

KATEDRA TELEKOMUNIKACJI MULTIMEDIALNEJ I MIKROELEKTRONIKI

WYDZIAŁ ELEKTRONIKI I TELEKOMUNIKACJI

POLITECHNIKA POZNAŃSKA

Streszczenie rozprawy

**SYNTEZA WIDOKÓW WIRTUALNYCH
W RZADKICH SYSTEMACH WIELOKAMEROWYCH
DLA ZASTOSOWAŃ W SWOBODNEJ NAWIGACJI**

Adrian Dziembowski

Promotor: prof. dr hab. inż. Marek Domański

Promotor pomocniczy: dr inż. Olgierd Stankiewicz

Poznań 2018

SPIS TREŚCI STRESZCZENIA

Spis treści streszczenia.....	2
1. Wprowadzenie.....	3
1.1. Synteza widoków wirtualnych dla swobodnej nawigacji.....	3
1.2. Problemy charakterystyczne dla rzadkich systemów wielokamerowych.....	4
1.3. Cel rozprawy	4
1.4. Teza rozprawy.....	4
1.5. Metodologia badań.....	4
2. Proponowana metoda syntezy widoków wirtualnych	5
2.2. Przetwarzanie pojedynczego widoku rzeczywistego	6
2.3. Łączenie sąsiednich pośrednich widoków wirtualnych	7
2.4. Wypełnianie odsłonieć	10
2.5. Uzupełnianie widoku wirtualnego	12
2.6. Podsumowanie.....	12
3. Proponowane techniki zwiększające jakość syntezy widoków wirtualnych	13
3.1. Filtracja przerzutowanych punktów	13
3.2. Rzutowanie punktów zapewniające ciągłość obiektów	17
3.3. Korekcja niespójności barwnej widoku wirtualnego.....	20
4. Ocena efektywności zaproponowanej metody syntezy.....	25
4.1. Ocena jakości syntezy	26
4.2. Pomiar czasu obliczeń	27
5. Synteza czasu rzeczywistego	28
6. Podsumowanie	33
6.1. Uzasadnienie tezy rozprawy.....	33
6.2. Przeprowadzone badania	34
6.3. Oryginalne osiągnięcia	35
7. Dorobek naukowy autora.....	36

1. WPROWADZENIE

1.1. SYNTEZA WIDOKÓW WIRTUALNYCH DLA SWOBODNEJ NAWIGACJI

Rozprawa dotyczy syntezy widoków wirtualnych dla celów systemów swobodnej nawigacji, a więc systemów, w których użytkownik ma możliwość wirtualnej nawigacji w zarejestrowanej scenie. System swobodnej nawigacji może znaleźć wiele zastosowań, jak chociażby wymienione w pracy [8] wydarzenia kulturalne (np. spektakle teatralne), instruktaże i kursy, na przykład tańca, naprawy samochodu czy też pierwszej pomocy. Przede wszystkim jednak, największy potencjał wydaje się mieć transmisja wydarzeń sportowych takich jak piłka nożna, zespołowe sporty halowe (np. siatkówka lub koszykówka) bądź też sporty indywidualne jak szermierka czy judo [8, 23, 25].

W takim systemie widz sam może wybierać swój punkt widzenia i kierunek patrzenia [45]. W celu umożliwienia widzowi wirtualnego przemieszczania się, pula punktów widzenia jaką dysponuje widz nie może być ograniczona wyłącznie do widoków zarejestrowanych przez kamery systemu wielokamerowego. Aby ruch widza był płynny, należy stworzyć (**zsyntezować**) dodatkowe widoki (**widoki wirtualne**). Widoki te są tworzone na podstawie widoków rzeczywistych, czyli widoków zarejestrowanych przez rzeczywiste kamery systemu wielokamerowego. Co więcej, oczekuje się, że różnica jakości pomiędzy widokami rzeczywistymi i wirtualnymi będzie jak najmniejsza, by widz nie był w stanie stwierdzić, czy ogląda obrazy zarejestrowane przez kamery, czy też obrazy syntetyczne.



Rysunek 1.1. Ilustracja idei swobodnej nawigacji; kolorem czarnym oznaczono rzeczywiste kamery, pomarańczowym – kamerę wirtualną, z której użytkownik systemu ogląda obraz i której pozycję może on w dowolny sposób zmieniać

Do przeprowadzenia syntezy widoków wirtualnych wymagane są widoki rzeczywiste oraz informacja przestrzenna o scenie (informacja o rozmieszczeniu obiektów w scenie, najczęściej reprezentowana poprzez mapy głębi – reprezentacja „obraz wielowidokowy + głębia”, MVD) i samym systemie wielokamerowym (informacja o rozmieszczeniu i charakterystyce optycznej kamer).

1.2. PROBLEMY CHARAKTERYSTYCZNE DLA RZADKICH SYSTEMÓW WIELOKAMEROWYCH

Synteza widoków wirtualnych w rzadkich systemach wielokamerowych jest zdecydowanie trudniejsza niż w przypadku systemów gęstych, dotychczas najczęściej rozpatrywanych w literaturze. Zwiększona trudność spowodowana jest dużą odległością pomiędzy poszczególnymi kamerami (np. 1,5 metra [23]) lub dużym kątem pomiędzy ich osiami optycznymi (wynoszącym np. 10° [38], 15° [30], a w przypadku kamer umieszczonych na dwóch prostopadłych ścianach nawet 90° [23]). Skutkuje to trzema podstawowymi problemami charakterystycznymi dla rzadkich systemów wielokamerowych:

1. Mniejszy obszar sceny jest widoczny jednocześnie w różnych widokach rzeczywistych, co skutkuje powstawaniem odsłonień w widoku wirtualnym.
2. Ograniczona rozdzielczość obrazu skutkująca powstawaniem pęknięć (ang. *cracks*) – nieciągłości obiektów w widoku wirtualnym.
3. Nielambertowskie odbicia światła skutkujące różną barwą tych samych obiektów w różnych widokach rzeczywistych.

Rozwiązanie trzech wymienionych problemów jest kluczowe dla opracowania metody syntezy widoków wirtualnych dla zastosowań w swobodnej nawigacji w rzadkich systemach wielokamerowych.

1.3. CEL ROZPRAWY

Celem pracy jest rozwiązanie problemów utrudniających syntezę dobrej jakości widoków wirtualnych w praktycznych systemach telewizji swobodnego punktu widzenia i opracowanie efektywnej metody syntezy widoków wirtualnych dla celów swobodnej nawigacji. Niniejsza rozprawa ma więc na celu poszerzenie stanu wiedzy w tej dziedzinie i umożliwienie rozwoju tego rodzaju systemów. W rozprawie zaprezentowano nową, wydajną metodę syntezy widoków wirtualnych dostosowaną do rzadkiego rozmieszczenia kamer.

1.4. TEZA ROZPRAWY

Możliwe jest opracowanie wydajnej metody syntezy widoków wirtualnych dobrej jakości poprzez wykorzystanie dalszych widoków rzeczywistych, filtrację krawędzi w widokach rzeczywistych i odpowiadających im mapach głębi oraz korekcję niespójności barwnej widoku wirtualnego.

1.5. METODOLOGIA BADAŃ

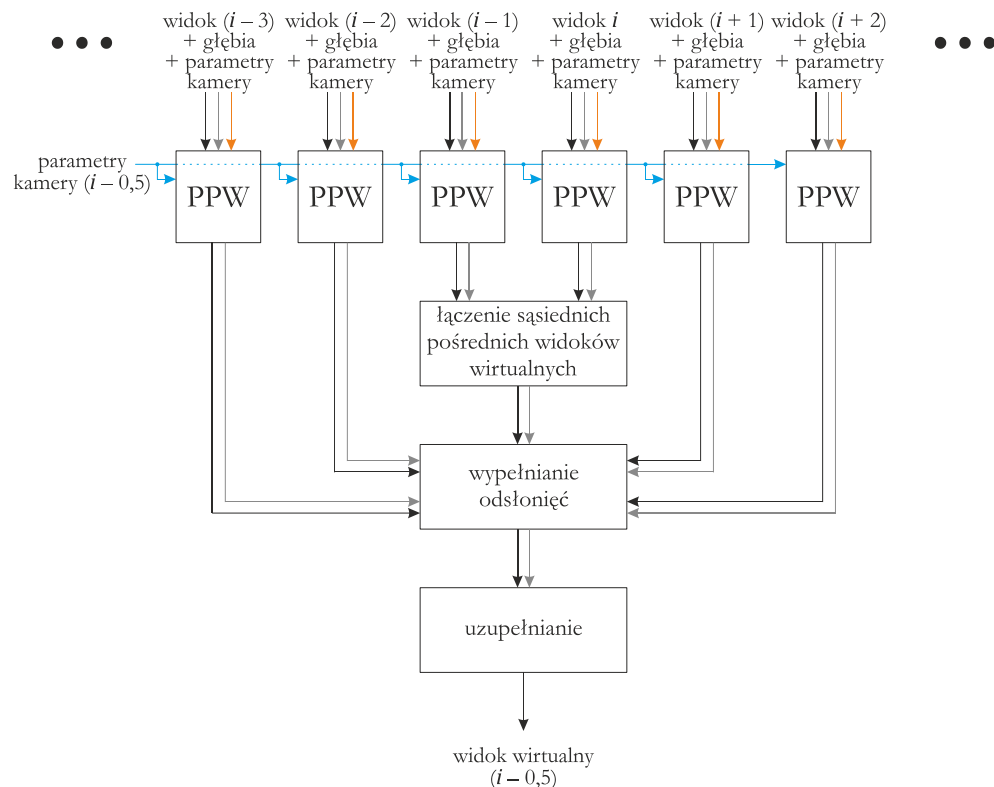
Jakość widoków wirtualnych syntezowanych przy użyciu zaproponowanej przez autora rozprawy i opisywanej w rozprawie metody MVS (*MultiView Synthesis*) była badana z wykorzystaniem dwunastu zróżnicowanych sekwencji testowych. Ocenie została poddana jakość mierzona w sposób zarówno obiektywny, jak i subiektywny. Do zmierzenia jakości obiektywnej użyto dwóch miar: PSNR

i SSIM. W celu estymacji jakości syntezowano widok wirtualny w pozycji widoku rzeczywistego, umożliwiając porównanie zsyntezowanego widoku z widokiem odniesienia. Do subiektywnego pomiaru jakości użyto metody porównywania dwóch obrazów – PC (*Pair Comparison*) rekomendowanej przez ITU-T (ITU-T P.910). W testach wzięło udział 44 uczestników.

W celu oceny opracowanej przez autora metody syntezy widoków wirtualnych MVS, jakość widoków syntezowanych przy użyciu metody MVS została porównana z jakością widoków wirtualnych syntezowanych przy użyciu metody odniesienia zaimplementowanej w rozwijanym przez grupę ekspertów MPEG (ISO/IEC JTC 1/SC29/WG 11) oprogramowaniu VSRS (*View Synthesis Reference Software*) w najnowszej wersji dostępnej w trakcie realizacji pracy, czyli wersji 4.2.

2. PROPONOWANA METODA SYNTEZY WIDOKÓW WIRTUALNYCH

Zaproponowana przez autora metoda syntezy widoków wirtualnych (MVS, *MultiView Synthesis*) wykorzystuje reprezentację MVD, a więc jej danymi wejściowymi są widoki rzeczywiste wraz z odpowiadającymi im mapami głębi oraz parametry kamer rzeczywistych i wirtualnej. Algorytm składa się z czterech głównych etapów przedstawionych na rysunku 2.1. Podstawową przewagą opracowanej metody nad metodami znanymi z literatury zapewnia trzeci etap, czyli wypełnianie odsłoneń.

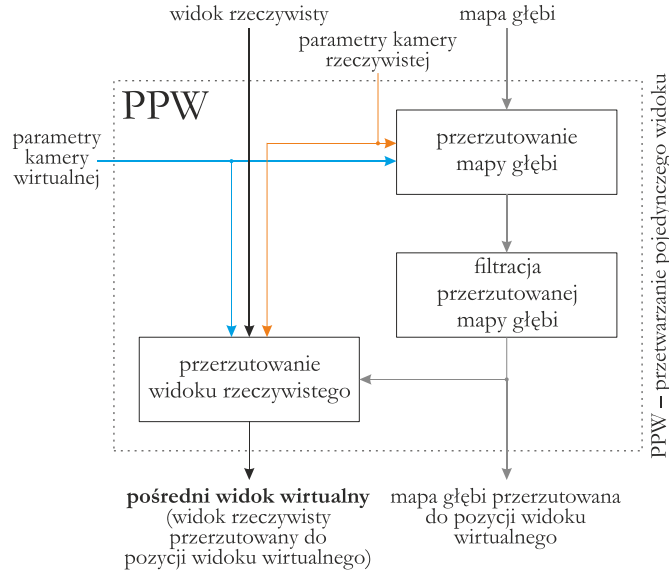


Rysunek 2.1. Ogólny schemat proponowanej metody syntezy widoków wirtualnych. Schemat bloku przetwarzania pojedynczego widoku (PPW) przedstawiony został na rysunku 2.2

2.1. PRZETWARZANIE POJEDYNCZEGO WIDOKU RZECZYWISTEGO

W pierwszym etapie zaproponowanej metody syntezy widoków wirtualnych wszystkie widoki rzeczywiste przetwarzane są niezależnie. Co więcej, przetwarzane są one w identyczny sposób, niezależnie od tego, czy są to widoki sąsiednie czy też dalsze. Sposób przetwarzania pojedynczego widoku (PPW) przedstawiono na rysunku 2.2.

Dane wejściowe stanowi widok rzeczywisty wraz z odpowiadającą mu mapą głębi, a także parametry (zewnętrzne i wewnętrzne) kamery rzeczywistej i wirtualnej. Na wyjściu otrzymywany jest pośredni widok wirtualny i odpowiadająca mu mapa głębi, zawierające informację przerzutowaną wyłącznie z jednego widoku. Rzutowanie przeprowadzane jest w dwóch krokach: w pierwszym kroku wyznaczana jest wirtualna mapa głębi, w kroku drugim mapa ta jest używana do rzutowania próbek barwy z widoku rzeczywistego.



Rysunek 2.2. Schemat przetwarzania informacji z pojedynczego widoku rzeczywistego wraz z odpowiadającą mu mapą głębi

W pierwszym kroku tworzona jest wirtualna mapa głębi. Dla każdego punktu rzeczywistej mapy głębi tworzony jest wektor $[x_R \ y_R \ z_R \ 1]^T$ zawierający położenie danego punktu w mapie głębi (x_R, y_R) i jego głębię z_R . Następnie wektor ten jest mnożony zgodnie z:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{P}_V \cdot \mathbf{P}_R^{-1} \cdot \begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix}, \quad (2.1)$$

$$x_V = \frac{x'}{w'}, \quad y_V = \frac{y'}{w'}, \quad z_V = \frac{z'}{w'}, \quad (2.2)$$

gdzie $\mathbf{P}_V$ jest macierzą projekcji widoku wirtualnego (łączącą parametry wewnętrzne i zewnętrzne), a $\mathbf{P}_R$ – macierzą projekcji widoku rzeczywistego. Dla każdego analizowanego punktu otrzymywany jest wektor zawierający współrzędne określające pozycję odpowiadającego punktu w widoku wirtualnym wraz z jego głębią $[x_V \ y_V \ z_V \ 1]^T$. Współrzędne punktu (x_V, y_V) są następnie zaokrąglane zgodnie z:

$$(\widetilde{x}_V, \widetilde{y}_V) = ([x_V], [y_V]), \quad (2.3)$$

gdzie operator $[\cdot]$ oznacza zaokrąglenie do najbliższej liczby całkowitej.

Podczas rzutowania punktów brana pod uwagę jest kwestia przysyłania się obiektów. Jeżeli w dany punkt wirtualnej mapy głębi przerzutowane zostanie więcej punktów niż jeden, wybierany jest ten spośród nich, którego wartość głębi z_V jest najmniejsza. Tym samym wybierany jest punkt reprezentujący najbliższy obiekt. Następnie dokonywana jest filtracja wirtualnej mapy głębi. Wartości przefiltrowanej wirtualnej mapy głębi są używane do przerzutowania informacji o barwie poszczególnych punktów tworzonego widoku wirtualnego. W tym celu dla każdego punktu wirtualnej mapy głębi tworzony jest wektor zawierający jego pozycję (x_V, y_V) i głębię z_V . Następnie zgodnie z:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \mathbf{P}_R \cdot \mathbf{P}_V^{-1} \cdot \begin{bmatrix} x_V \\ y_V \\ z_V \\ 1 \end{bmatrix}, \quad (2.4)$$

$$x_R = \frac{x'}{w'}, \quad y_R = \frac{y'}{w'}, \quad (2.5)$$

$$(\widetilde{x}_R, \widetilde{y}_R) = ([x_R], [y_R]) \quad (2.6)$$

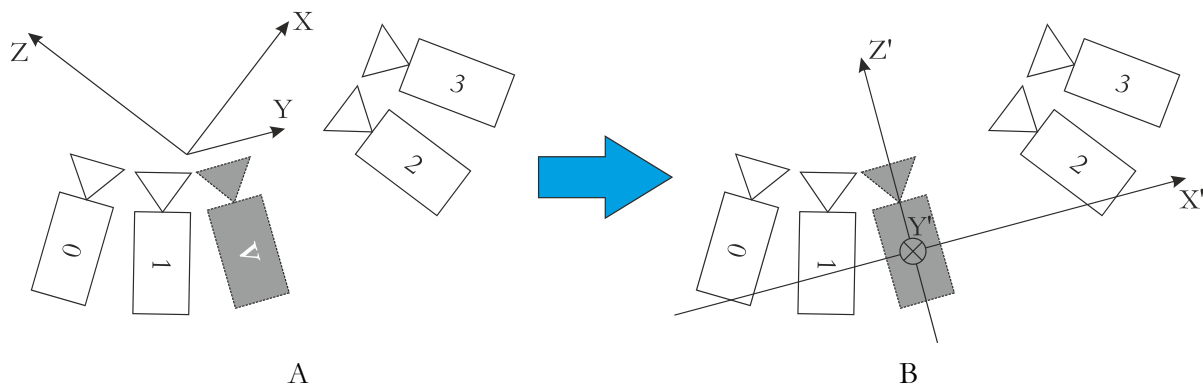
wyliczana jest pozycja w widoku rzeczywistym $(\widetilde{x}_R, \widetilde{y}_R)$ punktu, którego barwa zostanie skopiowana do analizowanego punktu danego pośredniego widoku wirtualnego.

2.2. ŁĄCZENIE SĄSIEDNICH POŚREDNICH WIDOKÓW WIRTUALNYCH

Po stworzeniu pośrednich widoków wirtualnych dla wszystkich widoków rzeczywistych, dokonywane jest łączenie pośrednich widoków wyznaczonych dla najbliższego lewego i najbliższego prawego widoku. Jako że informacja o sąsiedztwie widoków nie jest dana, należy wyznaczyć, które dwa widoki zostaną połączone.

W celu wyboru najbliższych widoków rzeczywistych opracowano uniwersalny sposób wyboru sąsiednich kamer rzeczywistych działający poprawnie dla dowolnego rozmieszczenia kamer [3]. Pierwszym krokiem zaproponowanej techniki jest zmiana układu współrzędnych z globalnego (rysunek 2.3A) do lokalnego układu współrzędnych kamery wirtualnej (rysunek 2.3B). Układ ten ma swój początek na środku przetwornika kamery wirtualnej i skierowanym jest tak, by oś Z' pokrywała

się z osią optyczną kamery wirtualnej, a osie X' i Y' były równoległe do, odpowiednio, poziomej i pionowej osi przetwornika kamery wirtualnej.



Rysunek 2.3. Przekształcenie globalnego układu współrzędnych (A) do układu współrzędnych kamery wirtualnej (B)

Wektor translacji kamer rzeczywistych w przekształconym układzie współrzędnych otrzymywany jest ze wzoru:

$$\mathbf{T}'_i = \mathbf{R}_V \cdot (\mathbf{T}_i - \mathbf{T}_V), \quad (2.7)$$

gdzie $\mathbf{R}_V$ jest macierzą rotacji kamery wirtualnej, $\mathbf{T}_i$ – wektorem translacji i -tej kamery w globalnym układzie współrzędnych, a $\mathbf{T}_V$ wektorem translacji kamery wirtualnej w globalnym układzie współrzędnych.

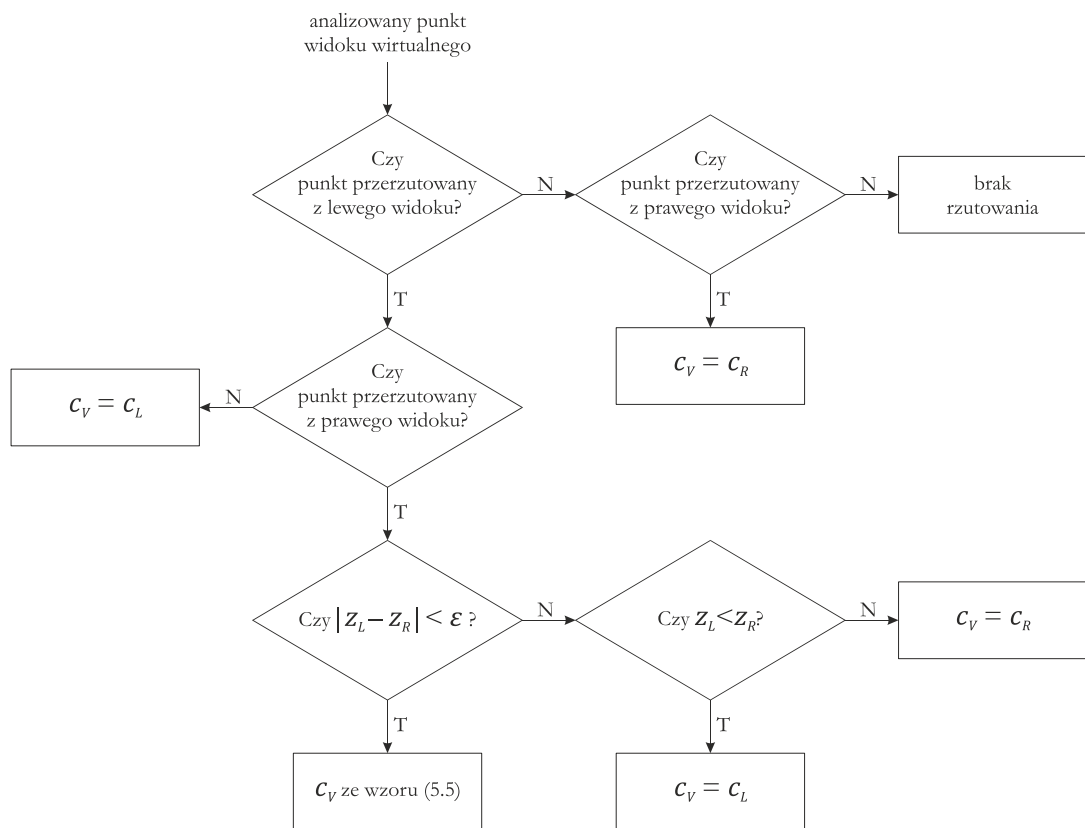
Następnie, osobno dla kamer cechujących się dodatnią i ujemną wartością składowej translacji t_x , wyznaczana jest odległość euklidesowa pomiędzy kamerą a początkiem układu współrzędnych. Ta z kamer o dodatniej wartości składowej t_x , dla której odległość od początku układu współrzędnych jest najmniejsza, jest najbliższą prawą kamerą rzeczywistą. Lewa najbliższa kamera rzeczywista jest wybierana spośród kamer cechujących się ujemną wartością składowej translacji t_x – i jest to ta spośród nich, której odległość od środka układu współrzędnych jest najmniejsza.

Po wybraniu najbliższej lewej i prawej rzeczywistej kamery następuje proces łączenia wyznaczonych dla nich pośrednich widoków wirtualnych. Po połączeniu, każdy punkt połączonego widoku wirtualnego zawierać będzie barwę przerzutowaną z widoku lewego, z widoku prawego albo wymieszaną barwę z obu widoków. Schemat łączenia widoków przedstawiono na rysunku 2.4.

Dla każdego punktu sprawdzane jest, czy został on przerzutowany z lewego lub prawego widoku rzeczywistego. Jeżeli został on przerzutowany wyłącznie z widoku lewego, do analizowanego punktu widoku wirtualnego zostanie skopiowana barwa z lewego pośredniego widoku wirtualnego. Analogicznie, jeżeli punkt został przerzutowany tylko z prawego widoku rzeczywistego, do

analizowanego punktu widoku wirtualnego skopiowana zostanie barwa z prawego pośredniego widoku wirtualnego.

Jeżeli punkt został przerzutowany z obu widoków rzeczywistych, porównywana jest wartość głębi przerzutowana z obu widoków. Jeżeli wartości głębi znacznie się różnią (a więc punkty przerzutowane z dwóch widoków reprezentują różne obiekty), do analizowanego punktu skopiowana zostanie barwa z tego widoku rzeczywistego, z którego przerzutowana została mniejsza wartość głębi – a więc barwa punktu reprezentującego bliższy obiekt. W przeciwnym wypadku, a więc jeżeli wartość głębi przerzutowanej z obu widoków rzeczywistych jest podobna, z dużym prawdopodobieństwem punkt reprezentuje ten sam obiekt w obu widokach, a więc jego barwę należy zmieszać.



Rysunek 2.4. Schemat łączenia pośrednich widoków wirtualnych:

c_v – barwa punktu w połączonym widoku wirtualnym; c_L , c_R – barwa punktu w lewym i prawym pośrednim widoku wirtualnym; z_L , z_R – głębia punktu przerzutowanego z lewego i prawego widoku rzeczywistego (odległość między punktem w przestrzeni i płaszczyzną przetwornika kamery wirtualnej)

W celu uniknięcia skokowej zmiany barwy obiektów przy zmianie punktu widzenia (zmianie pozycji kamery wirtualnej), zmieszana barwa punktu powinna zmieniać się w sposób proporcjonalny do odległości pomiędzy kamerą wirtualną i kamerami rzeczywistymi. Mieszanie barwy przerzutowanej z lewego (c_L) i barwy z prawego widoku rzeczywistego (c_R) zrealizowano więc poprzez średnią ważoną wyrażoną wzorem:

$$c_V(x, y) = \frac{c_L(x, y) \cdot \frac{d(R, V)}{z_L(x, y)} + c_R(x, y) \cdot \frac{d(L, V)}{z_R(x, y)}}{\frac{d(R, V)}{z_L(x, y)} + \frac{d(L, V)}{z_R(x, y)}}, \quad (2.8)$$

gdzie $c_L(x, y)$ i $c_R(x, y)$ jest barwą położonego w pozycji (x, y) punktu przerzutowanego z lewego i prawego widoku rzeczywistego, $z_L(x, y)$ i $z_R(x, y)$ jego głębcią (odległością punktu w przestrzeni do płaszczyzny przetwornika kamery wirtualnej), a $d(L, V)$ i $d(R, V)$ odległością euklidesową pomiędzy kamerą wirtualną i – odpowiednio – lewą i prawą kamerą rzeczywistą. Barwa punktu ważona jest zarówno odległością pomiędzy kamerą wirtualną i poszczególnymi kamerami rzeczywistymi (im mniejsza odległość, tym większa waga), jak i wartością głębci (im mniejsza głębć, tym większa waga).

2.3. WYPEŁNIANIE ODSŁONIEŃ

Widok wirtualny stworzony na podstawie dwóch najbliższych widoków rzeczywistych zawiera obszary odsłonięte, czyli obszary, które nie były widoczne ani w najbliższym lewym, ani w najbliższym prawym widoku rzeczywistym. Odsłonięcia są znanym problemem charakterystycznym dla syntezy widoków wirtualnych. W metodach opisanych w literaturze najczęściej wypełniane są one na podstawie otoczenia, co skutkuje błędami i pogorszeniem jakości synteżowanych widoków.

W zaproponowanej w ramach rozprawy metodzie syntezy widoków wirtualnych obszary odsłonięte są wypełniane przy wykorzystaniu informacji z pozostałych widoków rzeczywistych.

Zgodnie ze schematem przedstawionym na rysunku 2.1, do wypełniania odsłonieć używane są pośrednie widoki wirtualne (wraz z mapami głębci) przerzutowane z dalszych widoków rzeczywistych. Podczas wypełniania odsłonieć nie jest więc konieczne wykonywanie kolejnych rzutowań.

Podczas wypełniania analizowane są wyłącznie te punkty połączonego widoku wirtualnego, które nie zostały przerzutowane z widoków sąsiednich. Dla każdego z nich sprawdzany jest podstawowy warunek – czy punkt został przerzutowany z co najmniej jednego widoku rzeczywistego. Jeżeli powyższy warunek nie jest spełniony, dany punkt nie jest wypełniany. Jeżeli punkt został przerzutowany wyłącznie z jednego widoku, jego barwa jest wprost kopiowana. W przypadku, gdy punkt został przerzutowany z większej liczby widoków, porównywana jest wartość przerzutowanej głębci dla każdego z nich. Spośród wszystkich kandydatów wybierany jest zbiór punktów oznaczany jako P . Zbiór ten zawiera punkty reprezentujące najbliższy obiekt (mające najmniejszą wartość głębci).

Jeżeli zbiór P zawiera wyłącznie jeden punkt (a więc głębć w jednym widoku była zdecydowanie mniejsza, niż w pozostałych widokach), do widoku wirtualnego kopiowana jest barwa punktu z danego widoku. W innych przypadkach barwa punktu w wypełnianym obszarze odsłoniętym wyznaczana jest poprzez mieszanie barwy przerzutowanej z kilku widoków rzeczywistych. Tak jak w przypadku mieszania barw z najbliższych widoków (2.8), należy zapewnić łagodną zmianę barwy

obiektów podczas zmiany punktu widzenia. Tym samym, mieszanie barw z różnych widoków jest realizowane poprzez średnią ważoną wyznaczaną ze wzoru:

$$c_V(x, y) = \frac{\sum_{i \in P} c_i(x, y) \cdot \frac{D - d(i, V)}{z_i(x, y)}}{\sum_{i \in P} \frac{D - d(i, V)}{z_i(x, y)}}, \quad (2.9)$$

gdzie $c_i(x, y)$ jest barwą analizowanego punktu dla i -tego widoku, $z_i(x, y)$ jego głębią, $d(i, V)$ odległością między kamerą wirtualną a i -tą kamerą rzeczywistą, a D jest sumą odległości wszystkich kamer rzeczywistych do kamery wirtualnej:

$$D = \sum_{i \in P} d(i, V). \quad (2.10)$$

W tabeli 2.1 przedstawiono jakość widoków wirtualnych syntezowanych z zastosowaniem zaproponowanej techniki wypełniania odsłoneń z użyciem informacji z dalszych widoków. Jakość tę zestawiono z jakością widoków, w których odsłonecia były uzupełniane na podstawie otoczenia.

Tabela 2.1. Porównanie wypełniania odsłoneń z użyciem informacji z dalszych widoków i uzupełniania; PSNR liczony dla całego widoku wirtualnego, średnia dla wszystkich widoków; dla każdej sekwencji pogrubiono największą wartość PSNR

Sekwencja	PSNR [dB]	
	Bez wypełniania odsłoneń	Wypełnianie odsłoneń
BBB Butterfly	32,48	33,60
BBB Flowers	24,76	25,87
Poznan_Blocks	25,12	25,07
Poznan_Blocks2	29,35	31,23
Poznan_Fencing2	30,46	30,63
Poznan_Service2	25,19	25,47
Ballet	30,20	30,21
Breakdancers	31,96	31,97
Soccer Arc	22,65	22,68
Soccer Linear	34,20	34,44
Poznan_Carpark	33,74	33,80
Poznan_Street	35,44	35,46
Średnio	29,63	30,04

Jak pokazano, użycie informacji z dalszych widoków rzeczywistych do uzupełniania odsłonięć pozwala na osiągnięcie lepszej jakości syntezy (wyluczając odznaczającą się błędnymi mapami głębi sekwencję *Poznan_Blocks*) – tym lepszej, im większa była łączna powierzchnia odsłonięć i im bardziej skomplikowany fragment sceny był przysłonięty w sąsiednich widokach.

2.4. UZUPEŁNIANIE WIDOKU WIRTUALNEGO

Zsyntezowany widok wirtualny, pomimo wypełnienia obszarów odsłoniętych informacją z innych kamer, wciąż zawiera obszary zawierające punkty nieprzerzutowane – niewidoczne w żadnym z rzeczywistych widoków. Powierzchnia takich obszarów nie jest jednak duża. Tym samym zastosowano uzupełnianie dyfuzyjne, w którym uzupełniane obszary wypełniane są na podstawie bezpośredniego otoczenia.

Główną zaletą zastosowanego rozwiązania jest uwzględnianie informacji o głębi punktów, co skutkuje uzupełnianiem odsłonięć barwą tła (przysłoniętego w widokach rzeczywistych a odsłoniętego w syntezyowanym widoku wirtualnym), a nie bliższych obiektów. Tym samym, uzupełnianie nie zwiększa sztucznie powierzchni bliskich obiektów (nie „rozlewa” ich w przylegających do nich uzupełnianych obszarach).

2.5. PODSUMOWANIE

W rozdziale 2. opisano proponowaną metodę syntezy widoków wirtualnych z wypełnianiem odsłonięć przy użyciu informacji z dalszych widoków rzeczywistych. Przedstawiono, jak z widoków rzeczywistych i odpowiadających im map głębi stworzyć widok wirtualny.

W dalszych rozdziałach rozprawy skupiono się na opracowanych technikach dodatkowo poprawiających jakość syntezyowanych widoków wirtualnych:

- technice filtracji rozmytych krawędzi w widokach rzeczywistych i mapach głębi,
- sposobie rzutowania punktów do widoku wirtualnego, wymuszającego ciągłość obiektów,
- sposobie wstępnego zwiększania rozdzielczości widoków rzeczywistych i map głębi,
- technikach korekcji niespójności barwnej widoku wirtualnego.

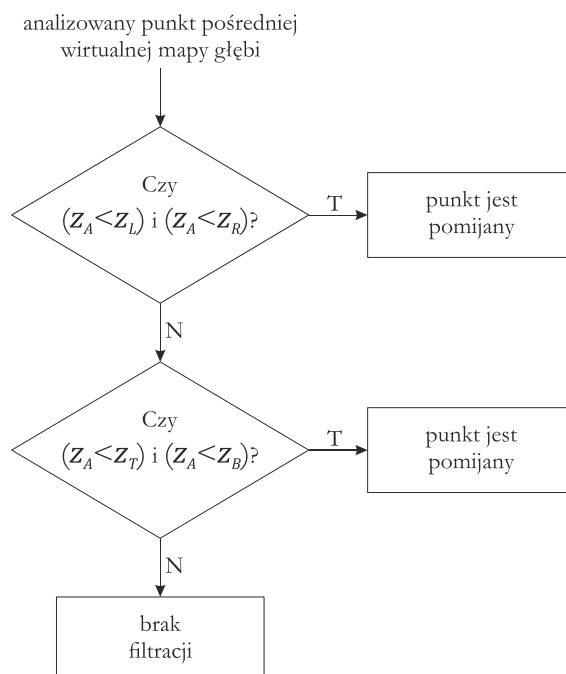
3. PROPONOWANE TECHNIKI POPRAWIAJĄCE JAKOŚĆ SYNTEZOWANYCH WIDOKÓW WIRTUALNYCH

3.1. FILTRACJA PRZERZUTOWANYCH PUNKTÓW

3.1.1. REDUKCJA BŁĘDÓW SPOWODOWANYCH ROZMYCIEM KRAWĘDZI W RZECZYWISTYCH MAPACH GŁĘBI

Wirtualna mapa głębi z tego rodzaju błędami, których powstanie mogło wynikać z rozmycia krawędzi w rzeczywistych mapach głębi, została pokazana na rysunku 3.2A. Opisywane błędy są widoczne jako małe jasne obszary na tle ciemniejszego, mającego większą głębokość otoczenia. Eliminacja takich obszarów jest kluczową kwestią dla osiągnięcia zadowalającej jakości subiektywnej podczas wirtualnej nawigacji, co jest to spowodowane małą wartością głębokości tych obszarów. Niewyeliminowanie błędów spowodowanych rozmytymi krawędziami skutkuje tym, iż niepoprawnie zsintezowane obszary znajdują się blisko widza (mają mniejszą głębokość niż otoczenie), a więc podczas wirtualnego poruszania się w scenie wydają się szybko przemieszczać, skupiając na sobie uwagę widza.

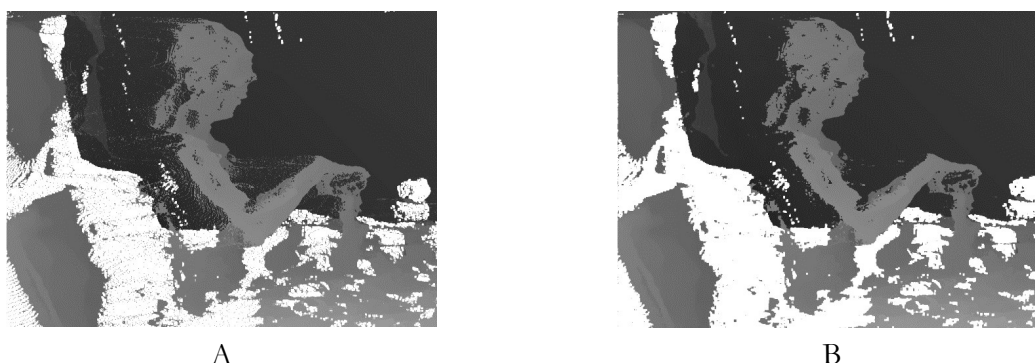
Do usunięcia opisywanych błędów stworzono algorytm (rys. 3.1), w którym głębokość każdego punktu wirtualnej mapy głębi porównywana jest z głębokością jego sąsiadów. Opracowany algorytm pozwala na lepszą eliminację błędnych obszarów, niż stosowana i opisywana w literaturze filtracja medianowa.



Rysunek 3.1. Schemat eliminacji rozmytych krawędzi rzeczywistych map głębi:
 z_A – głębokość analizowanego punktu; z_L , z_R – głębokości punktów z lewej i prawej strony analizowanego punktu;
 z_T , z_B – głębokości punktu powyżej i poniżej analizowanego punktu

W celu eliminacji jak największej liczby punktów z błędną głębią, głębia każdego punktu jest porównywana z głębią punktów sąsiadujących zarówno w pionie, jak i w poziomie. Jeżeli oba sąsiadujące punkty charakteryzują się większą głębią niż punkt analizowany, wartość głębi dla tego punktu jest usuwana, a on sam jest traktowany jako punkt nieprzerzutowany. Wynik działania proponowanej techniki pokazano na rys. 3.2B.

Zaprezentowane podejście ma jedną wadę: podczas eliminacji obiektów z niepoprawną głębią usunięte mogą zostać również istniejące obiekty charakteryzujące się niewielkim obszarem i mniejszą głębią niż głębia punktów w jego otoczeniu. Jednakże, eliminacja szybko przemieszczających się w scenie nieistniejących w rzeczywistości obiektów rekompensuje ewentualny brak drobnych szczegółów syntezy widoku [11].



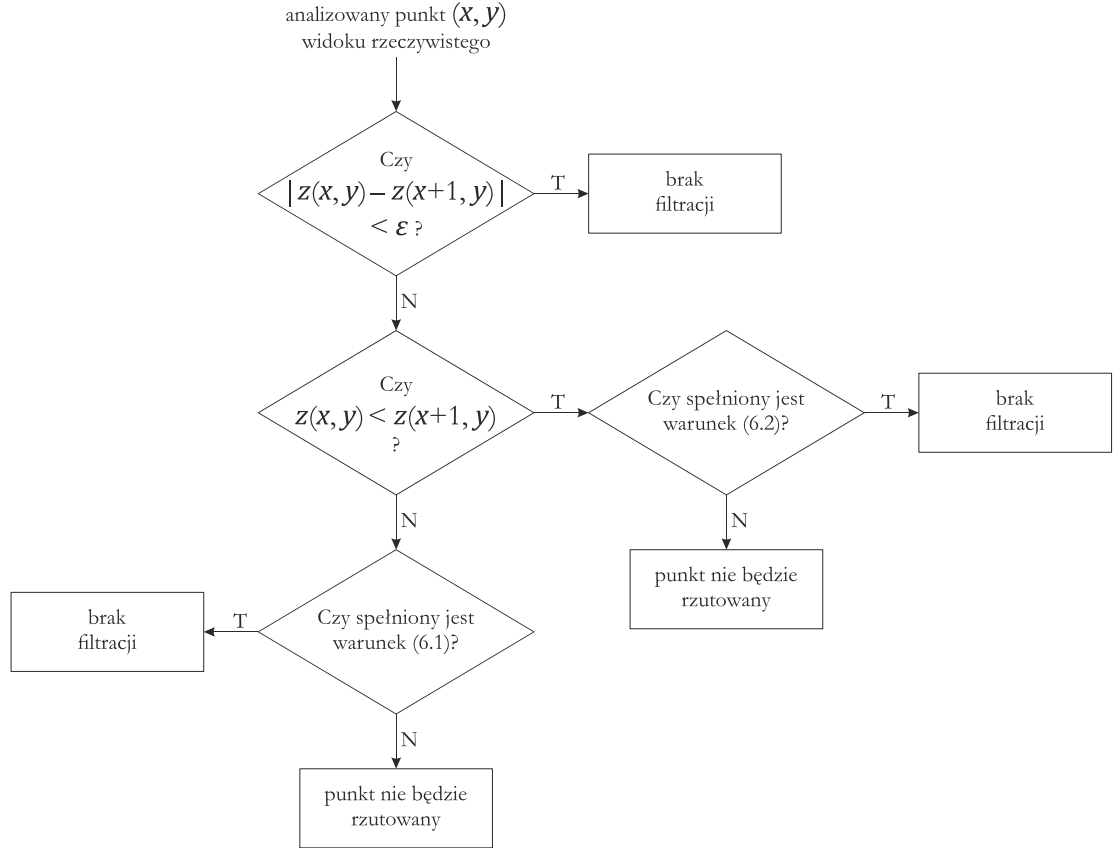
Rysunek 3.2. Nieistniejące obiekty w wirtualnej mapie głębi przed (A) i po (B) filtracji (sekwencja *Poznan_Blocks*, synteza z widoku 7 do widoku 4)

3.1.2. ELIMINACJA EFEKTÓW ROZMYTYCH KRAWĘDZI W WIDOKU RZECZYWISTYM

Rozmycie krawędzi w widokach rzeczywistych, a także przesunięcie krawędzi pomiędzy widokiem rzeczywistym a mapą głębi może skutkować pojawianiem się w widoku wirtualnym nieistniejących w rzeczywistej scenie krawędzi. Błąd tego rodzaju zaprezentowano na rysunku 3.4A. Opisywany efekt jest najbardziej widoczny, kiedy krawędź bliższego obiektu rzutowana jest na jednolity obszar, na przykład tło. Niemniej jednak, nieistniejące krawędzie widoczne są w widoku wirtualnym również w przypadku bardziej złożonych obszarów (na przykład obszarów zawierających krawędzie).

W celu usuwania opisywanych błędów zaproponowano technikę, w której punkty znajdujące się w okolicach krawędzi obiektów (a więc punkty, w których otoczeniu występuje skokowa zmiana głębi) są przetwarzane w inny sposób, niż pozostałe punkty. W odróżnieniu od technik opisywanych w literaturze, w zaproponowanym podejściu [2] dodatkowo sprawdzane są gradienty w sąsiedztwie analizowanego punktu w widoku rzeczywistym, umożliwiając rzutowanie punktów znajdujących się przy krawędziach obiektów w sytuacji, gdy ich barwa jest poprawna.

Dla każdego punktu widoku rzeczywistego estymowane są 3 gradienty, osobno dla każdej składowej Y , Cb i Cr . Gradient szacowany jest poprzez różnicę danej składowej dla analizowanego punktu i punktu następnego (po prawej stronie lub poniżej). Następnie dla każdego punktu tła graniczącego z obszarem o mniejszej głębi sprawdzane jest, czy norma jego gradientu jest większa, niż średnia norm gradientów trzech punktów sąsiadujących. Schemat zaproponowanego algorytmu przedstawiono na rysunku 3.3.



Rysunek 3.3. Schemat eliminacji rozmytych krawędzi w widokach rzeczywistych

Aby punkt tła graniczący lewostronnie z bliższym obiektem (mający po lewej stronie punkt o mniejszej wartości głębi) był rzutowany do widoku wirtualnego, musi spełniać warunek:

$$|c(x+1, y) - c(x+2, y)| < \frac{1}{3} \sum_{i=2}^4 |c(x+i, y) - c(x+i+1, y)|, \quad (3.1)$$

gdzie $c(x, y)$ oznacza wartość analizowanej składowej barwnej w punkcie (x, y) . Analogiczny warunek skonstruowany został dla przypadku punktu tła graniczącego z bliższym obiektem prawostronnie (mającego po prawej stronie punkt o mniejszej wartości głębi), a także pionowej składowej gradientu.

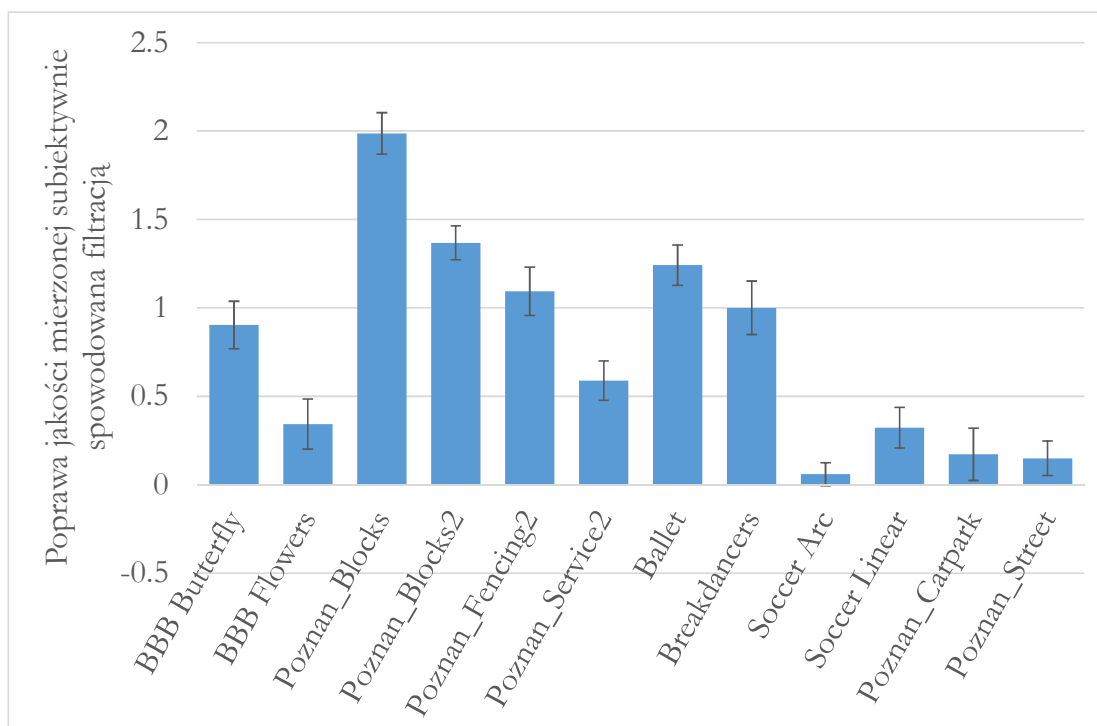
Efekt zastosowania zaproponowanej techniki filtracji przedstawiono na rysunku 3.4B.



Rysunek 3.4. Fragment widoku wirtualnego zsyntezowanego bez zastosowania filtracji (A) i z filtracją (B)

3.1.3. WPŁYW FILTRACJI NA JAKOŚĆ SYNTEZY

Aby zbadać wpływ zaproponowanych technik filtracji na subiektywne postrzeganie jakości syntezy widoku przeprowadzono również testy subiektywne, podczas których uczestnicy porównywali jakość widoków syntezy bez użycia filtracji i z zastosowaniem filtracji (rys. 3.5).



Rysunek 3.5. Poprawa mierzonej w sposób subiektywny jakości syntezy widoków wynikająca z zastosowania dodatkowej filtracji; dla każdej sekwencji oznaczono 95% przedział ufności; wartości dodatnie oznaczają, że jakość widoków syntezy przy użyciu MVS jest lepsza, niż w przypadku VSRS; metoda PC, skala: [-3, 3]

Jak zaprezentowano, dla niemal wszystkich sekwencji testowych poprawa jakości mierzonej w sposób subiektywny jest statystycznie istotna. Wyjątkiem jest sekwencja *Soccer Arc*, dla której zaproponowana technika filtracji również powoduje poprawę jakości, jednakże na tyle małą, iż na podstawie przeprowadzonych badań jakości subiektywnej nie można stwierdzić, że poprawa ta jest statystycznie istotna.

Największą poprawę otrzymano dla sekwencji *Poznan_Blocks*, która to charakteryzuje się rozmytymi krawędziami w rzeczywistych mapach głębi. Tym samym, bez dodatkowej filtracji w widoku wirtualnym pojawiają się liczne błędy degradujące jakość subiektywną.

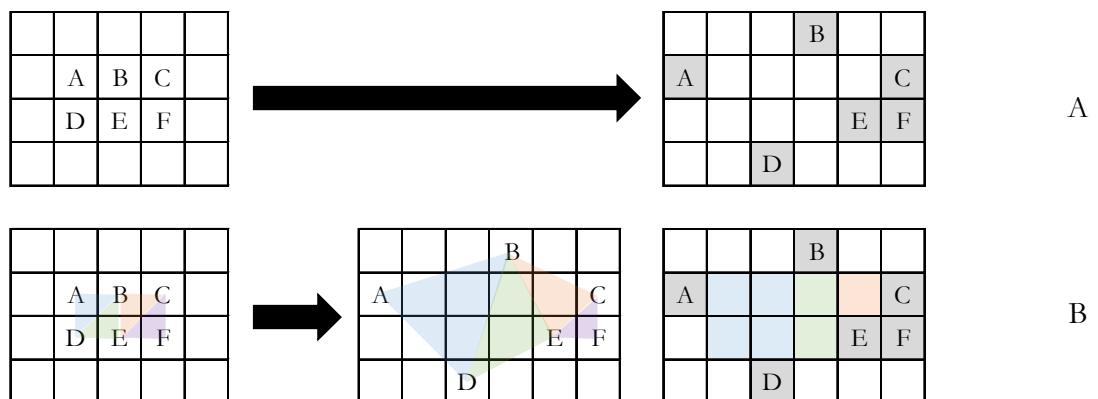
3.2. RZUTOWANIE PUNKTÓW ZAPEWNIĄCE CIĄGŁOŚĆ OBIEKTÓW

3.2.1. ALGORYTM

W typowym podejściu podczas tworzenia wirtualnej mapy głębi wszystkie punkty z rzeczywistych map głębi są rzutowane do wirtualnej mapy głębi w sposób niezależny. Tym samym, jeżeli dwa sąsiadujące w rzeczywistej mapie głębi punkty po przetrzutowaniu do wirtualnej mapy głębi znajdują się w większej odległości, pomiędzy nimi powstanie nieciągłość.

W przypadku, gdy punkty te odpowiadały różnym obiektom sceny (jeden z punktów reprezentował bliższy obiekt, drugi – obiekt dalszy), nieciągłość ta powinna powstać. Jeżeli jednak oba punkty reprezentowały jeden obiekt, a większa odległość pomiędzy nimi w wirtualnej mapie głębi wynika z próbkowania obrazu, nieciągłość ta jest niepożądana i powinna zostać wyeliminowana.

Zaproponowano, by typowe podejście do rzutowania punktów zmienić, nie rzutując wszystkich punktów niezależnie, a w grupach po trzy sąsiednie punkty. Oba podejścia zestawiono na rysunku 3.6.



Rysunek 3.6. Rzutowanie punktów z rzeczywistej do wirtualnej mapy głębi: rzutowanie niezależne (A), rzutowanie zapewniające ciągłość obiektów (B)

Zaproponowany algorytm rozwiązuje problem pęknięć, występujący w innych metodach znanych z literatury. Składa się on z pięciu głównych kroków:

1. Trzy sąsiadujące z sobą (tworzące trójkąt) punkty są rzutowane z rzeczywistej mapy głębi zgodnie z (2.1). W wyniku rzutowania otrzymywane są pozycje trzech punktów w pośrednim widoku wirtualnym: $A = (x_A, y_A)$, $B = (x_B, y_B)$ i $C = (x_C, y_C)$ wraz z odpowiadającymi im wartościami głębi: z_A , z_B i z_C .
2. Sprawdzana jest głębia rzutowanych punktów. Jeżeli wartości głębi poszczególnych punktów w widoku rzeczywistym znacząco się różniły (punkty reprezentowały różne obiekty sceny), pomiędzy tymi punktami nie jest zachowywana ciągłość. Tym samym, punkty te są przetwarzane w typowy sposób, opisany w rozdziale 2.1. W przeciwnym wypadku, w celu zachowania ciągłości pomiędzy punktami, wykonywany jest krok trzeci.
3. Wyznaczany jest prostokątny obszar ograniczający w mapie głębi odpowiadającej pośredniemu widokowi wirtualnemu:

$$\begin{aligned} x_{min} &= \min(x_A, x_B, x_C), & x_{max} &= \max(x_A, x_B, x_C), \\ y_{min} &= \min(y_A, y_B, y_C), & y_{max} &= \max(y_A, y_B, y_C). \end{aligned} \quad (3.2)$$

4. Dla wszystkich punktów $V = (x_V, y_V)$ spełniających warunki:

$$x_V \in \mathbb{Z} \wedge x_V \in [x_{min}, x_{max}] \wedge y_V \in \mathbb{Z} \wedge y_V \in [y_{min}, y_{max}] \quad (3.3)$$

sprawdza się, czy punkt V znajduje się wewnątrz trójkąta o wierzchołkach (x_A, y_A) , (x_B, y_B) i (x_C, y_C) . Sprawdzenie to dokonywane jest poprzez porównanie trzech iloczynów wektorowych:

$$\vec{a} = \overrightarrow{AV} \times \overrightarrow{AB}, \quad \vec{b} = \overrightarrow{BV} \times \overrightarrow{BC}, \quad \vec{c} = \overrightarrow{CV} \times \overrightarrow{CA}. \quad (3.4)$$

Jeżeli wektory $\vec{a}$, $\vec{b}$ i $\vec{c}$ mają ten sam zwrot, punkt V znajduje się wewnątrz analizowanego trójkąta, a więc wartość głębi tego zostanie wyznaczona na podstawie wartości głębi w wierzchołkach trójkąta.

5. Wyznaczana jest wartość głębi dla wszystkich punktów V znajdujących się wewnątrz przerzutowanego trójkąta. Głębia punktów znajdujących się wewnątrz trójkąta powinna zmieniać się w sposób płynny. Z tego powodu, wartość głębi punktu V estymowana jest jako średnia ważona głębi trzech wierzchołków trójkąta.

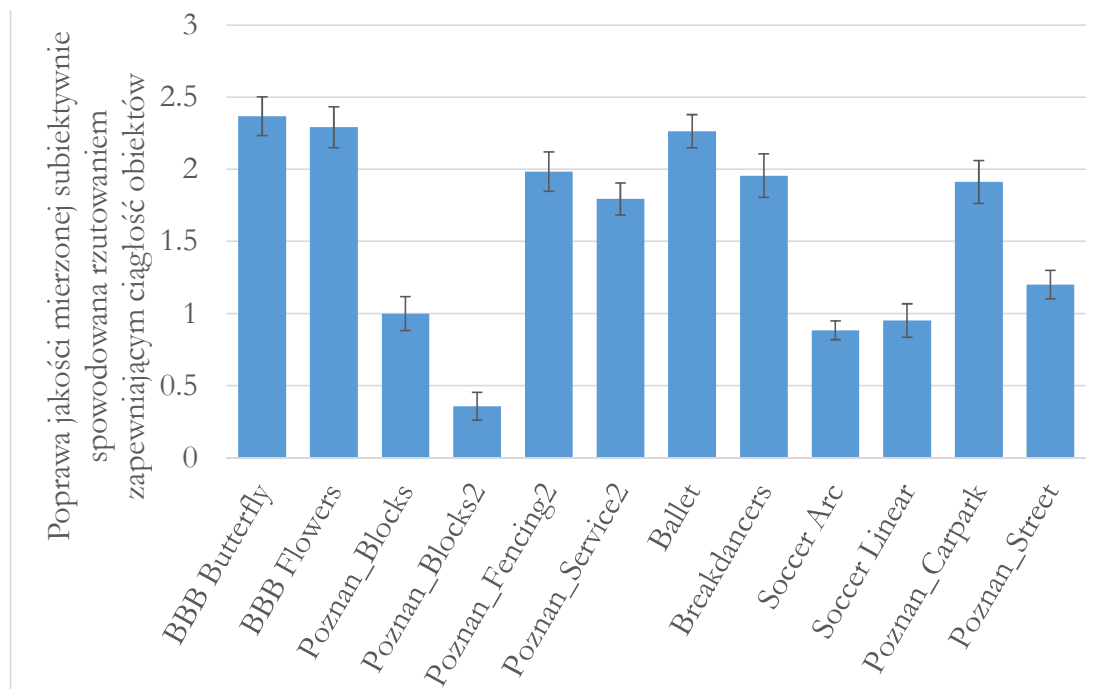
3.2.2. WYNIKI EKSPERYMENTALNE

Na rysunku 3.7 porównano widok wirtualny zsyntezowany w sposób typowy (z niezależnym rzutowaniem punktów) z widokami zsyntezowanymi przy zastosowaniu zaproponowanej techniki rzutowania punktów zapewniającej ciągłość obiektów. Jak pokazano na rysunku, rzutowanie trójkątami umożliwia poprawne przerzutowanie obiektów nawet z silnie oddalonych widoków rzeczywistych (kąt między osiami optycznymi kamer 06 i 84 wynosi 39 stopni).



Rysunek 3.7. Porównanie niezależnego rzutowania punktów (A) i rzutowania zapewniającego ciągłość obiektów (B); sekwencja *BBB Flowers* (B, D), synteza z widoku 06 do 84

Rzutowanie zapewniające ciągłość obiektów doskonale sprawdza się w jeszcze jednym przypadku – podczas wirtualnego „podejścia” do sceny, a więc w momencie, gdy użytkownik zbliża się do zarejestrowanych obiektów. W celu zmierzenia jakości w przypadku wirtualnego zbliżania się do sceny przeprowadzono testy subiektywne. W prezentowanych uczestnikom testów sekwencjach, wirtualna pozycja widza zmieniała się – w 10 sekund widz trzykrotnie przybliżał się do zarejestrowanej sceny, zaczynając od pozycji centralnej rzeczywistej kamery systemu wielokamerowego i poruszając się wzdłuż jej osi optycznej. Wynik przeprowadzonych testów zaprezentowano na rys. 3.8.



Rysunek 3.8. Poprawa mierzonej w sposób subiektywny jakości syntezowanych widoków przy zastosowaniu niezależnego rzutowania punktów i zaproponowanej techniki rzutowania zapewniającego ciągłość obiektów; dla każdej sekwencji oznaczono 95% przedział ufności; wartości dodatnie oznaczają, że jakość widoków syntezowanych przy użyciu MVS jest lepsza, niż w przypadku VSRS; metoda PC, skala: [-3, 3]

Jak pokazano, zaproponowana technika rzutowania punktów zapewniająca ciągłość obiektów pozwala na znaczącą poprawę jakości subiektywnej syntezy widoków w przypadku wirtualnego zbliżania się do zarejestrowanej sceny. Dla wszystkich dwunastu sekwencji testowych osiągnięta poprawa jakości jest statystycznie istotna.

3.3. KOREKCJA NIESPÓJNOŚCI BARWNEJ WIDOKU WIRTUALNEGO

Istotnym problemem w syntezie widoków wirtualnych dla rzadkich systemów wielokamerowych jest różna charakterystyka oświetlenia sceny w poszczególnych widokach rzeczywistych. Poza tym, różne kamery mogą cechować się różną charakterystyką barwną, a co za tym idzie w inny sposób rejestrować scenę. Oba wymienione problemy skutkują **niespójnością barwną**, a więc różnicą barwy tego samego punktu przerzutowanego z różnych widoków rzeczywistych. Niespójność barwna skutkuje gorszą jakością syntezy widoków.

Opisywane w literaturze techniki korekcji barwnej umożliwiają dopasowanie charakterystyki różnych widoków. Niemniej jednak, w celu zapewnienia widzowi wrażenia naturalności oglądanego materiału konieczne jest, aby charakterystyka barwna widoku wirtualnego dostosowana była do charakterystyki najbliższych widoków rzeczywistych. W przypadku dostosowania do widoków dalszych te obiekty, które odbijają światło w sposób nierównomierny, zostaną odwzorowane błędnie. Tym samym, chcąc odwzorować naturalne oświetlenie sceny, charakterystyka barwna powinna być dostosowywana do charakterystyki różnych widoków rzeczywistych – zależnie od tego, w którym miejscu znajduje się widz. Korekcja niespójności barwnej rzutowanych punktów musi być więc wykonywana **podczas syntezy**, a nie (jak w większości metod znanych z literatury) przed syntezą. Tym samym, zaproponowana technika powinna odznaczać się szybkością i niewielkim skomplikowaniem.

3.3.1. ADAPTACYJNA KOREKCJA NIESPÓJNOŚCI BARWNEJ

W przypadku opisywanej w rozprawie metody syntezy widoków wirtualnych korekcja niespójności barwnej [7] przeprowadzana jest podczas dwóch etapów syntezy: przy łączeniu pośrednich widoków wirtualnych wyznaczonych dla sąsiednich widoków rzeczywistych oraz podczas wypełniania odsłonień z dalszych widoków.

W widoku wirtualnym połączonym z dwóch sąsiednich pośrednich widoków wirtualnych wyróżnić można cztery rodzaje punktów:

1. Punkty przerzutowane z obu widoków rzeczywistych;
2. Punkty przerzutowane wyłącznie z widoku lewego;
3. Punkty przerzutowane wyłącznie z widoku prawego;
4. Punkty nieprzerzutowane.

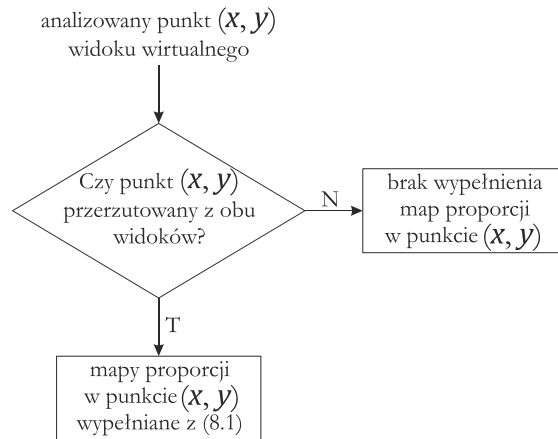
Punkty należące do pierwszej grupy charakteryzują się tym, iż ich barwa została wyznaczona poprzez zmieszanie barwy punktu przerzutowanego z lewego i prawego widoku rzeczywistego zgodnie z (2.8); barwa punktów z grupy drugiej i trzeciej natomiast została otrzymana poprzez skopiowanie barwy punktu z odpowiedniego pośredniego widoku wirtualnego. Charakterystyka barwna punktów z pierwszej grupy płynnie zmienia się podczas zmiany punktu widzenia, a więc nie musi zostać poprawiona. Korekcji zostają poddane wyłącznie punkty należące do grupy drugiej i trzeciej.

Dla uproszczenia przedstawione zostanie przetwarzanie wyłącznie luminancji, jednakże wszystkie opisywane operacje wykonywane są w sposób niezależny dla wszystkich trzech składowych Y , Cb i Cr .

W pierwszym kroku zarówno dla lewego jak i prawego widoku tworzona jest **mapa proporcji**. Sposób ich tworzenia przedstawiono w sposób schematyczny na rysunku 3.9. Dla każdego punktu z pierwszej grupy wyznaczany jest stosunek (proporcja) luminancji zmieszanej do luminancji przerzutowanej z pojedynczego widoku. Wartości tych proporcji umieszczane są w odpowiednich mapach proporcji:

$$M_L(x, y) = \frac{c_V(x, y)}{c_L(x, y)}, \quad M_R(x, y) = \frac{c_V(x, y)}{c_R(x, y)}, \quad (3.5)$$

gdzie M_L i M_R są mapami proporcji dla lewego i prawego widoku, $c_L(x, y)$ i $c_R(x, y)$ luminancją przerzutowaną z lewego i prawego widoku a $c_V(x, y)$ zmieszaną luminancją w punkcie (x, y) . W przypadku, gdy dany punkt nie został przerzutowany z obu widoków, stosunek nie jest liczony.



Rysunek 3.9. Tworzenie map proporcji

W drugim kroku następuje właściwa korekcja niespójności barwnej (rysunek 3.10). W kroku tym wszystkie punkty należące do grupy drugiej mnożone są przez lokalnie uśredniony współczynnik $\overline{M}_L(x, y)$, a grupy trzeciej przez $\overline{M}_R(x, y)$. Uśredniony współczynnik dla mapy proporcji lewego widoku wyznaczany jest ze wzoru:

Proponowane techniki poprawiające jakość syntezy widoków wirtualnych

$$\overline{M}_L(x, y) = \frac{1}{|P|} \sum_{(w, h) \in P} M_L(w, h), \quad (3.6)$$

gdzie P jest zbiorem wszystkich tych punktów w otoczeniu punktu (x, y) , dla których znana jest wartość w mapie proporcji, a poprzez otoczenie rozumie się kwadratowe okno o wysokości i szerokości kilkunastu okresów próbkowania. Współczynnik $\overline{M}_R(x, y)$ liczony jest w sposób analogiczny.

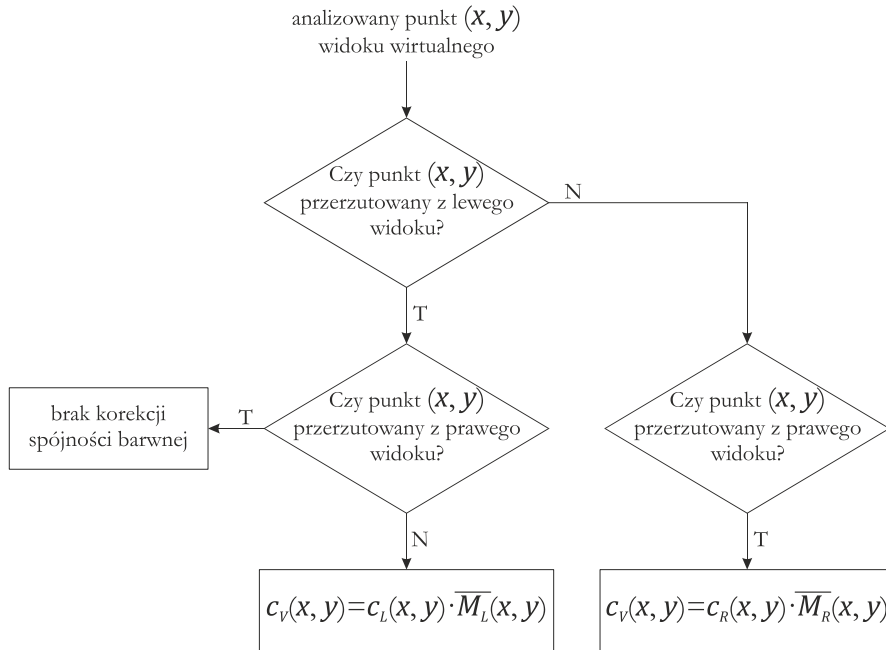
Poprawiona wartość luminancji punktu widoku wirtualnego wyznaczana jest ze wzoru:

$$c_V(x, y) = c_L(x, y) \cdot \overline{M}_L(x, y) \quad (3.7)$$

dla punktów przerzutowanych wyłącznie z lewego widoku oraz:

$$c_V(x, y) = c_R(x, y) \cdot \overline{M}_R(x, y) \quad (3.8)$$

dla punktów przerzutowanych z widoku prawego.



Rysunek 3.10. Algorytm adaptacyjnej korekcji niespójności barwnej:

c_V – poprawiona wartość luminancji; c_L, c_R – wartość luminancji przerzutowana z lewego i prawego widoku;

$\overline{M}_L, \overline{M}_R$ – uśredniony współczynnik dla lewej i prawej mapy proporcji

W podobny sposób przeprowadzana jest korekcja niespójności barwnej punktów rzutowanych z dalszych widoków rzeczywistych.

3.3.2. SZYBKĄ KOREKCJA NIESPÓJNOŚCI BARWNEJ

Dodatkowy czas przetwarzania widoku wirtualnego potrzebny do przeprowadzenia adaptacyjnej korekcji niespójności barwnej wynosi – w zależności od sekwencji liczony jest w pojedynczych sekundach (na typowym komputerze klasy PC, Intel i7-4770). Czas ten jest jednak zbyt długi, by zaproponowany algorytm mógł zostać użyty w systemie, w którym widoki wirtualne generowane są w czasie rzeczywistym, jak choćby w [23].

Aby sprostać wymaganiom rzeczywistego systemu swobodnej nawigacji, a więc umożliwić syntezę czasu rzeczywistego, w ramach rozprawy opracowano prostszy i znacznie szybszy algorytm korekcji niespójności barwnej [20]. Zauważono, iż najbardziej widocznymi błędami wynikającymi z różnej charakterystyki barwnej są przekłamania barwy na jednolitych powierzchniach – na tle. Opracowana technika szybkiej korekcji niespójności barwnej jest techniką nieadaptacyjną i opiera się na wyznaczeniu globalnej różnicy charakterystyk poszczególnych widoków.

Tak jak w przypadku opisanej wcześniej korekcji adaptacyjnej, szybka korekcja niespójności barwnej wykonywana jest dwukrotnie: podczas łączenia punktów przerzutowanych z sąsiednich widoków rzeczywistych i podczas wypełniania odsłonięć z pozostałych widoków.

W pierwszym kroku dla wszystkich tych punktów widoku wirtualnego, które zostały przerzutowane zarówno z lewego, jak i prawego widoku rzeczywistego (i których przerzutowana wartość głębi jest podobna), liczona jest różnica wszystkich trzech składowych Y , Cb i Cr pomiędzy barwą przerzutowaną z lewego widoku i barwą punktu z widoku prawego. Następnie wyznaczana jest uśredniona dla całego widoku wartość tej różnicy (ϵ), niezależnie dla każdej składowej. W ostatnim kroku średnia różnica dla poszczególnych składowych jest wykorzystywana do modyfikacji tych składowych dla punktów przerzutowanych z obu widoków rzeczywistych.

Poprawiona wartość poszczególnych składowych dla punktów przerzutowanych z widoku lewego dana jest wzorem:

$$c'_L(x, y) = c_L(x, y) - \frac{\epsilon \cdot d(L, V)}{d(L, V) + d(R, V)}, \quad (3.9)$$

gdzie $c_L(x, y)$ jest wartością danej składowej w punkcie (x, y) przed korekcją niespójności barwnej, ϵ – średnią różnicą wartości danej składowej przerzutowanej z lewego i prawego widoku, a $d(L, V)$ i $d(R, V)$ stanowią odległość euklidesową pomiędzy kamerą wirtualną a, odpowiednio, lewą i prawą kamerą rzeczywistą. Poprawiona wartość składowych barwnych dla punktów przerzutowanych z prawego widoku wyznaczana jest w sposób analogiczny.

Podobne operacje wykonywane są w przypadku punktów przerzutowanych z dalszych widoków rzeczywistych. Dla każdego punktu połączonego widoku wirtualnego sprawdzane jest, czy

został on przerzutowany również z dalszych widoków rzeczywistych. Jeżeli tak (i jeżeli przerzutowana wartość głębi jest podobna), dla tych widoków liczona jest różnica wszystkich trzech składowych jego barwy pomiędzy danym widokiem a połączonym widokiem wirtualnym. Następnie dla każdego widoku wyznaczana jest średnia wartość różnicy, niezależnie dla wszystkich trzech składowych.

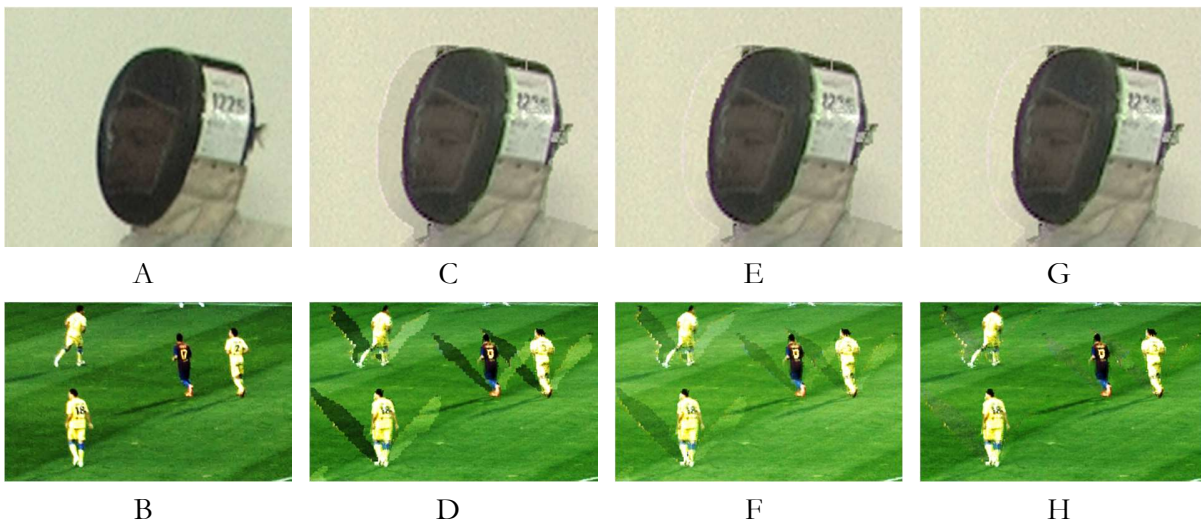
Poprawiona wartość poszczególnych składowych barwnych dla punktów przerzutowanych z widoku rzeczywistego i dana jest wzorem:

$$c'_i(x, y) = c_i(x, y) - \epsilon_i, \quad (3.10)$$

gdzie ϵ_i jest uśrednioną wartością różnicy danej składowej między widokiem i oraz połączonym widokiem wirtualnym. Do wykonania uproszczonej korekcji niespójności barwnej wymagane jest wykonanie jedynie dwóch dodatkowych operacji dodawania na każdy punkt obrazu dla każdego widoku rzeczywistego.

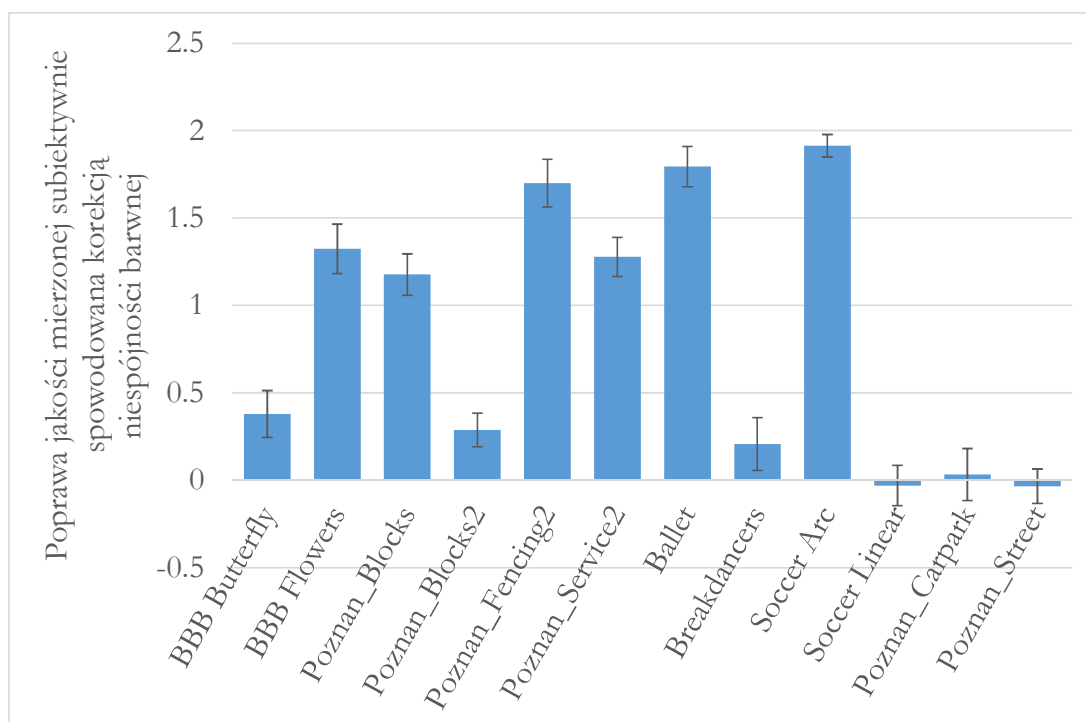
3.3.3. WYNIKI EKSPERYMENTALNE

Badania eksperymentalne przeprowadzono dla wszystkich 12 sekwencji testowych. Dla każdej z nich zbadano wpływ zaproponowanych technik korekcji niespójności barwnej na jakość syntezy widoków. Na rysunku 3.11 przedstawiono wynik syntezy dla fragmentów czterech wybranych sekwencji. W każdym przypadku zamieszczono 4 obrazy: fragment widoku odniesienia, widoku zsintezowanego bez korekcji niespójności barwnej, a także fragmenty dwóch widoków zsintezowanych z użyciem zaproponowanych technik korekcji niespójności barwnej.



Rysunek 3.11. Fragmenty widoków odniesienia (A, B), widoków syntezy bez korekcji niespójności barwnej (C, D), z szybką korekcją niespójności barwnej (E, F) i z adaptacyjną korekcją niespójności barwnej (G, H); dla uwypuklenia różnic zmodyfikowano jasność i kontrast przedstawionych fragmentów

W celu pokazania wpływu zaproponowanej korekcji niespójności barwnej na subiektywną jakość syntezowanych widoków, przeprowadzono testy subiektywne. Podczas testów grupa uczestników porównywała jakość widoków syntezowanych bez korekcji niespójności barwnej z jakością widoków otrzymanych przy włączonych obu technikach korekcji. Wynik badań przedstawiono na rysunku 3.12.



Rysunek 3.12. Poprawa mierzonej w sposób subiektywny jakości syntezowanych widoków z korekcją niespójności barwnej i bez korekcji; dla każdej sekwencji oznaczono 95% przedział ufności; wartości dodatnie oznaczają, że jakość widoków syntezowanych przy użyciu MVS jest lepsza, niż w przypadku VSRS; metoda PC, skala: [-3, 3]

Jak zaprezentowano, dla większości sekwencji testowych poprawa jakości mierzonej w sposób subiektywny jest statystycznie istotna. Wyjątkiem są trzy sekwencje zarejestrowane systemem z liniowym rozmieszczeniem kamer, *Poznan_Carpark*, *Poznan_Street* i *Soccer_Linear*, dla których jakość widoków syntezowanych z użyciem korekcji niespójności barwnej i bez niej jest porównywalna.

Największą poprawę jakości otrzymano dla sekwencji *Soccer_Arc*, gdzie wyeliminowane zostały błędy pokazane na rysunku 3.11D, a także *Poznan_Fencing2* i *Ballet*, gdzie dopasowana została charakterystyka tych obszarów tła, które były widoczne wyłącznie w jednym sąsiednim widoku.

4. OCENA EFEKTYWNOŚCI ZAPROPONOWANEJ METODY SYNTEZY

W niniejszym rozdziale zaproponowana metoda (MVS, *MultiView Synthesis*) została porównana z metodą odniesienia (zaimplementowaną w oprogramowaniu VSRS, *View Synthesis Reference Software*) rozwijanym przez międzynarodową grupę ekspertów MPEG.

4.1. OCENA JAKOŚCI SYNTEZY

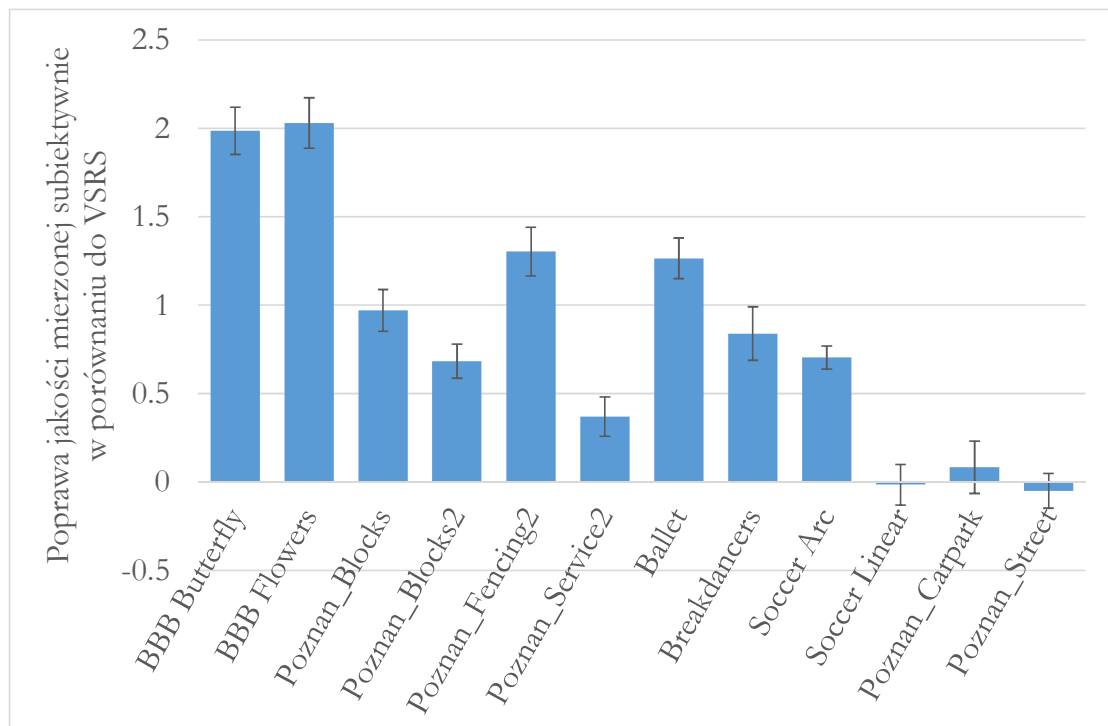
W tabeli 4.1 przedstawiono wartości PSNR syntezowanych widoków wirtualnych dla VSRS (lewa kolumna) i trzech konfiguracji zaproponowanej metody MVS.

Tabela 4.1. Porównanie uśrednionej jakości syntezowanych widoków dla VSRS i MVS
(kursywą oznaczono sekwencje zarejestrowane za pomocą systemów z gęstym, liniowym rozmieszczeniem kamer)

Sekwencja	PSNR [dB]			
	VSRS	MVS (2 widoki)	MVS (4 widoki)	MVS (wszystkie widoki)
BBB Butterfly	31,25	32,52	33,37	33,58
BBB Flowers	22,62	24,95	25,88	25,97
Poznan_Blocks	24,78	25,25	25,21	25,21
Poznan_Blocks2	29,60	29,89	31,53	31,62
Poznan_Fencing2	29,62	30,51	30,69	30,69
Poznan_Service2	24,68	25,09	25,50	25,48
Ballet	29,21	30,09	30,10	30,10
Breakdancers	30,62	31,62	31,64	31,64
Soccer Arc	20,51	21,06	21,24	21,21
<i>Soccer Linear</i>	34,18	34,03	34,26	34,28
<i>Poznan_Carpark</i>	33,50	33,80	33,89	33,89
<i>Poznan_Street</i>	35,48	35,69	35,72	35,72
Średnio	28,84	29,54	29,92	29,95

W przypadku metody MVS jakość widoków wirtualnych estymowano dla przypadku użycia dwóch widoków rzeczywistych, czterech widoków rzeczywistych i wszystkich dostępnych widoków rzeczywistych. Jak przedstawiono, uśredniona dla wszystkich sekwencji wartość PSNR widoku syntezowanego MVS jest o ponad 1,1 dB większa, niż dla widoku generowanego przy użyciu VSRS. Jeżeli wziąć pod uwagę wyłącznie sekwencje testowe zarejestrowane rzadkimi systemami wielokamerowymi (a więc wyłączając sekwencje *Soccer Linear*, *Poznan_Carpark* i *Poznan_Street*), korzyść płynąca z zastosowania MVS jest większa i wynosi niemal 1,5 dB (VSRS: 27,06 dB, MVS: 28,53 dB).

W celu porównania mierzonej w subiektywny sposób jakości widoków wirtualnych syntezowanych przy użyciu zaproponowanej metody i metody odniesienia, przeprowadzono testy, w czasie których uczestnicy oceniali różnicę jakości widoków zsintezowanych przy użyciu opracowanej w ramach rozprawy metody MVS i metody odniesienia VSRS. Czas trwania każdej oglądanej przez uczestników sekwencji wynosił 10 sekund. Wyniki przeprowadzonych testów przedstawiono na rysunku 4.1.



Rysunek 4.1. Poprawa mierzonej w sposób subiektywny jakości widoków syntezowanych MVS i VSRS; dla każdej sekwencji oznaczono 95% przedział ufności; wartości dodatnie oznaczają, że jakość widoków syntezowanych przy użyciu MVS jest lepsza, niż w przypadku VSRS; metoda PC, skala: [-3, 3]

Jak pokazano, dla większości sekwencji testowych użycie zaproponowanej metody powoduje poprawę jakości mierzonej w sposób subiektywny. Dla dziewięciu z nich poprawa jakości jest statystycznie istotna. Wśród sekwencji, dla których nie osiągnięto statystycznie istotnej poprawy jakości są *Soccer Linear*, *Poznan_Carpark* i *Poznan_Street*, które to zostały zarejestrowane przy użyciu gęstych, liniowych systemów wielokamerowych. Zauważyć należy, iż użycie zaproponowanej metody nie spowodowało statystycznie istotnej straty jakości dla żadnej sekwencji testowej, w tym sekwencji liniowych.

4.2. POMIAR CZASU OBLICZEŃ

Oprócz jakości syntezowanych widoków, istotny jest czas syntezy pojedynczego widoku wirtualnego. W celu zbadania i porównania czasu obliczeń przeprowadzono eksperyment. Komputer, jakiego użyto do testów miał następującą charakterystykę: procesor Intel i7-4770 (3,4 GHz), 32 GB pamięci RAM, dysk SSD.

W tabeli 4.2 przedstawiono porównanie czasu obliczeń dla metody VSRS oraz MVS w przypadku syntezy z użyciem dwóch, czterech i wszystkich widoków rzeczywistych. Zaprezentowany w tabeli czas jest czasem potrzebnym do zsyntezowania widoku wirtualnego

w najlepszej konfiguracji zaproponowanego algorytmu, a więc z włączonym rzutowaniem wymuszającym ciągłość obiektów oraz korekcją niespójności barwnej (adaptacyjną i szybką).

Tabela 4.2. Czas potrzebny do wygenerowania jednego widoku wirtualnego – dla VSRS i 3 konfiguracji MVS (dla syntezy z użyciem informacji z 2 i 4 sąsiednich widoków rzeczywistych oraz wszystkich widoków rzeczywistych (bez wstępnego zwiększania rozdzielczości))

Sekwencja	Czas syntezy 1 ramki [s]			
	VSRS	MVS (2 widoki)	MVS (4 widoki)	MVS (wszystkie widoki)
BBB Butterfly	1,58	1,06	1,41	1,79
BBB Flowers	1,54	0,92	1,44	2,00
Poznan_Blocks	3,68	1,63	2,65	4,82
Poznan_Blocks2	3,64	2,74	3,98	5,42
Poznan_Fencing2	2,24	2,00	3,17	5,20
Poznan_Service2	2,42	2,25	3,09	5,44
Ballet	0,90	0,51	0,81	1,44
Breakdancers	0,86	0,51	0,87	1,41
Soccer Arc	2,62	2,46	3,14	4,09
Soccer Linear	1,74	0,89	1,60	2,57
Poznan_Carpark	2,10	1,27	2,33	3,49
Poznan_Street	2,16	1,19	2,19	3,22
Średnio	2,12	1,45	2,22	3,41

W przypadku użycia 4 lub mniejszej liczby widoków rzeczywistych czas syntezy dla MVS stał się porównywalny z VSRS. Wykorzystanie wszystkich dostępnych widoków rzeczywistych zwiększa czas potrzebny na zsyntezowanie 1 widoku wirtualnego do niecałych 3,5 sekundy. Trzy i pół sekundy na ramkę nie jest jednak czasem, który umożliwiłby syntezę widoków wirtualnych w rzeczywistym systemie swobodnego punktu widzenia. Aby umożliwić widzowi swobodną nawigację w scenie, synteza widoków wirtualnych musi działać w czasie rzeczywistym [5].

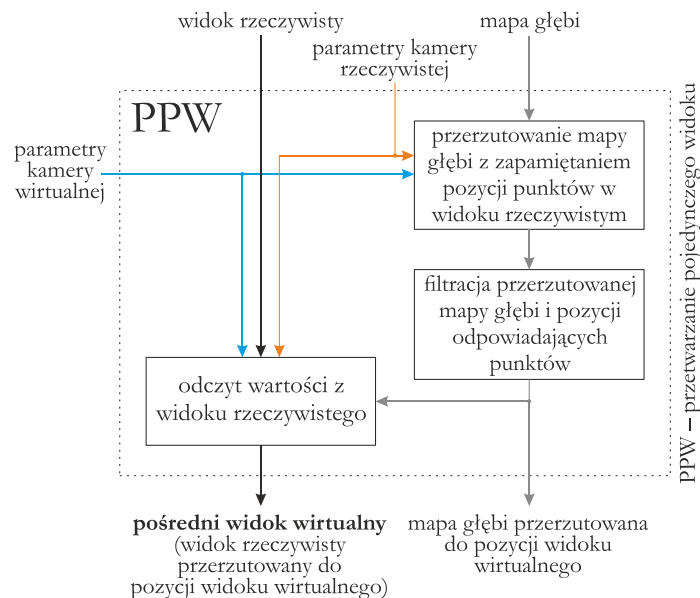
5. SYNTEZA CZASU RZECZYWISTEGO

W celu opracowania metody syntezy widoków wirtualnych działającej w czasie rzeczywistym (rtMVS, *real-time MultiView Synthesis*), zbadano czas potrzebny na wykonanie każdego etapu algorytmu. Aby widoki syntezowane były w czasie rzeczywistym, jedna ramka powinna być generowana w maksymalnie 40 milisekund (dla 25 obrazów na sekundę). W celu osiągnięcia wymaganej szybkości obliczeń dokonano szeregu optymalizacji oraz uproszczeń. Podstawowymi zmianami są:

1. Zastąpienie typowej syntezy wstecz metodą hybrydową.
2. Uproszczenie filtracji widoku i mapy głębi.
3. Uproszczenie sposobu łączenia informacji przerzutowanej z różnych widoków.
4. Uproszczenie algorytmu uzupełniania niezszyntezowanych obszarów.

W syntezie wstecz rzutowanie punktów przeprowadzane jest dwa razy – pierwszy raz w celu wyznaczenia wirtualnej mapy głębi i drugi raz w celu przerzutowania punktów z widoków rzeczywistych na bazie tejże mapy. Rzutowanie punktów wykonywane jest poprzez czasochłonne mnożenie pozycji i głębi tych punktów przez macierze projekcji, zgodnie z (2.1).

W celu uniknięcia drugiego cyklu rzutowania, opracowano szybką, hybrydową metodę syntezy [40, 4]. Zmodyfikowany schemat przetwarzania pojedynczego widoku (PPW) przedstawiono na rysunku 5.1.



Rysunek 5.1. Zmodyfikowany schemat przetwarzania informacji z pojedynczego widoku rzeczywistego wraz z odpowiadającą mu mapą głębi (schemat przed modyfikacjami przedstawiono na rysunku 2.2)

W metodzie tej rzutowanie punktów odbywa się wyłącznie raz – w celu wyznaczenia wirtualnej mapy głębi. Jest to możliwe dzięki temu, iż dla każdego punktu wirtualnej mapy głębi zapamiętywana jest pozycja, z której został on przerzutowany do widoku wirtualnego. Przykładowo, jeżeli punkt znajdujący się w wirtualnej mapie głębi na pozycji (x_V, y_V) znajdował się na pozycji (x_R, y_R) w widoku rzeczywistym, położenie (x_R, y_R) zostanie dla niego zapamiętane.

Filtracja wirtualnej mapy głębi dokonywana jest w dwójnasób. Po pierwsze, brakujące próbki głębi są interpolowane na podstawie próbek przerzutowanych. Równolegle dla wypełnianych próbek głębi interpolowane są również pozycje odpowiadających im punktów w widoku rzeczywistym.

Pozycje te wyznaczone są z dokładnością do połowy okresu próbkowania – dzięki temu operacja ta może zostać wykonana poprzez przesunięcie bitowe.

Również filtracja widoku wirtualnego została uproszczona. W rozdziale 3.1.2 opisano technikę bazującą na gradiencie otoczenia. W realizacji czasu rzeczywistego technika ta została zmodyfikowana. W uproszczonym algorytmie dla każdego punktu znajdującego się przy krawędziach obiektów w mapie głębi sprawdzane jest, czy ma on większą głębokość niż punkt po jego prawej stronie. Jeżeli tak, porównywana jest luminancja analizowanego punktu z luminancją punktu po stronie lewej. W przypadku znacznej różnicy punkt nie jest rzutowany do widoku wirtualnego. Jeżeli to punkt po lewej stronie charakteryzował się większą głębokością niż punkt analizowany, różnica luminancji sprawdzana jest dla lewego punktu i punktu z nim sąsiadującego. Analogiczne przetwarzanie wykonywane jest również w kierunku pionowym. Tym samym, dla każdego punktu widoku rzeczywistego leżącego przy krawędzi głębi wykonywane są jedynie cztery operacje porównania.

W rozdziale 2.2 przedstawiono sposób łączenia informacji przerzutowanej z dwóch widoków, bazujący na mieszaniu barwy punktów zależnie od odpowiadającej im wartości wirtualnej mapy głębi i odległości pomiędzy kamerą wirtualną i obiema sąsiednimi kamerami rzeczywistymi. W metodzie czasu rzeczywistego również ten etap został uproszczony, a mieszanie barw realizowane jest poprzez średnią arytmetyczną barw przerzutowanych z obu widoków, a więc może być zaimplementowane poprzez przesunięcia bitowe.

W syntezie czasu rzeczywistego zastosowano również zdecydowanie szybszą, acz mniej dokładną technikę uzupełniania widoku. W zastosowanym algorytmie uzupełniania wartości wypełnianych punktów wyznaczana jest na podstawie wartości czterech punktów – najbliższych przerzutowanych punktów z lewej i z prawej strony, a także najbliższych punktów powyżej i poniżej.

W pierwszym kroku sprawdzane jest, który z czterech analizowanych punktów ma największą głębokość. Następnie dla trzech pozostałych punktów sprawdzane jest, czy ich głębokość jest podobna do wyznaczonego maksimum. Jeżeli tak, punkty są uwzględniane przy uzupełnianiu – tym samym do uzupełniania może zostać użyte od jednego do czterech punktów. Dzięki takiemu podejściu możliwe jest wykonanie uzupełniania z użyciem operacji przesunięcia bitowego i sumy, bez używania czasochłonnej operacji dzielenia.

Czas potrzebny na wykonanie poszczególnych operacji dla przyspieszonej wersji algorytmu syntezy widoków wirtualnych przedstawiono w tabeli 5.1. Dzięki zastosowanym uproszczeniom i optymalizacjom udało się zmniejszyć czas obliczeń z 1160 ms do niecałych 90 ms. Zauważyć należy jednak, iż podany czas wyznaczony został dla implementacji jednowątkowej. W przypadku użycia większej liczby wątków czas ten oczywiście się zmniejszy.

Tabela 5.1. Średni czas wykonania etapów syntezy widoku wirtualnego (dla rozdzielczości 1920×1080)

Etap syntezy widoku wirtualnego	Czas obliczeń [ms]	Czas obliczeń [%]
Rzutowanie mapy głębi	50,00	57
Filtracja wirtualnej mapy głębi	6,40	7
Rzutowanie widoku rzeczywistego	8,26	9
Filtracja widoku wirtualnego	6,41	7
Szybka korekcja niespójności barwnej	6,00	7
Łączenie sąsiednich widoków pośrednich	1,04	1
Wypełnianie odsłoneń	5,00	6
Uzupełnianie widoku wirtualnego	4,17	5
Cały proces syntezy	87,28	100

Najprostszą do zrównoleglenia operacją jest rzutowanie mapy głębi. Operacja ta jest wykonywana niezależnie dla czterech używanych rzeczywistych map głębi, a jej wynikiem są cztery osobne mapy głębi, odpowiadające poszczególnym pośrednim widokom wirtualnym. Tym samym operacja może zostać w pełni zrównoleglona, a jej czas wykonania zmaleje czterokrotnie, skutkując czasem syntezy 1 widoku mniejszym niż 50 ms. Taki czas przetwarzania oznacza, iż przedstawiony algorytm umożliwia syntezę czasu rzeczywistego dla częstotliwości 20 obrazów na sekundę.

Tabela 5.2. Jakość syntezowanych widoków dla VSRS, rtMVS, MVS, średnia dla wszystkich widoków (w przypadku MVS użyto wszystkich widoków rzeczywistych; nie zastosowano wstępnego zwiększania rozdzielczości)

Sekwencja	PSNR [dB]		
	VSRS	rtMVS	MVS
BBB Butterfly	31,25	32,39	32,98
BBB Flowers	22,62	24,76	25,21
Poznan_Blocks	24,78	24,91	25,02
Poznan_Blocks2	29,60	30,30	31,00
Poznan_Fencing2	29,62	29,68	30,10
Poznan_Service2	24,68	24,80	25,17
Ballet	29,21	29,26	29,66
Breakdancers	30,62	31,28	31,29
Soccer Arc	20,51	20,69	20,88
Soccer Linear	34,18	34,14	34,18
Poznan_Carpark	33,50	33,62	33,64
Poznan_Street	35,48	35,48	35,49
Średnio	28,84	29,28	29,55

Po dokonaniu optymalizacji przeprowadzono badania sprawdzające, czy redukcja czasu obliczeń nie wpłynęła w sposób istotny na jakość syntezowanych widoków. W tabeli 5.2 jakość widoków syntezowanych w czasie rzeczywistym (za pomocą opracowanej metody rtMVS) porównano z jakością widoków syntezowanych przy użyciu MVS i metody odniesienia VSRS.

Jak zaprezentowano, jakość widoków syntezowanych przy użyciu zaproponowanej metody czasu rzeczywistego jest gorsza, niż w przypadku widoków syntezowanych algorytmem MVS. Taki wynik był jednak wynikiem oczekiwanym, jako że w rtMVS zastosowano uproszczenia zmniejszające jakość syntezowanych widoków. Pomimo tego, widoki syntezowane przy użyciu rtMVS mają wyraźnie lepszą jakość, niż widoki syntezowane za pomocą metody odniesienia. Jednocześnie, czas estymacji pojedynczego widoku został mocno zredukowany. Średni czas estymacji dla wszystkich sekwencji testowych i badanych metod porównano w tabeli 5.3.

Tabela 5.3. Średni czas syntezy pojedynczego widoku dla VSRS, rtMVS (implementacja 1- i 4-wątkowa) i MVS (w przypadku MVS użyto wszystkich widoków rzeczywistych; nie zastosowano wstępnego zwiększania rozdzielczości)

Sekwencja	Czas syntezy 1 ramki [ms]			
	VSRS	rtMVS (1 wątek)	rtMVS (4 wątki)	MVS
BBB Butterfly	1275	42	24	887
BBB Flowers	1488	47	28	877
Poznan_Blocks	3764	97	54	1700
Poznan_Blocks2	2738	89	50	1443
Poznan_Fencing2	2219	89	49	2119
Poznan_Service2	2189	91	52	1984
Ballet	989	34	19	504
Breakdancers	745	34	21	532
Soccer Arc	2379	90	49	2150
Soccer Linear	1524	56	33	1074
Poznan_Carpark	2151	74	40	1577
Poznan_Street	1883	74	42	1501
Średnio	1945	68	38	1362

W przypadku implementacji czterowątkowej, jedynym równolegle wykonywanym etapem było rzutowanie mapy głębi. Pozostałe operacje wykonywane były na jednym wątku procesora. Co istotne, w odróżnieniu od innych implementacji czasu rzeczywistego opisanych w literaturze, realizowanych na FPGA czy kartach graficznych, przedstawiony algorytm zaimplementowano na **procesorze**. Tym samym wykazano, iż możliwa jest szybka synteza widoków wirtualnych nawet dla tanich komputerów czy laptopów ze zintegrowanymi kartami graficznymi.

6. PODSUMOWANIE

6.1. UZASADNIENIE TEZY ROZPRAWY

W rozdziale 1.4 sformułowano tezę rozprawy:

TEZA

Możliwe jest opracowanie wydajnej metody syntezy widoków wirtualnych dobrej jakości poprzez wykorzystanie dalszych widoków rzeczywistych, filtrację krawędzi w widokach rzeczywistych i odpowiadających im mapach głębi oraz korekcję niespójności barwnej widoku wirtualnego.

W celu sprawdzenia prawdziwości tezy przeprowadzono szereg badań, w których wykorzystano autorską metodę syntezy widoków wirtualnych opracowaną w ramach rozprawy. Poszczególne fragmenty tezy wykazano badając poszczególne opracowane techniki, dotyczące kolejno: wykorzystania dalszych widoków rzeczywistych, filtracji i korekcji niespójności barwnej.

W rozdziale 2.4. dokonano porównania jakości widoków wirtualnych, w których do wypełnienia odsłonień użyto punktów z dalszych widoków rzeczywistych z jakością widoków wirtualnych, w których obszary odsłonięte były uzupełniane na podstawie otoczenia. Średnia poprawa jakości spowodowana użyciem informacji z dalszych widoków wyniosła **0,41 dB**.

Aby uniezależnić otrzymane rezultaty od wpływu zaproponowanego algorytmu uzupełniania dokonano również porównania z metodą odniesienia. Jako, że w metodzie odniesienia nie przewidziano możliwości wypełniania odsłonień z innych widoków, porównania dokonano wyłącznie dla obszarów wypełnionych z pozostałych widoków rzeczywistych. Średni wzrost jakości wynikający z zastosowania wypełniania odsłonień wyniósł ponad **2 dB**.

W rozdziale 3. opisano zaproponowane algorytmy filtracji krawędzi obiektów w widokach rzeczywistych (rozdział 3.1.2) i odpowiadających im mapach głębi (rozdział 3.1.1). Jakość widoków syntezy przy zastosowaniu opracowanych technik filtracji porównano z jakością widoków otrzymanych bez przeprowadzenia dodatkowej filtracji krawędzi.

Dla przyjętego zbioru sekwencji testowych zaproponowana technika filtracji krawędzi umożliwiła uzyskanie średnio **0,1 dB** większej wartości PSNR, niż w przypadku braku dodatkowej filtracji. Zdecydowanie większą poprawę jakości uzyskano w przypadku oceny jakości subiektywnej, gdzie dla 9 spośród 12 sekwencji testowych (w tym wszystkich sekwencji zarejestrowanych za pomocą rzadkich systemów wielokamerowych) otrzymano statystycznie istotną poprawę jakości.

W 3.3. rozdziale rozprawy opisano dwie opracowane techniki korekcji niespójności barwnej:

1. Technikę adaptacyjną, pozwalającą dopasować lokalną charakterystykę barwną obszarów widoku wirtualnego przerzutowanych z różnych widoków rzeczywistych,
2. Technikę globalną, wyrównującą charakterystykę barwną wszystkich punktów rzutowanych z poszczególnych widoków rzeczywistych.

Jak pokazano w rozdziale 3.3, zaproponowane techniki umożliwiają zwiększenie jakości syntezowanych widoków wirtualnych. W przypadku zastosowania obu technik korekcji niespójności barwnej średni wzrost jakości mierzonej poprzez PSNR wyniósł niemal **0,2 dB**. Tak jak w przypadku filtracji, również korekcja niespójności barwnej ma większy wpływ na jakość mierzoną w sposób subiektywny. Jak zaprezentowano na rysunku 3.12, dla 8 z 12 sekwencji testowych osiągnięto statystycznie istotną różnicę jakości.

W rozdziale 5. pokazano, iż zaprezentowany algorytm może – po dokonaniu kilku uproszczeń i optymalizacji – działać w czasie rzeczywistym, również dla sekwencji wysokiej rozdzielczości. Opracowana metoda rtMVS pozwoliła na uzyskanie wartości PSNR większą średnio o **0,5 dB** w porównaniu do widoków syntezowanych przy użyciu metody odniesienia przy jednoczesnym **51-krotnym** skróceniu czasu obliczeń.

Jak pokazano w rozdziale 5, dla implementacji jednowątkowej uzyskano średni czas syntezy jednego widoku mniejszy niż 70 ms. W przypadku użycia czterech wątków czas ten zmniejszył się do niecałych **40 ms** w przypadku uśrednienia czasu syntezy dla wszystkich sekwencji i niecałych **50 ms** dla sekwencji o rozmiarze obrazu 1920×1080.

6.2. PRZEPROWADZONE BADANIA

W celu przeprowadzenia badań zawartych w rozprawie autor stworzył ponad **13 000 linii kodu**, wliczając w to ponad 11 500 linii napisanych w języku C++ i 1500 linii w języku Python. Poza tym, w celu przeprowadzenia porównań z oprogramowaniem odniesienia VSRS autor dokonywał modyfikacji w tym oprogramowaniu (liczącym niemal 6000 linii kodu w języku C++). Tym samym, autor rozprawy pracował z niemal 20 tysiącami linii kodu.

Autor uczestniczył również w rejestracji i kalibracji wielowidokowych sekwencji testowych, wliczając w to sekwencje *Poznan_Blocks*, *Poznan_Blocks2* i *Poznan_Fencing2*.

Eksperymenty przeprowadzone w celu wygenerowania wyników zawartych w rozprawie wymagały niemal **200 godzin** nieprzerwanych obliczeń na jednym wątku komputera.

6.3. ORYGINALNE OSIĄGNIĘCIA

Najistotniejszym oryginalnym osiągnięciem rozprawy jest opracowanie dostosowanej do systemów z rzadkim rozmieszczeniem kamer metody syntezy widoków wirtualnych, w której do syntezy widoku wirtualnego wykorzystywana jest informacja pochodząca z wielu widoków rzeczywistych. W opracowanej metodzie obszary nieprzerzutowane z dwóch sąsiednich widoków rzeczywistych wypełniane są z widoków dalszych.

Analizując rezultaty dla dwunastu zróżnicowanych sekwencji testowych wykazano, iż jakość widoków wirtualnych syntezyowanych w zaproponowany sposób jest lepsza, niż jakość widoków syntezyowanych przy użyciu metody odniesienia. Średnia poprawa wartości PSNR wyniosła niemal **1,2 dB**. Co ważne, dla wszystkich sekwencji zarejestrowanych przy użyciu rzadkich systemów wielokamerowych uzyskano **statystycznie istotną poprawę** jakości mierzonej w sposób subiektywny.

Ponadto, opracowano kilka technik poprawiających jakość syntezyowanych widoków:

1. Technikę filtracji rozmytych krawędzi w widokach rzeczywistych bazującą na analizie gradientów,
2. Technikę filtracji rozmytych krawędzi w rzeczywistych mapach głębi,
3. Wymuszającą ciągłość obiektów technikę rzutowania punktów z widoków rzeczywistych,
4. Technikę adaptacyjnej korekcji niespójności barwnej, wyrównującą lokalną charakterystykę obszarów przerzutowanych z różnych widoków rzeczywistych,
5. Technikę szybkiej korekcji niespójności barwnej, wyrównującą globalną, wspólną dla całych widoków charakterystykę barwną obszarów rzutowanych z różnych widoków rzeczywistych,
6. Technikę wstępnego zwiększania rozdzielczości widoków rzeczywistych i odpowiadających im map głębi.

Niewątpliwym osiągnięciem rozprawy jest również opracowanie algorytmu syntezy widoków wirtualnych działającego w **czasie rzeczywistym** bez konieczności zastosowania kart graficznych czy programowalnych układów FPGA. Co istotne, zaproponowany algorytm umożliwia syntezę widoków mających średnio o **0,5 dB** większą wartość PSNR niż widoki syntezyowane przy użyciu metody odniesienia VSRS przy jednoczesnym **51-krotnym** skróceniu czasu obliczeń.

Podsumowując, w rozprawie wykazano, iż jest możliwe stworzenie efektywnej metody syntezy widoków wirtualnych dla rzadkich systemów wielokamerowych, a tym samym możliwe jest stworzenie systemu swobodnej nawigacji, w którym użytkownik może wirtualnie poruszać się w scenie zarejestrowanej przy pomocy rzadkiego systemu wielokamerowego.

7. DOROBK NAUKOWY AUTORA

CYTOWANIA

Łącznie ponad 100 cytowań, w tym:

- 5 w IEEE Transactions,
- 20 w innych czasopismach o zasięgu międzynarodowym,
- 67 w materiałach konferencji zamieszczonych w bazie IEEE Xplore,
- 12 w materiałach innych konferencji międzynarodowych,
- 17 w innych publikacjach.

Według systemu Google Scholar h-indeks wynosi 7.

ARTYKUŁY W CZASOPISMACH O ZASIĘGU MIĘDZYNARODOWYM

- [1] O. Stankiewicz, M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, J. Samelak, *A free-viewpoint television system for horizontal virtual navigation*, IEEE Transactions on Multimedia, tom 20, nr 8, s. 2182-2195, 08.2018, indeksowany w Web of Science, Impact Factor: 3.977.
- [2] A. Dziembowski, M. Domański, *View and depth preprocessing for view synthesis enhancement*, International Journal of Electronics and Telecommunications, tom 64, nr 3, s. 269-275, 2018, indeksowany w Web of Science.

ARTYKUŁY W MATERIAŁACH KONFERENCJI MIĘDZYNARODOWYCH ZAMIESZCZONYCH W BAZIE IEEE XPLORE ORAZ INDEKSOWANYCH W BAZACH SCOPUS I WEB OF SCIENCE

- [3] A. Dziembowski, J. Samelak, M. Domański, *View selection for virtual view synthesis in free navigation systems*, International Conference on Signals and Electronic Systems, ICSES 2018, Kraków, Polska, 10-12.09.2018, zamieszczony w bazie IEEE Xplore, oczekuje na indeksację w Web of Science.
- [4] A. Dziembowski, J. Stankowski, *Real-time CPU-based virtual view synthesis*, International Conference on Signals and Electronic Systems, ICSES 2018, Kraków, Polska, 10-12.09.2018, w bazie IEEE Xplore, oczekuje na indeksację w Web of Science.
- [5] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, D. Mieloch, O. Stankiewicz, J. Stankowski, K. Wegner, *Real-time virtual navigation provision by simple means*, International Conference on Signals and Electronic Systems, ICSES 2018, Kraków, Polska, 10-12.09.2018, w bazie IEEE Xplore, oczekuje na indeksację w Web of Science.

- [6] K. Wegner, D. Łosiewicz, T. Grajek, O. Stankiewicz, A. Dziembowski, M. Domański, *Omnidirectional view synthesis and test images*, International Conference on Signals and Electronic Systems, ICSES 2018, Kraków, Polska, 10-12.09.2018, zamieszczony w bazie IEEE Xplore, oczekuje na indeksację w Web of Science.
- [7] A. Dziembowski, M. Domański, *Adaptive color correction method in virtual view synthesis*, 3DTV Conference 2018, Sztokholm/Helsinki, Szwecja/Finlandia, 3-5.06.2018, zamieszczony w bazie IEEE Xplore, oczekuje na indeksację w Web of Science.
- [8] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, K. Klimaszewski, D. Mieloch, R. Ratajczak, O. Stankiewicz, J. Siast, J. Stankowski, K. Wegner, *Demonstration of a simple free viewpoint television systems*, IEEE International Conference on Image Processing, ICIP 2017, Pekin, Chiny, 17-20.09.2017.
- [9] D. Mieloch, A. Dziembowski, A. Grzelka, O. Stankiewicz, M. Domański, *Graph-based multiview depth estimation using segmentation*, IEEE International Conference on Multimedia and Expo, ICME 2017, Hongkong, Chiny, 10-14.07.2017.
- [10] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, M. Domański, *Enhancing view synthesis with image and depth map upsampling*, International Conference on Systems, Signals and Image Processing, IWSSIP 2017, Poznań, Polska, 22-24.05.2017.
- [11] D. Mieloch, A. Dziembowski, A. Grzelka, O. Stankiewicz, M. Domański, *Temporal enhancement of graph-based depth estimation method*, International Conference on Systems, Signals and Image Processing, IWSSIP 2017, Poznań, Polska, 22-24.05.2017.
- [12] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domański, *Multiview Synthesis – improved view synthesis for virtual navigation*, 32nd Picture Coding Symposium, PCS 2016, Norymberga, Niemcy, 4-7.12.2016.
- [13] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, M. Domański, *Depth map upsampling and refinement for FTV systems*, International Conference on Signals and Electronic Systems, ICSES 2016, Kraków, Polska, 5-7.09.2016.
- [14] M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, *Optimization of camera positions for free-navigation applications*, International Conference on Signals and Electronic Systems, ICSES 2016, Kraków, Polska, 5-7.09.2016.
- [15] M. Domański, M. Bartkowiak, A. Dziembowski, T. Grajek, A. Grzelka, A. Łuczak, D. Mieloch, J. Samelak, O. Stankiewicz, J. Stankowski, K. Wegner, *New results in free-viewpoint television systems for horizontal virtual navigation*, IEEE International Conference on Multimedia and Expo, ICME 2016, Seattle, Stany Zjednoczone, 11-15.07.2016.
- [16] A. Dziembowski, M. Domański, A. Grzelka, D. Mieloch, J. Stankowski, K. Wegner, *The influence of a lossy compression on the quality of estimated depth maps*, 23rd International Conference on Systems, Signals and Image Processing, IWSSIP 2016, Bratysława, Słowacja, 23-25.05.2016.

- [17] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, Ł. Kowalski, M. Kurc, A. Łuczak, D. Mieloch, R. Ratajczak, J. Samelak, O. Stankiewicz, J. Stankowski, K. Wegner, *Methods of high efficiency compression for transmission of spatial representation of motion scenes*, IEEE International Conference on Multimedia and Expo, ICME 2015, Turyn, Włochy, 29.06-3.07.2015.
- [18] M. Domański, A. Dziembowski, D. Mieloch, A. Łuczak, O. Stankiewicz, K. Wegner, *A practical approach to acquisition and processing of free viewpoint video*, 31st Picture Coding Symposium, PCS 2015, Cairns, Australia, 31.05-3.06.2015.
- [19] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, *Experiments on acquisition and processing of video for free-viewpoint television*, 3DTV Conference 2014, Budapeszt, Węgry, 2-4.07.2014.

ARTYKUŁY W MATERIAŁACH KONFERENCJI MIĘDZYNARODOWYCH ORGANIZOWANYCH PRZEZ GRUPĘ MPEG (ISO/IEC JTC1/SC29/WG11)

- [20] A. Dziembowski, O. Stankiewicz, *[MPEG-I Visual] Fast color correction technique for view synthesis*, ISO/IEC JTC1/SC29/WG11 MPEG2018, Ljubljana, Słowenia, M43694, 13-20.07.2018.
- [21] A. Dziembowski, D. Mieloch, K. Wegner, O. Stankiewicz, M. Domański, *Proposal of enhanced version of View Synthesis Reference Software with multiple input views*, ISO/IEC JTC1/SC29/WG11 MPEG2018, M42941, Ljubljana, Słowenia, 13-20.07.2018.
- [22] M. Domański, D. Łosiewicz, T. Grajek, O. Stankiewicz, K. Wegner, A. Dziembowski, D. Mieloch, *Extended VSRS for 360 degree video*, ISO/IEC JTC1/SC29/WG11 MPEG2018, M41990, Gwangju, Korea Południowa, 22-26.01.2018.
- [23] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, D. Mieloch, R. Ratajczak, J. Samelak, O. Stankiewicz, J. Stankowski, K. Wegner, *Free-viewpoint television demonstration for sports events*, ISO/IEC JTC1/SC29/WG11 MPEG2018, M41994, Gwangju, Korea Południowa, 22-26.01.2018.
- [24] K. Wegner, O. Stankiewicz, A. Dziembowski, D. Mieloch, M. Domański, *Exploration experiments on omnidirectional 6-DoF/3-DoF+ rendering*, ISO/IEC JTC1/SC29/WG11 MPEG2017, M41807, Makau, Chiny, 23-27.10.2017.
- [25] K. Wegner, O. Stankiewicz, A. Dziembowski, D. Mieloch, M. Domański, *Evaluation of step-in/step-out capability of state-of-the-art view synthesis technology*, ISO/IEC JTC1/SC29/WG11 MPEG2017, M40809, Turyn, Włochy, 17-21.07.2017.
- [26] K. Wegner, O. Stankiewicz, A. Dziembowski, D. Mieloch, M. Domański, *Omnidirectional 6-DoF/3-DoF+ rendering*, ISO/IEC JTC1/SC29/WG11 MPEG2017, M40806, Turyn, Włochy, 17-21.07.2017.

- [27] M. Domański, A. Dziembowski, A. Grzelka, Ł. Kowalski, D. Mieloch, J. Samelak, J. Stankowski, O. Stankiewicz, K. Wegner, *Experimental video coding software for free navigation applications*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M39527, Chengdu, Chiny, 17-21.10.2016.
- [28] M. Domański, A. Dziembowski, A. Grzelka, Ł. Kowalski, D. Mieloch, J. Samelak, J. Stankowski, O. Stankiewicz, K. Wegner, *Coding results for Poznan Fencing 2 and Poznan Blocks 2 test sequences in free navigation scenario*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M39215, Chengdu, Chiny, 17-21.10.2016.
- [29] M. Domański, A. Dziembowski, A. Grzelka, Ł. Kowalski, D. Mieloch, J. Samelak, J. Stankowski, O. Stankiewicz, K. Wegner, *[FTV AHG] Extended results of Poznan University of Technology proposal for Call for Evidence on free-viewpoint television*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M38246, Genewa, Szwajcaria, 30.05-3.06.2016.
- [30] M. Domański, A. Dziembowski, A. Grzelka, A. Łuczak, D. Mieloch, O. Stankiewicz, K. Wegner, *Multiview test video sequences for free navigation exploration obtained using pairs of cameras*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M38247, Genewa, Szwajcaria, 30.05-3.06.2016.
- [31] M. Domański, A. Dziembowski, A. Grzelka, D. Mieloch, *Study on nonuniform distributions of cameras located on an arc*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M38248, Genewa, Szwajcaria, 30.05-3.06.2016.
- [32] M. Domański, A. Dziembowski, A. Grzelka, Ł. Kowalski, D. Mieloch, J. Samelak, J. Stankowski, O. Stankiewicz, K. Wegner, *[FTV AHG] Technical Description of Poznan University of Technology proposal for Call for Evidence on free-viewpoint television*, ISO/IEC JTC1/SC29/WG11 MPEG2016, M37893, San Diego, Stany Zjednoczone, 22-26.02.2016.
- [33] M. Domański, A. Dziembowski, T. Grajek, A. Grzelka, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, *[FTV AHG] Video and depth multiview test sequences acquired with circular camera arrangement – „Poznan Service” and „Poznan People”*, ISO/IEC JTC1/SC29/WG11 MPEG2015, M36569, Warszawa, Polska, 20-27.06.2015.
- [34] M. Domański, K. Klimaszewski, A. Dziembowski, D. Mieloch, A. Łuczak, O. Stankiewicz, K. Wegner, *Freeview navigation (FN) anchor generation using 3D-HEVC with depth for „Poznan Blocks” sequence*, ISO/IEC JTC1/SC29/WG11 MPEG2015, M36565, Warszawa, Polska, 20-27.06.2015.
- [35] M. Domański, A. Dziembowski, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, *Poznan University of Technology test multiview video sequences acquired with circular camera arrangement – „Poznan Team” and „Poznan Blocks” sequences*,

- ISO/IEC JTC1/SC29/WG11 MPEG2015, M35846, Genewa, Szwajcaria, 14-20.02.2015.
- [36] M. Domański, A. Dziembowski, K. Klimaszewski, A. Łuczak, D. Mieloch, O. Stankiewicz, K. Wegner, *Comments on further standardization for free-viewpoint television*, ISO/IEC JTC1/SC29/WG11 MPEG2015, M35842, Genewa, Szwajcaria, 14-20.02.2015.
- [37] M. Domański, D. Mieloch, A. Dziembowski, O. Stankiewicz, K. Wegner, *Super multiview image compression: results for Bee sequence (FTV EE3)*, ISO/IEC JTC1/SC29/WG11 MPEG2014, M35070, Strasburg, Francja, 20-24.10.2014.
- [38] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, *Poznan Blocks – a multiview video test sequence and camera parameters for free viewpoint television*, ISO/IEC JTC1/SC29/WG11 MPEG2014, M32243, San Jose, Stany Zjednoczone, 13-17.01.2014.
- [39] K. Wegner, O. Stankiewicz, A. Dziembowski, D. Mieloch, M. Domański, *First version of depth maps for Poznan Blocks multiview video test sequence*, ISO/IEC JTC1/SC29/WG11 MPEG2014, M32248, San Jose, Stany Zjednoczone, 13-17.01.2014.

ARTYKUŁY W CZASOPISMACH POLSKICH

- [40] A. Dziembowski, J. Stankowski, *Szybka synteza widoków wirtualnych w systemach telewizji swobodnego punktu widzenia*, Przegląd Telekomunikacyjny, nr 6, s. 330-333, 2018.
- [41] A. Dziembowski, A. Grzelka, D. Mieloch, *Zwiększanie rozdzielczości obrazu i mapy głębi w celu poprawy jakości syntezy widoków wirtualnych*, Przegląd Telekomunikacyjny, tom 86, nr 6, s. 405-408, 2017.
- [42] D. Mieloch, A. Dziembowski, A. Grzelka, *Estymacja głębi dla systemów wielowidokowych*, Przegląd Telekomunikacyjny, tom 86, nr 6, s. 479-482, 2017.
- [43] A. Dziembowski, A. Grzelka, D. Mieloch, *Wielowidokowa synteza w systemach telewizji swobodnego punktu widzenia*, Przegląd Telekomunikacyjny, tom 88, nr 6, s. 233-236, 2016.
- [44] D. Mieloch, A. Dziembowski, A. Grzelka, *Segmentacja obrazu w estymacji map głębi*, Przegląd Telekomunikacyjny, tom 88, nr 6, s. 241-244, 2016.
- [45] M. Domański, A. Dziembowski, A. Kuehn, D. Mieloch, *Telewizja swobodnego punktu widzenia – nowa usługa czy futurystyczna wizja?*, Przegląd Telekomunikacyjny, nr 8-9, s. 734-737, 2014.
- [46] A. Dziembowski, A. Kuehn, A. Łuczak, D. Mieloch, K. Wegner, *Realizacja eksperymentalnego systemu telewizji swobodnego punktu widzenia z łukowym ustawieniem kamer*, Przegląd Telekomunikacyjny, nr 6, s. 161-164, 2014.