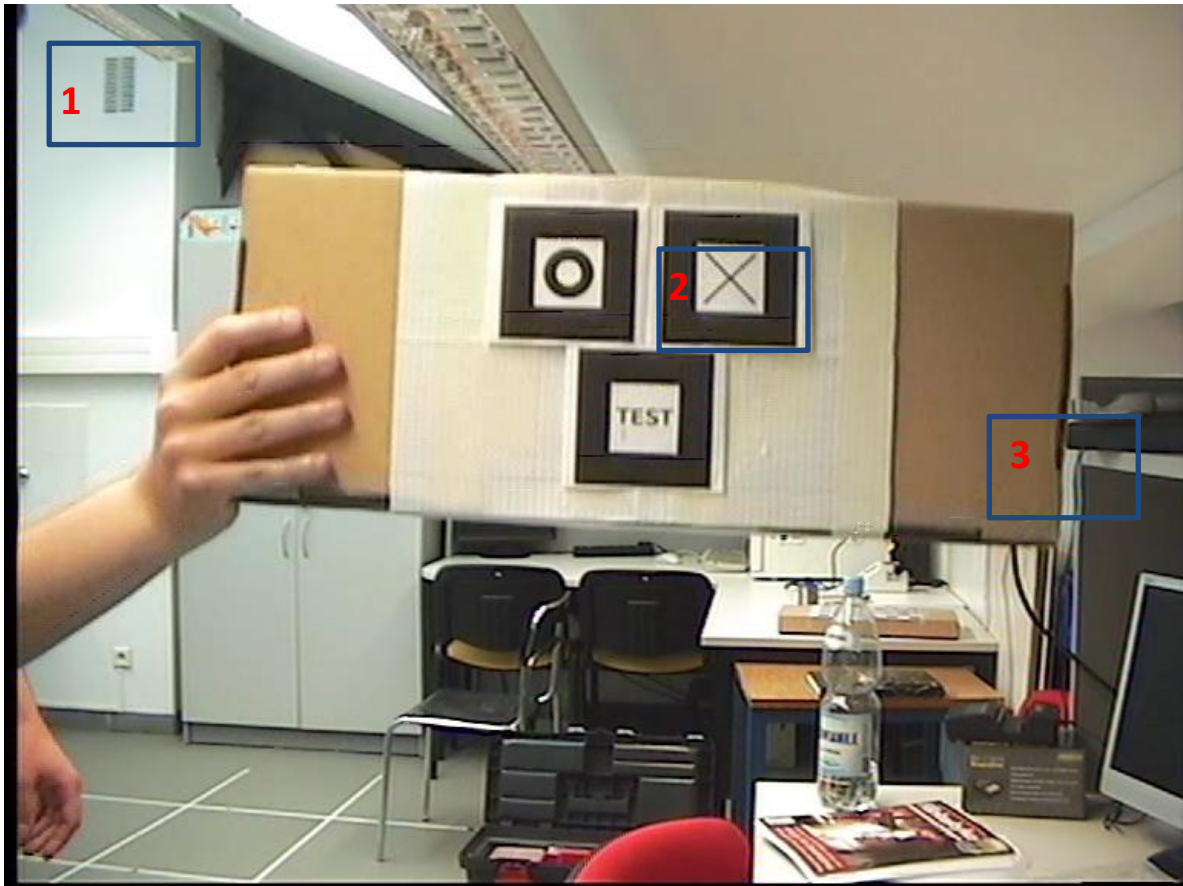
Tabela IV.23: Tabela pomiarów algorytmu autorskiego, wersja dwuklatkowa (źródło: opracowanie własne)

IV.2.9 Metoda Gunnara Thalina

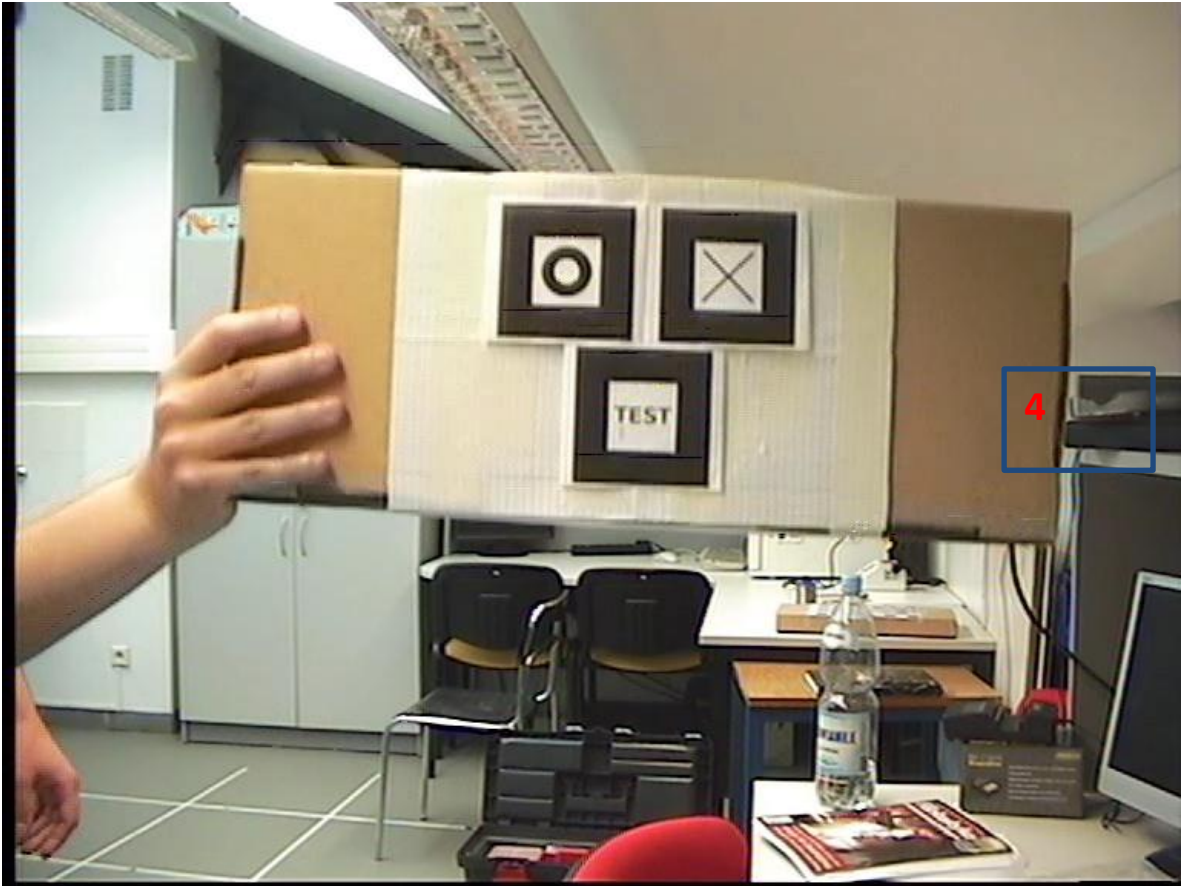
Wizualnie, efekty zastosowania metody Gunnara Thalina są niemal identyczne do metody autorskiej, przez co wnioski wypływające z obserwacji jakości tej metody są tożsame:

- obraz jest ostry, nie utracił detali, zarówno obszary statyczne (1) i dynamiczne (2)
- obraz nie wykazuje tendencji do migotania,
- w wersji dwuklatkowej jest płynny ze względu na zwiększenie częstotliwości wyświetlania klatek (3) (4).

Wyniki obu wariantów metod są spójne do wyników obserwowanych przy badaniu metody autorskiej, same pomiary oraz wyciągnięte z nich średnie pomiarów są zbliżone.



Rysunek IV.19: Ramka wynikowa algorytmu Gunnara Thalina powstała na bazie pola górnego (źródło: opracowanie własne)



Rysunek IV.20: Ramka wynikowa algorytmu Gunnara Thalina powstała na bazie pola dolnego (źródło: opracowanie własne)

Metoda Gunnara Thalina, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	73,62	0,9596	76,43	0,8452	78,12	0,7938	63,58	0,9173	60,88	0,9541	62,87	0,6872
2	73,35	0,9541	76,71	0,8477	77,05	0,7972	63,66	0,9196	61,17	0,9556	62,36	0,6903
3	73,48	0,9602	77,61	0,8468	77,91	0,7929	63,29	0,9172	61,04	0,9482	62,44	0,6896
4	73,33	0,9572	76,64	0,8423	77,3	0,7974	63,01	0,9143	61,19	0,9552	62,68	0,6867
5	73,25	0,9516	76,5	0,8463	77,65	0,7961	62,97	0,9138	60,78	0,9502	63,08	0,6868
6	74,56	0,9543	77,57	0,8451	77,02	0,7917	63,37	0,9124	60,38	0,9496	62,47	0,6896
7	74,61	0,9547	77,04	0,8498	77,67	0,7915	62,76	0,914	60,55	0,9487	63,19	0,6893
8	74,09	0,9564	77,46	0,8481	77,84	0,7925	63,51	0,9171	61,07	0,9562	62,96	0,6898
9	73,89	0,9589	76,73	0,8458	78,18	0,7923	63,59	0,9127	60,44	0,9517	62,39	0,6906
10	74,2	0,9564	77,33	0,8422	76,88	0,7953	63,24	0,9118	60,89	0,9562	63,27	0,6873
AVG	74,55	0,9516	77,24	0,8469	77,23	0,7983	62,68	0,9181	61,13	0,9499	62,72	0,6921
AVG	Wykrywalność [%]			69,26			Pewność			0,8594		

Tabela IV.24: Tabela pomiarów metody Gunnara Thalina, wersja jednoklatkowa (źródło: opracowanie własne)

Metoda Gunnara Thalina, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.	Wyk [%]	Pwn.
1	74,53	0,9663	77,07	0,8697	78,18	0,8159	62,79	0,9138	62,19	0,9401	63,13	0,6854
2	73,9	0,9574	77,15	0,8689	77,4	0,8147	62,78	0,9124	61,51	0,9338	63,72	0,6837
3	74,19	0,9658	76,72	0,867	77,23	0,8135	63,23	0,9099	61,39	0,9369	63,23	0,6856
4	74,05	0,958	77,02	0,8707	77,77	0,8079	63,39	0,9088	61,48	0,9417	63,32	0,6897
5	73,7	0,9586	76,35	0,8707	76,67	0,8083	63,26	0,9154	61,22	0,9377	63,66	0,6895
6	74,26	0,9616	76,92	0,8688	76,71	0,8148	63,72	0,9131	61,25	0,9411	63,26	0,6854
7	73,53	0,9593	77,29	0,8688	77,55	0,8142	62,91	0,9084	61,35	0,934	63,35	0,6879
8	73,91	0,9619	76,27	0,8632	77,04	0,8148	62,69	0,9117	61,82	0,9407	64,05	0,6881
9	74,52	0,9627	76,76	0,8661	77,06	0,81	62,54	0,9141	61,89	0,9407	63,24	0,6852
10	73,8	0,9578	77,41	0,8638	77,3	0,8106	62,95	0,9112	62,08	0,9375	63,27	0,6883
AVG	73,69	0,9645	76,68	0,8664	76,87	0,8104	62,47	0,9098	61,26	0,9364	63,99	0,6833
AVG	Wykrywalność [%]			69,16			Pewność			0,8618		

Tabela IV.25: Tabela pomiarów algorytmu Gunnara Thalina, wersja dwuklatkowa (źródło: opracowanie własne)

IV.3 Porównanie wykrywania pozycji przestrzennej wskaźników

Ostatnim badaniem, które obejmuje niniejsze opracowanie jest porównanie algorytmów usuwania przeplotu pod względem jakości wykrywania pozycji wskaźników w przestrzeni trójwymiarowej oraz orientacji wskaźnika względem kamery. Wybór metody porównania był, podobnie jak w poprzednich przypadkach, podyktowany związkiem istotności badanych parametrów w robotyce. Jakość określania położenia i nachylenia znaczników ma kluczowy wpływ w większości miejsc, gdzie stosowane są markery – trudno sobie wyobrazić prawidłowe działanie algorytmu odwzorowującego otoczenie w 3D, jeśli na wejściu dostawałby niedokładne lub w ogóle nieprawdziwe informacje dotyczące umiejscowienia znacznika w przestrzeni albo działanie robota, który ma precyzyjnie pochwycić delikatny obiekt i trafiający obok lub, co gorsza, w sam obiekt [26] [27] [28] [29].

Pomiar odchylenia pozycji jest rozumiany jako różnica pozycji w osiach X, Y oraz Z bieżącej pozycji wskaźnika w przestrzeni (wartości te są pobierane z macierzy translacji wykrytego markera) względem poprzednio wykrytej pozycji. Wartości odchylenia są uśredniane zarówno dla składowych X, Y oraz Z przestrzeni euklidesowej, jak i wszystkich takich odchylenia w czasie trwania całego wyszukiwania znaczników. Im wartość tego odchylenia jest mniejsza, tym trajektoria wykrywania znacznika jest bardziej dokładna, a co za tym idzie – płynna. Skala wartości odchylenia jest tożsama ze skalą biblioteki OpenGL, tj. dla $z = 0$, $x_{\min} = -1,333$, $x_{\max} = 1,333$, $y_{\min} = -1$ oraz $y_{\max} = 1$. Dla obiektów będących w głębi obrazu, skala zmniejsza się liniowo.

Odchylenie nachylenia znacznika względem kamery jest reprezentowane poprzez bezwzględną różnicę poszczególnych kątów eulerowskich markera bieżącego oraz znalezionej krok wcześniej. Powyższe odchylenie jest mierzone w stopniach. Dodatkowo, maksymalne odchylenie dla danego kąta nie może przekroczyć 180 stopni (nie ma znaczenia w którą stronę się marker odchyli, może to być maksymalnie -180 oraz +180 stopni), dlatego dla kątów większych niż półpełne – te kąty odejmuje się od kąta pełnego, z odpowiednim znakiem. Odchylenia, podobnie jak w przypadku pozycji, są uśredniane pomiędzy poszczególnymi trzema kątami (po jednym dla par osi X, Y i Z) oraz pomiędzy

wszystkimi odchyleniami w trakcie trwania danej próby. Im mniejsza średnia różnica kątów, tym bardziej dokładne jest odwzorowane nachylenie znacznika względem kamery.

IV.3.1 Metoda weaving

Weaving prezentuje się przeciętnie w odniesieniu do lokalizacji liniowej oraz kątowej markerów, zajmując środkowe pozycje w obu przypadkach. Próby nie odbiegają znacząco w żadnym punkcie w porównaniu z wynikami badań innych algorytmów (patrz: tabela IV.26).

Weaving												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	18,76	29,15	18,5	76,27	15,7	21,62	23,35	61,33	23,93	152,47	19,69	88,82
2	18,62	29,23	18,45	76,7	15,55	21,77	23,14	60,99	23,68	151,5	19,69	88,4
3	18,76	29,06	18,36	76,03	15,45	21,59	23,22	61,27	23,73	152,11	19,61	88,74
4	18,78	29,24	18,36	76,17	15,74	21,6	23,23	61,33	23,8	152,31	19,56	88,18
5	18,48	29,2	18,54	76,06	15,6	21,69	23,11	61,23	23,89	152,14	19,62	88,83
6	18,5	29,31	18,45	76,29	15,49	21,69	23,37	61,09	23,91	152,09	19,6	88,47
7	18,71	29,19	18,44	76,47	15,54	21,77	23,21	61,12	23,8	151,63	19,71	88,29
8	18,53	29,25	18,68	76,22	15,57	21,71	23,03	61,03	23,83	151,81	19,6	88,47
9	18,56	29,11	18,34	76,39	15,69	21,64	23,27	61,28	23,79	151,4	19,69	88,79
10	18,84	29,3	18,34	76,46	15,6	21,72	23,18	61,23	23,77	151,99	19,79	88,25
AVG	18,73	29,06	18,63	76,27	15,62	21,73	23,28	61,47	23,55	151,26	19,81	88,29
AVG	Odchylenie pozycji		19,94				Odchylenie kąta		71,35			

Tabela IV.26: Tabela pomiarów algorytmu weaving (źródło: opracowanie własne)

Objaśnienia:

Od. poz. oznacza średnią wartość odchylenia pozycji znacznika

Od. ką. oznacza średnią wartość odchylenia kątowego znacznika

IV.3.2 Metoda blending

Wyniki badań pozycji i nachylenia pasują algorytm blending podobnie do metody weaving – średnie odchylenie pozycji jest nieznacznie poprawione (choć należy zauważyć, że ta średnia mogłaby być lepsza, gdyby nie bardzo słaby wynik dla markera „O” dla prób z włączoną historią śledzenia), zaś średnia różnica kątowa jest minimalnie gorsza (patrz: tabela IV.27).

Blending												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	24,28	37,76	17,68	67,7	14	23,05	20,11	71,49	20,65	155,12	18,68	79,06
2	24,6	37,71	17,66	67,93	13,93	23,02	19,99	71,35	20,55	155,79	18,68	79,57
3	24,19	37,84	17,65	67,91	14,01	23,03	19,98	71,62	20,6	154,44	18,65	79,25
4	24,29	37,81	17,87	67,97	13,8	23,04	20,18	71,35	20,9	155,1	18,43	79,16
5	24,58	37,99	17,76	67,4	14,04	23,12	20,27	71,36	20,83	154,45	18,54	79,18
6	24,17	38,02	17,82	67,66	13,91	23,02	20,04	71,6	20,88	155,9	18,46	79,08
7	24,57	38,05	17,88	67,65	13,96	23,15	20,04	71,61	20,89	155,29	18,54	79,5
8	24,15	37,97	17,71	67,51	13,95	23,02	19,95	71,79	20,63	154,9	18,55	79,22
9	24,47	37,94	17,79	67,62	13,87	23,1	20,28	71,71	20,66	155,58	18,7	79,43
10	24,57	37,71	17,82	67,75	13,82	23,2	20,25	71,98	20,83	155,19	18,69	78,93
AVG	24,35	37,83	17,88	67,71	13,84	23,07	20,21	71,58	20,6	154,71	18,52	79,55
AVG	Odchylenie pozycji		19,23				Odchylenie kąta		72,41			

Tabela IV.27: Tabela pomiarów algorytmu blending (źródło: opracowanie własne)

IV.3.3 Metoda selective blending

Wyniki algorytmu selective blending plasują ten algorytm tuż przed metodami weaving oraz blending, notując niewielkie zwwyżki przy pomiarach średniej odchyień pozycji oraz nachylenia względem znacznika względem kamery (patrz: tabela IV.28).

Selective blending												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	24,1	36,54	17,42	66,34	13,87	23,09	20,03	69,26	20,27	151,71	18,35	77,72
2	24,28	36,73	17,66	66,64	13,74	23,01	19,84	69,3	20,52	151,73	18,33	77,73
3	24,2	36,64	17,52	66,22	13,82	23,05	20,18	69,27	20,31	150,89	18,39	77,47
4	24,03	36,6	17,45	66,55	13,63	22,93	19,96	69,09	20,32	151,09	18,38	77,59
5	24,29	36,62	17,5	66,59	13,85	23,05	19,96	69,47	20,29	150,61	18,34	77,72
6	24,13	36,75	17,62	66,65	13,68	23,04	19,84	69,05	20,26	151,24	18,34	77,79
7	24,01	36,72	17,62	66,74	13,82	23,05	20,01	69,39	20,53	151,17	18,42	77,44
8	24,19	36,41	17,62	66,23	13,67	23,01	20,19	69,03	20,18	151,08	18,37	77,49
9	23,96	36,61	17,4	66,81	13,8	23,1	19,97	69,02	20,2	151,36	18,16	77,84
10	24,27	36,43	17,62	66,35	13,89	22,98	19,82	69,07	20,4	150,72	18,19	77,56
AVG	24,08	36,52	17,46	66,54	13,77	22,95	19,95	69,07	20,13	151,62	18,15	78,01
AVG	Odchylenie pozycji		18,92				Odchylenie kąta		70,79			

Tabela IV.28: Tabela pomiarów algorytmu selective blending (źródło: opracowanie własne)

IV.3.4 Metody discard

O ile rodzina metod discard radziła sobie bardzo dobrze w testach wydajnościowych oraz przyzwoicie w testach wykrywania markera, tutaj, w przypadku badania związanego z pozycją znacznika i jego nachyleniem wobec rejestratora, kompletnie sobie nie radzi (poza przypadkiem odchylenia kąтового metody dwuklatkowej, o czym za chwilę). Wyniki średniego odchylenia liniowego (32,09 oraz 33,73 jednostek kolejno dla wariantu jednoklatkowego oraz wariantu dwuklatkowego) są najgorsze z zarejestrowanych. Średnie

odchylenie kątowe dla metody jednoklatkowej zajmuje ostatnią pozycję w rankingu, zaś dla metody dwuklatkowej – miejsce pierwsze (patrz: tabela IV.29 oraz IV.30). Autorowi opracowania trudno jest wytłumaczyć tą anomalię, ale znając przebieg próby oraz charakter wykrywania znaczników w obrazie przekształconym algorytmem discard (znaczniki są wykrywane pod bardzo różnymi kątami przez niemożliwość dopasowania kwadratowego wzorca do prostokątnego znacznika w obrazie) – ten wynik należałoby zaklasyfikować jako bardzo specyficzny przypadek i nie traktować tego wyniku wiążąco.

Discard, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	32,69	45,7	30,35	60,33	32,3	42,03	32,79	74	30,6	149,29	34,19	96,58
2	32,62	45,73	30,07	59,82	32,15	41,94	32,34	74,04	30,35	150,33	33,68	97,15
3	32,72	45,96	30,19	60,24	32,06	42,09	32,4	73,5	30,37	149,58	34,2	96,64
4	32,51	45,8	30,06	60,09	32,33	41,97	32,72	74,15	30,71	148,95	33,86	97,2
5	33,1	45,56	30,07	59,83	32,3	41,86	32,5	73,51	30,42	149,54	34,09	97,01
6	32,52	45,77	30,18	59,88	31,95	42,09	32,66	74,04	30,53	149,24	34,09	96,95
7	32,58	45,57	30,18	60,3	32,2	41,94	32,56	73,54	30,49	149,5	34,25	96,73
8	33,12	45,87	29,94	59,86	32,39	41,92	32,52	73,42	30,48	150,14	34,24	97,41
9	32,89	45,63	30,15	59,95	32,35	41,85	32,72	73,87	30,87	150,01	34,31	97,25
10	32,57	45,7	30,43	60,04	31,92	41,79	32,23	73,71	30,51	150,01	34,17	97,37
AVG	32,88	45,93	30,02	60,19	32,35	41,94	32,5	73,49	30,48	149,64	34,28	97,17
AVG	Odchylenie pozycji		32,09				Odchylenie kąta		78,06			

Tabela IV.29: Tabela pomiarów algorytmu discard, wersja jednoklatkowa (źródło: opracowanie własne)

Discard, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	25,19	37,45	27,29	41,1	27	36,68	67,32	26,78	26,77	147,62	29,24	89,45
2	25,22	37,34	27,05	41,21	27,05	36,41	66,78	26,83	27,14	147,72	29,18	89,85
3	25,45	37,24	27,29	41,4	27,12	36,45	66,07	26,73	26,95	148,07	28,98	89,32
4	25,28	37,44	27,08	41,41	27,4	36,46	66,56	26,68	27,24	146,81	29,27	89,92
5	25,27	37,39	27,21	41,47	27,22	36,56	66,31	26,8	26,73	146,78	28,84	89,5
6	25,08	37,15	27,26	41,28	27,19	36,65	66,37	26,83	27,17	147,43	28,81	89,69
7	25,02	37,19	27,23	41,39	27,22	36,43	66,97	26,85	27,13	146,7	28,9	89,56
8	25,31	37,26	27,22	41,23	27,42	36,66	66,55	26,69	26,92	147,91	29,23	89,93
9	25,36	37,24	26,97	41,42	26,92	36,58	66,74	26,71	26,95	146,89	29,04	89,79
10	25,19	37,25	26,79	41,29	27,35	36,45	66,37	26,78	26,81	147,73	29,23	89,87
AVG	25,39	37,29	26,85	41,41	27,29	36,5	67,19	26,73	26,77	147,34	28,9	89,27
AVG	Odchylenie pozycji		33,73				Odchylenie kąta		63,09			

Tabela IV.30: Tabela pomiarów algorytmu discard, wersja dwuklatkowa (źródło: opracowanie własne)

IV.3.5 Metody double

Wersja jednoklatkowa metody double plasuje się bardzo przeciętnie w ranking pomiarów odchyień, minimalnie lepiej niż weaving w pomiarze odchyień liniowych oraz nieznacznie lepiej od metody discard w przypadku odchylenia kąтового (patrz: tabela IV.31).

Dwuklatkowy wariant algorytmu double po raz pierwszy pokazuje się metod, które zwiększają rozdzielczość czasową wynikowej sekwencji obrazów – pomiar odchyień liniowych jest prawie dwukrotnie mniejszy niż w przypadku zarejestrowanych prób dla algorytmów jednoklatkowych. Odchylenie kątowe, w tym przypadku, jest lepsze niż dla pierwszej wersji double, ale nadal jest to wynik tylko przeciętny (patrz: tabela IV.32).

Double, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	21,49	36,34	21,41	89,74	16,15	26,22	20,25	54,55	20,58	161,88	18,33	86,97
2	21,36	36,29	21,36	89,74	16,02	26,3	20,2	54,62	20,76	161,88	18,18	86,83
3	21,5	36,62	21,68	89,52	16,17	26,22	20,27	54,69	20,44	161,98	18,1	86,4
4	21,54	36,37	21,4	89,56	16,02	26,19	20,06	54,33	20,67	162,3	18,09	86,61
5	21,68	36,55	21,45	89,46	15,98	26,26	20,31	54,46	20,57	161,68	18,17	86,88
6	21,27	36,27	21,41	89,51	16,17	26,3	20,26	54,67	20,5	162,34	18,25	86,6
7	21,47	36,55	21,38	89,33	16,04	26,23	20,08	54,22	20,56	162,98	18,21	86,99
8	21,56	36,59	21,43	89,18	15,88	26,2	20,39	54,2	20,53	161,98	18,3	87,14
9	21,4	36,41	21,44	89,57	15,96	26,08	20,05	54,39	20,61	161,61	18,18	87,03
10	21,53	36,45	21,35	89,26	16	26,13	20,02	54,39	20,58	162,05	18,14	86,87
AVG	21,55	36,5	21,68	89,62	16,13	26,32	20,2	54,43	20,47	162,01	18,2	86,59
AVG	Odchylenie pozycji		19,71				Odchylenie kąta		75,91			

Tabela IV.31: Tabela pomiarów algorytmu double, wersja jednoklatkowa (źródło: opracowanie własne)

Double, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	11,01	28,41	11,21	77,9	8,54	18,21	11,92	50,38	12,32	164,4	12,02	92,07
2	11,04	28,43	11,11	77,71	8,64	18,13	11,94	50,24	12,31	163,99	12,12	92,45
3	11,07	28,39	11,12	77,75	8,58	18,15	11,89	50,34	12,45	163,57	12,11	92,01
4	11,03	28,34	11,16	78,29	8,5	18,2	11,94	50,23	12,45	164,97	12,1	92,71
5	11,09	28,42	11,22	77,93	8,49	18,11	12,01	50,07	12,46	164,92	12,14	92,1
6	11	28,3	11,1	78,15	8,61	18,21	11,81	50,24	12,46	164,23	12,15	92,82
7	11,14	28,4	11,12	78,24	8,62	18,13	11,95	50,32	12,39	163,99	11,99	92,26
8	11,03	28,24	11,28	77,8	8,59	18,19	11,78	50,18	12,33	164,01	11,97	91,99
9	11,01	28,42	11,16	77,97	8,63	18,27	11,9	50,21	12,32	164,44	12,15	92,07
10	11,06	28,28	11,23	78,38	8,55	18,15	11,86	50,21	12,26	163,4	12,02	92,06
AVG	11,18	28,32	11,11	77,72	8,62	18,23	11,81	50,25	12,45	164,46	12,14	92,25
AVG	Odchylenie pozycji		11,22				Odchylenie kąta		71,87			

Tabela IV.32: Ramka wynikowa algorytmu double, wersja dwuklatkowa (źródło: opracowanie własne)

IV.3.6 Metody quarter

Wyniki obu wariantów metody quarter można zaklasyfikować jako przeciętne, ze średnią wartością odchyień pozycjonowania znaczników na poziomie 20,34 jednostki dla wersji jednoklatkowej oraz 18,29 jednostki dla wersji dwuklatkowej (co samo w sobie jest dobrym wynikiem, ale słabym wobec reszty wersji dwuklatkowych – co uwidacznia, że czterokrotne zmniejszenie liczby pikseli w obrazie nie jest rekompensowane przez zwiększoną rozdzielczość czasową); w przypadku odchyień kątowych średni wynik wariantu jednoklatkowego (74,61°) jest nieznacznie poprawiony w drugim wariantcie (70,24°, patrz: tabele IV.33 oraz IV.34).

Quarter, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	21,05	46,42	20,16	86,52	16,49	24,59	21,94	60,34	23,56	149,8	18,32	81,05
2	21,25	46,37	20,29	86,41	16,42	24,45	21,85	60,42	23,57	148,7	18,32	81,09
3	20,95	46,01	20,3	86,43	16,32	24,54	21,72	60,4	23,9	148,68	18,52	81,32
4	20,88	46,03	20,2	86,69	16,45	24,44	21,68	59,97	23,86	148,83	18,61	81,31
5	21,06	46,04	20,23	86,3	16,29	24,58	21,82	60,36	23,88	148,84	18,55	80,72
6	21,18	46,22	20,23	85,94	16,47	24,45	21,83	60,41	23,49	149,92	18,3	80,99
7	21,25	46,29	20,35	86,08	16,28	24,42	21,69	60,34	23,74	149,74	18,38	80,86
8	21,27	46,2	20,21	86,71	16,21	24,47	21,56	59,9	23,9	149,25	18,52	80,59
9	21,08	46,25	20,34	86,4	16,3	24,38	21,54	60,12	23,92	149,45	18,4	80,77
10	20,93	46,23	20,49	85,99	16,33	24,59	21,69	60,3	23,85	148,67	18,55	80,59
AVG	21,07	46,36	20,4	86,62	16,3	24,51	21,73	60,25	23,9	149,21	18,61	80,7
AVG	Odchylenie pozycji		20,34				Odchylenie kąta		74,61			

Tabela IV.33: Ramka wynikowa algorytmu quarter, wersja jednoklatkowa (źródło: opracowanie własne)

Quarter, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	19,12	46,36	18,12	76,3	13,67	18,23	19,66	56,92	21,73	145,12	17,06	78,94
2	19,21	46,41	18,17	76,26	13,44	18,25	19,8	56,93	21,61	145,62	17,31	78,87
3	19,2	46,08	18,28	76,4	13,66	18,36	19,63	57,02	21,71	145,33	17,25	78,68
4	19	46,24	18,34	76,5	13,64	18,23	19,61	56,84	21,62	145,88	17,19	78,77
5	18,98	46,21	18,47	76,12	13,59	18,36	19,68	56,93	21,89	144,88	17,07	79,28
6	19,19	46,16	18,27	76,5	13,6	18,36	19,66	56,96	21,58	145,36	17,01	78,68
7	19,02	46,19	18,28	76,22	13,54	18,2	19,87	57,3	21,62	146	17,26	78,64
8	19,2	46,39	18,43	76,71	13,6	18,24	19,88	57,21	21,57	144,88	17,33	78,93
9	19,23	46,21	18,4	76,07	13,48	18,34	19,88	57,06	21,53	145,55	17,18	78,71
10	19,09	46,29	18,32	76,38	13,49	18,35	19,86	57,29	21,76	145,25	17,08	79,3
AVG	19,16	46,22	18,25	76,2	13,45	18,28	19,91	56,89	21,87	145,09	17,07	78,76
AVG	Odchylenie pozycji		18,29				Odchylenie kąta		70,24			

Tabela IV.34: Tabela pomiarów algorytmu quarter, wersja dwuklatkowa (źródło: opracowanie własne)

IV.3.7 Metoda progressive scan

Podczas tego testu, algorytm progressive scan uplasował się wynikami na bardzo wysokiej pozycji (pozycja trzecia w przypadku odchyłeń liniowych oraz pozycja szósta dla odchyłeń kątowych, patrz: tabela IV.35). Wyniki wskazują na niewielką poprawę jakości wyznaczania pozycji względem metody double (dwuklatkowej) oraz umiarkowaną poprawę badania odchyłeń kątowych – co najprawdopodobniej wynika z poprawy jakości obrazu w mniej dynamicznych scenach.

Progressive scan												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	11,03	27,34	11,1	74,16	8,46	17,19	11,59	47,33	12,35	154,12	12,15	91,08
2	11,05	27,32	11,17	73,75	8,42	17,18	11,51	47,25	12,33	154,3	12,12	91,5
3	10,9	27,29	10,99	73,73	8,45	17,26	11,61	47,33	12,41	154,95	12,16	91,16
4	10,99	27,36	10,99	73,97	8,53	17,15	11,62	47,25	12,29	154,14	12,09	91,18
5	10,94	27,48	11,04	74,37	8,38	17,17	11,52	47,11	12,33	154,51	12,1	91,78
6	10,99	27,41	11,14	74,12	8,48	17,2	11,71	47,37	12,21	154,4	11,97	91,64
7	11,06	27,28	11,1	73,95	8,44	17,21	11,67	47,45	12,23	154,69	12,03	91,45
8	11,06	27,24	11,11	74,03	8,48	17,16	11,61	47,44	12,27	153,55	12,06	91,73
9	10,98	27,25	11,18	73,97	8,41	17,19	11,59	47,54	12,27	154,6	12,06	91,07
10	10,9	27,36	11,14	74,31	8,48	17,24	11,49	47,34	12,39	153,41	12,1	91,61
AVG	10,95	27,4	11,13	74,2	8,48	17,13	11,64	47,46	12,4	154,15	11,99	91,61
AVG	Odchylenie pozycji		11,1				Odchylenie kąta		68,66			

Tabela IV.35: Tabela pomiarów algorytmu progressive scan (źródło: opracowanie własne)

IV.3.8 Metoda autorska

Obie wersje metody autorskiej zajęły bardzo wysokie miejsca w ogóle wyników tej próby. Wersja jednoklatkowa, z wynikiem 17,54 jednostek dla pomiaru odchylenia pozycjonowania oraz wynikiem 67,68° dla odchylenia kątowych zajmuje pozycję piątą i czwartą, odpowiednio (należy zauważyć, że są to najlepsze wyniki wśród metod jednoklatkowych, patrz: tabela IV.36). Wynik w postaci 9,83 jednostki średniego odchylenia liniowego jest najlepszym zarejestrowanym wynikiem wśród algorytmów, bezsprzecznie wyprzedzając pozostałe algorytmy; podobnie jest z pomiarem odchylenia kąтового (64,38°, nie licząc problematycznego wyniku dla dwuklatkowej wersji algorytmu discard, patrz: tabela IV.37). Ciekawą obserwacją dla tej metody jest obniżenie skuteczności wykrywania nachylenia znacznika dla markera „X” – co najprawdopodobniej wynika ze zwięzienia poszczególnych linii markera na skutek działania algorytmu interpolującego.

Metoda autorska, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	20	33,85	17,6	65,96	12,76	17,41	19,37	51,23	19,31	160,09	16,3	76,71
2	19,9	33,78	17,72	65,97	12,63	17,39	19,43	51,21	19,48	160,46	16,34	76,95
3	19,89	33,8	17,63	66,03	12,78	17,32	19,53	51,09	19,43	160,81	16,31	76,63
4	20,13	33,99	17,62	66,16	12,8	17,37	19,29	51,41	19,47	160,35	16,43	77,32
5	20,11	33,77	17,55	66,05	12,67	17,3	19,44	51,14	19,44	160,74	16,33	76,89
6	19,94	33,96	17,54	65,9	12,64	17,31	19,46	51,2	19,42	159,45	16,53	76,96
7	19,86	33,93	17,65	66,08	12,74	17,45	19,43	51,21	19,48	160,24	16,52	76,83
8	20,2	33,92	17,53	65,98	12,72	17,4	19,22	51,02	19,45	160,27	16,48	76,73
9	20,14	33,72	17,6	66,04	12,7	17,39	19,57	51,4	19,55	160,62	16,6	77,25
10	19,85	33,93	17,53	66,02	12,7	17,42	19,35	51,04	19,49	160,27	16,55	76,58
AVG	19,83	33,86	17,46	66,13	12,62	17,39	19,47	51,37	19,39	160,33	16,47	76,98
AVG	Odchylenie pozycji		17,54				Odchylenie kąta		67,68			

Tabela IV.36: Tabela pomiarów algorytmu autorskiego, wersja jednoklatkowa (źródło: opracowanie własne)

Metoda autorska, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	10,46	26,12	11	64,85	6,47	12,71	11,35	46,36	11,41	163,55	8,33	72,85
2	10,34	26,14	10,95	64,61	6,47	12,71	11,25	46,4	11,45	163,32	8,35	73,14
3	10,38	26,08	11,01	64,46	6,49	12,7	11,25	46,57	11,34	163,68	8,37	73,25
4	10,42	26,14	11,03	64,3	6,47	12,71	11,4	46,48	11,49	163,11	8,39	73,06
5	10,46	26,28	10,89	64,29	6,4	12,73	11,43	46,71	11,54	163,81	8,35	73
6	10,48	26,06	10,93	64,68	6,47	12,76	11,27	46,29	11,34	163,13	8,3	72,82
7	10,46	26,29	11,02	64,35	6,45	12,77	11,43	46,33	11,4	162,57	8,29	73,14
8	10,46	26,1	10,91	64,63	6,46	12,81	11,38	46,38	11,55	162,58	8,26	72,82
9	10,38	26,15	10,89	64,41	6,43	12,73	11,41	46,56	11,37	162,79	8,39	72,58
10	10,31	26,19	11,02	64,53	6,4	12,72	11,42	46,46	11,34	162,81	8,33	72,95
AVG	10,33	26,22	10,86	64,84	6,43	12,79	11,41	46,59	11,47	163	8,37	72,81
AVG	Odchylenie pozycji		9,81				Odchylenie kąta		64,38			

Tabela IV.37: Tabela pomiarów algorytmu autorskiego, wersja dwuklatkowa (źródło: opracowanie własne)

IV.3.9 Metoda Gunnara Thalina

Podobnie jak w przypadku metody autorskiej, obie wersje metody Gunnara Thalina osiągnęły bardzo dobre rezultaty w tej próbie. Wyniki obu wersji algorytmów, zarówno dla wersji jednoklatkowej oraz dwuklatkowej osiągnęły rezultaty zbliżone z metodą autorską, odpowiednio (patrz tabele: IV.38 oraz IV.39):

- z wynikiem 17,59 jednostek dla pomiaru odchylenia pozycjonowania oraz wynikiem 67,72° dla odchylenia kątowych,
- z wynikiem w postaci 9,85 jednostki średniego odchylenia liniowego jest najlepszym zarejestrowanym wynikiem wśród algorytmów, bezsprzecznie wyprzedzając pozostałe algorytmy; podobnie jest z pomiarem odchylenia kąтового (64,34°).

Metoda Gunnara Thalina, wersja jednoklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	19,91	33,75	17,77	65,95	12,75	17,43	19,43	51,25	19,4	160,29	16,34	77,06
2	19,84	33,74	17,44	66,08	12,8	17,35	19,19	51,07	19,25	159,71	16,2	77,02
3	19,99	33,98	17,76	65,92	12,68	17,36	19,48	51,23	19,35	160,15	16,41	76,52
4	20,18	33,8	17,76	65,99	12,79	17,45	19,48	51,17	19,19	160,07	16,41	76,45
5	20,05	33,75	17,66	65,9	12,71	17,38	19,39	51,17	19,35	160,33	16,17	76,77
6	19,83	33,83	17,48	65,75	12,87	17,33	19,39	51,39	19,23	160,31	16,42	76,86
7	19,89	33,84	17,59	66	12,75	17,48	19,29	51,26	19,23	159,29	16,18	76,51
8	19,84	33,93	17,6	65,84	12,68	17,42	19,3	51,07	19,42	159,79	16,31	76,45
9	20,16	33,94	17,51	65,78	12,65	17,37	19,42	51,29	19,22	160,06	16,37	76,54
10	20,06	33,92	17,49	65,85	12,84	17,45	19,53	51,01	19,26	159,92	16,24	77,08
AVG	20,11	33,68	17,46	66,12	12,82	17,47	19,39	51,21	19,33	160,8	16,4	77,05
AVG	Odchylenie pozycji		17,59				Odchylenie kąta		67,72			

Tabela IV.38: Tabela pomiarów algorytmu Gunnara Thalina, wersja jednoklatkowa (źródło: opracowanie własne)

Metoda Gunnara Thalina, wersja dwuklatkowa												
Próba	Historia śledzenia włączona						Bez historii śledzenia					
	O		X		TEST		O		X		TEST	
	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.	Od. poz.	Od. ką.
1	10,41	26,25	10,95	64,64	6,51	12,7	11,36	46,44	11,53	163,58	8,28	73,01
2	10,32	26,19	10,98	64,65	6,4	12,76	11,31	46,56	11,41	162,72	8,29	72,71
3	10,45	26,29	10,86	64,77	6,51	12,7	11,29	46,42	11,54	163,66	8,28	72,64
4	10,31	26,13	10,86	64,24	6,39	12,82	11,25	46,34	11,4	162,99	8,35	72,86
5	10,44	26,17	11	64,59	6,4	12,74	11,41	46,39	11,41	163,1	8,26	72,97
6	10,5	26,28	10,95	64,79	6,42	12,73	11,26	46,32	11,51	162,79	8,28	73,24
7	10,37	26,3	10,96	64,72	6,5	12,7	11,26	46,72	11,44	162,75	8,27	73,01
8	10,35	26,05	10,92	64,32	6,41	12,7	11,37	46,46	11,48	163,17	8,33	72,97
9	10,38	26,28	10,88	64,78	6,4	12,77	11,35	46,39	11,55	163,48	8,33	72,65
10	10,44	26,28	10,92	64,47	6,51	12,74	11,25	46,59	11,51	163,55	8,38	72,68
AVG	10,49	26,14	10,87	64,78	6,45	12,81	11,42	46,6	11,56	162,58	8,28	73,12
AVG	Odchylenie pozycji			9,85			Odchylenie kąta			64,34		

Tabela IV.39: Tabela pomiarów algorytmu Gunnara Thalina, wersja dwuklatkowa (źródło: opracowanie własne)

Podsumowaniem tego rozdziału jest tabela ogólnego porównania zastosowania poszczególnych algorytmów usuwania przeplotu (oceny zostały nadane w skali od 1 do 10):

Tabela ogólnego porównania algorytmów				
Numer	Metoda	Zastosowanie	Ocena wydajności	Ocena jakości
1	Weaving	Nieadekwatna do większości zastosowań, bardzo zniekształca obraz	10	1
2	Blending	Nieadekwatna do większości zastosowań, bardzo zniekształca obraz	2	1
3	Selective blending	Nieadekwatna do większości zastosowań, bardzo zniekształca obraz	1	2
4	Discard, jednoklatkowa	Nieadekwatna do większości zastosowań, zniekształca stosunek rozmiarów obrazu	7	4
5	Discard, dwuklatkowa	Nieadekwatna do większości zastosowań, zniekształca stosunek rozmiarów obrazu	4	4
6	Double, jednoklatkowa	Ogólnie dobra metoda, nie wyróżnia się w żadnej kategorii	8	6
7	Double, dwuklatkowa	Ogólnie dobra metoda, nie wyróżnia się w żadnej kategorii	3	6
8	Quarter, jednoklatkowa	Mały obraz wynikowy, dobra do bliskich zastosowań	10	3
9	Quarter, dwuklatkowa	Mały obraz wynikowy, dobra do bliskich zastosowań	9	3
10	Progressive scan	Ogólnie dobra metoda, zastosowanie w szerokim spektrum rodzajów sekwencji obrazów	2	8
11	Metoda autorska, jednoklatkowa	Bardzo dobra jakość obrazu, zastosowanie w szerokim spektrum rodzajów sekwencji obrazów	8	9
12	Metoda autorska, dwuklatkowa	Bardzo dobra jakość obrazu, zastosowanie w szerokim spektrum rodzajów sekwencji obrazów	6	10
13	Gunnar-Thalin, jednoklatkowa	Bardzo dobra jakość obrazu, zastosowanie w szerokim spektrum rodzajów sekwencji obrazów	7*	9
14	Gunnar-Thalin, dwuklatkowa	Bardzo dobra jakość obrazu, zastosowanie w szerokim spektrum rodzajów sekwencji obrazów	4*	10

Tabela IV.40: Tabela ogólnego porównania algorytmów (źródło: opracowanie własne)

* Wartości szacowane na podstawie zastosowanych uproszczeń oraz optymalizacji w trakcie implementacji metody autorskiej

Wnioski i podsumowanie

V.1 Wnioski z badań

Wyniki poszczególnych testów przeprowadzonych ramach rozdziału czwartego precyzyjnie pokazują właściwości wydajnościowe oraz jakościowe przytoczonych algorytmów usuwania przeplotu.

Algorytm metody autorskiej, w wersji dwuklatkowej, w czterech z pięciu badanych kategorii zajął pierwsze miejsce wśród badanych metod, w szczególności: liczbie wykrytych znaczników obrazu, pewności wykrycia markera, odchylenia liniowego i nachylenia znacznika – tak naprawdę kategorii najważniejszych przy śledzeniu obiektów w sekwencji obrazów (oraz wiążących się z tym zastosowań w robotyce). Metoda nie jest wprawdzie najszybsza – szybszych od niej kilka metod, w tym jedna dwuklatkowa – ale żadna z nich nie dorównuje możliwościami algorytmowi autorskiemu. Co więcej, warto zauważyć, że osiągnięty wynik przetwarzania 47-miu klatek na sekundę na ośmio-letnim procesorze to nadal bardzo dobry wynik, w zupełności wystarczający do przetwarzania obrazu w trybie on-line, co jak już wcześniej zostało wskazane – ma kluczowe znaczenie dla wszelkich zastosowań mobilnych.

Druga wersja metody autorskiej, wariant jednoklatkowy, jest już ponad półtora razy szybszy od powyższej metody i co najważniejsze, wcale nie jej ustępuje możliwościami. W prawie wszystkich kategoriach (tylko oprócz kategorii wydajnościowej) metoda wypada lepiej od innych algorytmów jednoklatkowych, a nawet, w kilku przypadkach, lepiej od algorytmów dwuklatkowych. Jeśli chodzi o test wydajnościowy – z metod jednoklatkowych – metoda autorska ustępuje tylko jednej, metodzie doubling; różnica jest niewielka – niecałe 20% w przetwarzanej liczbie klatek na sekundę.

Z pełną odpowiedzialnością można powiedzieć, że cel pracy został osiągnięty – poszczególne algorytmy zostały szczegółowo porównane oraz zostało dowiedzione, że w szerokim spektrum zastosowań, nie tylko związanym z śledzeniem znaczników, obie wersje metody autorskiej przewyższają możliwościami metody pozostałe.

V.2 Możliwości rozwoju

Jednym z aspektów, które porusza niniejsze opracowanie, jest aspekt wydajnościowy przetwarzania sekwencji obrazów. Autor przedstawił porównanie według tego (wydajnościowego) kryterium w oparciu o badania przeprowadzone na dwóch platformach – komputerach klasy Intel Pentium 4 oraz Intel Atom, które są platformami jednordzeniowymi. W obecnej chwili standardem są już platformy 2-, 4-, 6-, a nawet 8-rdzeniowe. Porównanie wydajnościowe na powyższych platformach mogłoby rozszerzyć dokonane przez autora porównanie, ale w trakcie pisania tej pracy, autor nie posiadał dostępu do powyższych technologii. Istniejące opracowania pokazują, że wzrost wydajności na poszczególnych procesorach wielordzeniowych nie jest wprawdzie liniowo zależny od liczby rdzeni, ale przykładowo dla procesora o czterech rdzeniach, ten współczynnik można przyjmować za „3”, mając dodatkowo na uwadze podatność algorytmów przetwarzania obrazów na wykorzystanie wielowątkowości (wszystkie przytoczone w niniejszym opracowaniu algorytmy opierają się na pętlach i mogą z powodzeniem zostać rozbite na n-mniejszych pętli, gdzie n to liczba wątków, które są do wykorzystania na danej platformie) [30].

Implementacja algorytmów w tejże pracy, ze względów tożsamyh z tymi z poprzedniego akapitu, nie przewiduje również zastosowania kart graficznych do przeprowadzania obliczeń. Taką zaś możliwość oferują nowoczesne karty graficzne posiadające zunifikowane jednostki cieniujące (dotyczy to zarówno dedykowanych, jak i zintegrowanych kart graficznych), dzięki którym możliwe jest skorzystanie z bibliotek takich jak OpenCL (Open Computing Library) do wykonywania jednorodnych komputacji. Wprawdzie rdzenie cieniujące nie są tak wydajne jak nowoczesne procesory, ale ich liczba może sięgać od kilkunastu do nawet kilkuset w bardziej zaawansowanych kartach graficznych dostępnych na rynku komercyjnym.

Ciekawą możliwością rozwoju metody autorskiej, jest możliwość automatyzacji wykrywania najlepszych parametrów uruchomieniowych metody (wartość progowa oraz wartość krawędziowa). Autor opracowania dobierał parametry eksperymentalnie, ale nic nie stoi na przeszkodzie, aby utworzyć wersję programu, która automatycznie wskaże

najlepsze wartości obu zmiennych progowych, przy których procentowość odnajdywania wskaźnika w obrazie będzie najwyższy – wartość dobranych przez autora parametrów dobrze spisuje się w badanym przypadku, ale może być mniej adekwatna w innych warunkach (np. bardziej dynamiczne tło, zmienne oświetlenie).

Zamiarem autora było również zbadanie możliwości, jakie prezentują metody typu motion compensation, (autorowi) nie udało się znaleźć żadnego rozwiązania w tej dziedzinie, które byłoby ogólnodostępne na rynku. W przemyśle filmowym, na potrzeby edycji zapisywanych cyfrowo, filmy nagrywane w technice analogowej każdorazowo są analizowane ręcznie klatka-po-klatce i w zależności od obrazu, dobierane jest najbardziej odpowiednie rozwiązanie dla danego przypadku. Metoda ta jest bardzo czasochłonna i bardzo droga w użyciu, przez co wykorzystywana jest głównie w ramach wysokobudżetowych projektów.

V.3 Podsumowanie

Ważnym pytaniem, które może zadać sobie czytelnik po lekturze niniejszego opracowania, powinno być pytanie „czy niniejsza praca jest nadal aktualna?”.

Najprostszą odpowiedzią jest – „tak”; mimo upływu lat technologia przeplotu (oraz co za tym idzie, potrzeby jego usuwania) nadal jest bardzo popularna. Mogłoby się wydawać, że wraz z wyparciem ekranów kineskopowych era tej zamierzchłej technologii dobiegnie końca, ale pragmatyzm i adaptacja do współczesnych warunków pozwala przeplotowi przeżywać „drugą młodość”. Od końca lat dziewięćdziesiątych ubiegłego wieku, gdy do standardu wizyjnej elektroniki użytkowej zaczęły trafiać ekrany ciekłokrystaliczne sterowane cyfrowo, pojawiła się potrzeba usuwania przeplotu z zapisów z kamer analogowych z wcześniejszych kilkadziesiąt lat. Wraz ze zmieniającym się formatem zapisu strumieni video (obecnie już upowszechnionych formatów HD oraz Full HD, a także rozwijanych, np. 4K) zachodzi potrzeba przetworzenia archiwalnego zapisu do formatu cyfrowego wysokiej jakości. I tak ze wzrostem liczby pikseli możliwych do wyświetlenia na ekranie odbiornika, rosną wymagania stawiane przed technikami usuwania przeplotu. O ile większość istniejących filtrów przystosowanych jest do przetwarzania filmów przed jego

wyświetleniem, gdzie liczy się jak najlepszy efekt wizualny, to zawsze istnieje możliwość przystosowania ich (filtrów) do wymagań jakie stawia przetwarzanie strumienia obrazu w czasie rzeczywistym, które muszą spełniać rygorystyczne warunki czasowe narzucone przez liczbę przetwarzanych klatek na sekundę obrazu. Dodatkowo, cyfrowe kamery oraz rejestratory przemysłowe obrazu zaczęły wypierać ich analogowe odpowiedniki dopiero 10 lat później; proces ten nadal trwa, gdyż większość starych urządzeń w dalszym ciągu w pełni zaspokaja potrzeby użytkowników – przy obecnym braku znaczących bodźców ukierunkowujących te zmiany, ta tendencja może się długo utrzymywać. Co więcej, na rynku elektroniki użytkowej oraz przemysłowej w dalszym ciągu pojawiają się urządzenia cyfrowe, które zachowują obraz korzystając z techniki przeplotu. Podyktowane jest to niczym innym tylko oszczędnościami na podzespołach, które mieszczą się w atrakcyjnym przedziale cenowym, nie są w stanie obsłużyć zbyt szerokiego strumienia danych na wyjściu przetwornika – stąd częstokroć w specyfikacjach urządzeń komórkowych oraz amatorskim kamer cyfrowych znajdują oznaczenia „720i”, „1080i” – świadczące o wykorzystaniu przeplotu w jednym z etapów rejestracji obrazu – należy jednak dodać, że coraz więcej urządzeń tego typu zapewnia możliwość skanowania progresywnego. Co więcej, standardy „720i” oraz „1080i” są szeroko stosowane również w przekazie telewizyjnym i jest wynikiem ograniczenia potrzebnego pasma na przesył obrazu z przeplotem względem obrazu progresywnego – obraz z podziałem na pola wymaga tylko połowy pasma potrzebnego na przesłanie obrazu bez przeplotu [31].

Bibliografia

- [1] Deinterlacing Guide [online], trac.handbrake.fr, 2012 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <https://trac.handbrake.fr/wiki/DeinterlacingGuide#whatisit>
- [2] Interlaced Video on a Computer Display, <http://scanline.ca>, 2012 [data dostępu: 10 czerwca 2012]. Dostępne w internecie: <http://scanline.ca/deinterlacing/visual.html>
- [3] Standard-definition television [online], Wikipedia, 2012 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://en.wikipedia.org/wiki/Standard-definition_television
- [4] HDTV [online], Wikipedia, 2012 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://en.wikipedia.org/wiki/Hdtv>
- [5] 1080p [online], Wikipedia, 2012 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://en.wikipedia.org/wiki/1080p>
- [6] From VCRs to IP-Surveillance, Axis Communications, 2003 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://www.axis.com/ja/documentation/whitepaper/video/vcr_to_ipsurveillance.pdf
- [7] Total Cost of Ownership (TCO), Comparison of IP - and analog-based surveillance systems, Axis Communications, 2008 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://www.axis.com/files/whitepaper/wp_axis_tco_31196_en_0802_lo.pdf
- [8] Deinterlacing [online], Wikipedia, 2012 [data dostępu: 10 Czerwiec 2012]. Dostępne w Internecie: <http://en.wikipedia.org/wiki/Deinterlacing>
- [9] What is deinterlacing? Facts, solutions, examples [online], 100fps.com [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://www.100fps.com>
- [10] When is it harmful to deinterlace? When is it better to leave a picture interlaced? [online], 100fps.com [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://www.100fps.com/video_resolution_vs_fluidity.htm
- [10] Lee, G.G.; Li, H.-T.; Wang, M.-J.; Lin, H.-Y.: Motion Adaptive Deinterlacing via Edge Pattern Recognition. W: IEEE Transactions on Circuits and Systems, 2007, s. 2662 - 2665.
- [11] Jeon, G.; Won, J.; Lee, R.; Jeong, J.: A Rough Set Approach For Video Deinterlacing. W: IEEE Transactions on Multimedia and Expo, 2007, s. 1942 - 1945
- [12] Jeon, G.; Falcon, R.; Kim, D.; Lee, R.; Jeong, J.: Application of Bayesian Belief Network in Reliable Analysis for Video Deinterlacing. W: IEEE Transactions on Consumer Electronics, 2008, s. 123 - 130
- [13] Lee, G. G.; Ling, H.-Y.; Wang, M.-J.; Lai, R.-L.; Jhuo, C. W.: A high-quality spatial-temporal content-adaptive deinterlacing algorithm. W: IEEE International Symposium on Circuits and Systems, 2008, s. 2594 - 2597
- [14] Chiang, A.; Fan, Y.-C.: Novel Artifact Detection for Motion Compensated Deinterlacing. W: Canadian Conference on Electrical and Computer Engineering, 2008, s. 493 - 496

- [15] Jeon, G.; Anisetti, M.; Bellandi, V.; Damiani, E.; Jeong, J.; Suh I. H.: Robust Fuzzy Filter for Noise Reduction in Video Deinterlacing. W: IEEE International Conference on Signal Image Technology and Internet Based Systems, 2008, s. 511 - 517
- [16] Dai, S.; Baker, S.; KangAn, S. B.: MRF-Based DeInterlacing Algorithm With Exemplar-Based Refinement, Ieee Transactions On Image Processing, Vol. 18, No. 5, 2009, s. 956 - 958
- [17] Brox,, P.; Baturone, I.; Sánchez-Solano, S.: Fuzzy Motion-Adaptive Interpolation With Picture Repetition Detection for Deinterlacing. W: IEEE Transactions On Instrumentation And Measurement, Vol. 58, No. 9, 2009, s. 2952 - 2958
- [18] Tu, S.-F.; Au, O. C.; Wu, Y.; Luo, E.; Yeung, C.-H.: A Robust Spatial-Temporal Line-Warping Based Deinterlacing Method. W: IEEE International Conference on Multimedia and Expo, 2009, s. 77 - 80
- [19] Yang, S.; Kim, D.; Jeong, J.: Fine Edge-Preserving Deinterlacing Algorithm for Progressive Display. W: IEEE Transactions on Consumer Electronics, Vol. 55, No. 3, 2009, s. 1654 - 1662
- [20] Parkm S.-J.; Jeon, G.; Jeong, J.: Deinterlacing Algorithm using Edge Direction from Analysis of the DCT Coefficient Distribution. W: IEEE Transactions on Consumer Electronics, Vol. 55, No. 3, 2009, s. 1674 - 1681
- [21] Kernighan, B; Ritchie, D.: The C Programming Language, Prentice Hall, 1988 [22] Fog, A.: Optimizing software in C++ , An optimization guide for Windows, Linux and Mac, angner.org [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://www.agner.org/optimize/optimizing_cpp.pdf
- [23] VirtualDub Filters, [online], guthspot.se [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://www.guthspot.se/video/index.htm>
- [24] Recommendation ITU-R BT.601-7. Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios, International Telecommunication Union, 2011 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://www.itu.int/rec/R-REC-BT.601-7-201103-l/en>
- [25] All About Video Fields, Lurkertech [online], 2011 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://lurkertech.com/lg/fields>
- [26] Klopschitz, M.; Schmalstieg, D.: Automatic Reconstruction of Wide-Area Fiducial Marker Models. W: IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, s. 71 - 74
- [27] Lim, H.; Lee, Y. S.: Real-Time Single Camera SLAM Using Fiducial Markers. W: ICROS-SICE International Joint Conference, 2009, s. 177 - 182
- [28] Stathakis, A.; Petriu, E. M.: Robust Pseudo-Random Fiducial Marker for Indoor Localization. W: IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2011, s. 19 - 24
- [29] Hamzeen, M. H. M.; Perera, M. U. S.: Pattern Independent Fiducial Marker Detection for an Interactive Public Display. W: The International Conference on Advances in ICT for Emerging Regions, 2011, s. 52 - 57
- [30] Multi-Core Scaling Performance of AMD's Bulldozer [online], Phoronix, 2011 [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: http://www.phoronix.com/scan.php?page=article&item=amd_bulldozer_scaling
- [31] High Definition (HD) Image Formats for Television Production, EBU-UER, 2010, [data dostępu: 10 czerwca 2012]. Dostępne w Internecie: <http://tech.ebu.ch/docs/tech/tech3299.pdf>

