PARAMETRYZACJA GEOMETRII SCENY DLA ESTYMACJI GŁĘBI W DEKODERZE WIZJI WSZECHOGARNIAJĄCEJ

PARAMETRIZATION OF SCENE GEOMETRY FOR DECODER SIDE DEPTH ESTIMATION IN IMMERSIVE VIDEO

Błażej Szydełko, Dominika Klóska, Adrian Dziembowski
Instytut Telekomunikacji Multimedialnej, Politechnika Poznańska, Poznań
szydelkob@hotmail.com, dominikakkloska@gmail.com, adrian.dziembowski@put.poznan.pl

DOI: 10.15199/59.2022.4.64

Streszczenie**: Technika MPEG Immersive Video (MIV) jest rozwijana w celu efektywnego dostarczania wizji w systemie swobodnej nawigacji. W technice tej przewidziano brak przesyłania informacji o geometrii sceny, a rekonstruowanie jej po stronie dekodera w czasochłonnym procesie estymacji map głębi. Sposobem usprawnienia procesu estymacji jest wykorzystanie dodatkowej informacji o geometrii sceny przesyłanej jako sparametryzowane cechy map głębi. W artykule skupiono się na analizie skuteczności istniejącego rozwiązania, rozszerzając je o rekurencyjną ekstrakcję cech.**

Abstract**: The MPEG Immersive Video (MIV) technique is being developed for efficient delivery of immersive video. In one scheme, instead of transmitting geometry information, the technique aims to reconstruct it at the decoder side in a time-consuming depth map estimation process. The way to improve the estimation process is to use additional information about the scene geometry transmitted as parametrized features of the depth maps. This paper focuses on the efficiency analysis of an existing solution, extending it with recursive feature extraction.**

Słowa kluczowe**: estymacja głębi, wizja wszechogarniająca, rzeczywistość wirtualna, system swobodnej nawigacji**

Keywords**: depth estimation, immersive video, virtual reality, free navigation system**

1. WSTĘP

Ostatnimi czasy, wizja wszechogarniająca (immersyjna) stała się znaczącym kierunkiem rozwoju cyfrowych systemów multimedialnych [20]. W odróżnieniu od klasycznej wizji, gdzie widz zależny jest od woli reżysera bądź producenta, wizja immersyjna umożliwia zmianę punktu widzenia lub nawet ruch w obrębie sceny trójwymiarowej [17]. Doświadczenie immersji nie byłoby możliwe bez rozpowszechnienia się wyświetlaczy nagłownych (ang. Head-Mounted Display – HMD), które są w stanie w pewnym stopniu przekonać zmysły widza o przebywaniu w przedstawionej treści. Oprócz urządzeń HMD, oglądanie treści immersyjnych umożliwiają również standardowe wyświetlacze (w tym urządzenia mobilne) przy pomocy dowolnego urządzenia sterującego, takiego jak mysz komputerowa, joystick czy ekran dotykowy [16].

Zapewnienie użytkownikom swobody ruchu wymaga rejestracji sceny z kilku perspektyw oraz informacji o jej geometrii, by w procesie syntezy generować widoki pośrednie [4], [3]. Obecnie do reprezentacji sceny wykorzystuje się sekwencje wielowidokowe w formacie MVD (ang. Multiview Video + Depth) [15], składające się na zarejestrowane widoki i odpowiadające im mapy głębi (Rys. 1), które powstają w procesie estymacji lub są brane wprost z narzędzi do modelowania 3D, np. Blender (dla scen generowanych komputerowo).

Rys. 1. Sekwencja wielowidokowa (1 ramka, 2 widoki)

Ilość danych generowanych przez system immersyjny stawia wyzwanie wobec istniejących metod strumieniowania wizji. Zastosowanie standardowego kodera wizyjnego do niezależnego kodowania każdego z widoków oraz map głębi byłoby bardzo nieefektywne i niepraktyczne. Te niedogodności oraz rozwój technologii zapewniających immersję zintensyfikowały wysiłki standaryzacyjne w tej dziedzinie. Obecnie rozwijany standard MPEG Immersive Video (MIV) [1] ma na celu znormalizowanie efektywnego kodowania sekwencji wielowidokowych. Podczas gdy jego główny profil koncentruje się na usuwaniu redundancji między widokami i upakowaniu pozostałych danych [1], profil „Geometry Absent" (MIV GA) realizuje ideę estymacji geometrii sceny po stronie dekodera (ang. decoder-side depth estimation – DSDE) [5], [11]. W MIV GA strumień bitowy zawiera wyłącznie zestaw widoków wejściowych i ich parametrów. Pomimo rezygnacji z przesyłania informacji o geometrii sceny, jest ona wciąż dostępna po stronie kodera. Umożliwia to koderowi ekstrakcję zestawu parametrów i przesłanie ich w celu ułatwienia estymacji głębi po stronie dekodera [6], [5].

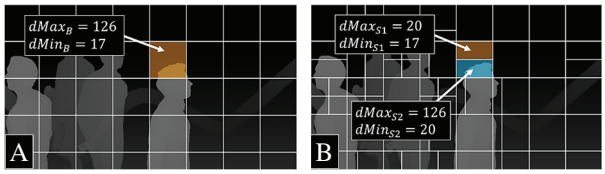
W niniejszym artykule przedstawiono propozycję rozszerzenia metody wspieranej w standardzie MIV o rekurencyjny algorytm ekstrakcji parametrów [18] wraz z efektywnym sposobem kompresji parametrów map głębi opartej na algorytmie kodowania kontekstowego CABAC (ang. Context-based Adaptive Binary Arithmetic Coding) [10]. Propozycja ta pozwala na lepsze dopasowanie parametrów do struktury map głębi, a tym samym na poprawę jakości i zmniejszenie czasu obliczeniowego estymacji głębi.

2. PARAMETRYZACJA GEOMETRII

Wykorzystanie parametrów pochodzących z map głębi, które są dostępne w koderze pomaga rozwiązać

podstawowe problemy związane z estymacją głębi po stronie dekodera. Po pierwsze, taka estymacja ma duży wpływ na złożoność procesu dekodowania, gdyż stanowi jego integralną część. Nawet najszybsze metody są kosztowne obliczeniowo i nie są w stanie zapewnić wysokiej jakości map głębi w czasie rzeczywistym [9]. Ponadto, jako że mapy głębi są estymowane przy użyciu skompresowanych widoków, zwykle obserwuje się pewne pogorszenie ich jakości [13]. Tę degradację spowodowaną kompresją można zmniejszyć, stosując metody redukcji artefaktów kompresji [7] lub poprawy jakości map głębi [12]. Jednakże zastosowanie dodatkowych etapów w przetwarzaniu końcowym wydłuża czas od momentu pozyskania strumienia bitów do zaprezentowania widoku wirtualnego końcowemu odbiorcy.

Podejście estymacji głębi zaprezentowane w [5] zakłada uwzględnienie dwóch kluczowych typów parametrów w zakodowanym strumieniu bitów. Parametry te są ekstrahowane bezpośrednio z map głębi i organizowane w postaci siatki bloków. Pierwszym typem jest zakres głębi w każdym bloku wejściowych map głębi (Rys. 2A). Przesłanie tej informacji zwiększa jakość estymowanej głębi poprzez redukcję możliwych błędów estymacji (które mogą być efektem kompresji widoków).



Rys. 2. Podstawowa idea ekstrakcji zakresu głębi; (A) siatka bloków głównych; (B) idea podziału bloków: blok B został podzielony na podbloki $S1$ i $S2$

Drugim typem parametru jest flaga pomijania estymacji, która informuje estymator, że dany blok głębi nieznacznie różni się od bloku w poprzedniej ramce i w związku z tym można skopiować blok z poprzedniej ramki. Należy nadmienić, że rozmiar bloków nie jest stały i istnieje możliwość podziału ich na cztery kwadratowe podbloki [5] lub na dwa prostokątne, zaproponowane przez autorów niniejszego artykułu w [19]. Wszystkie możliwe typy podziału bloków przedstawiono w Tab. 1.

Tab. 1. Możliwe typy bloków i odpowiadające im słowa kodowe [8].Oznaczenia typów: Q – czwórkowy (ang. quad), R – prostokątny (ang. rectangular)

Typ / Flaga		Q	R					
split	0	1	1	1	1	1	1	1
split_quad	-	1	0	0	0	0	0	0
split_orientation	-	-	0	0	0	1	1	1
split_symmetry	-	-	1	0	0	1	0	0
split_first_bigger	-	-	-	0	1	-	0	1

3. REKURENCYJNY PODZIAŁ BLOKÓW

3.1. Ekstrakcja parametrów

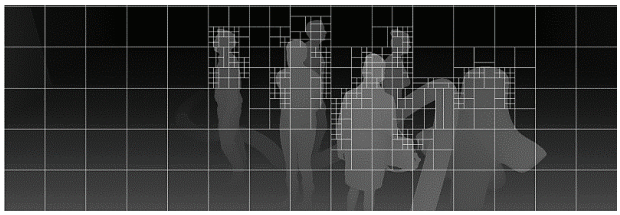
Obecne rozwiązania [5], [2] pozwalają podzielić blok tylko jeden raz (Rys. 2B). Z tego powodu nie ma możliwości dopasowania siatki bloków do bardziej skomplikowanej struktury mapy głębi. Aby temu zaradzić, proponujemy rozwiązanie rekurencyjne gdzie każdy blok może być wielokrotnie podzielony w celu dopasowania struktury bloków do struktury krawędzi mapy głębi.

Argument stojący za dokładniejszą parametryzacją geometrii wynika z podejścia stosowanego w estymacji głębi. Estymator nie sprawdza wszystkich możliwych wartości głębi (rozbieżności), a jedynie ogranicza się do zadanego zakresu ($dMin$, $dMax$). W związku z tym, jeżeli w bloku zawarte są punkty mapy głębi reprezentujące pojedynczy obiekt (np. fragment ściany), to zakres badanej rozbieżności jest bardzo wąski, co skutkuje redukcją czasu estymacji głębi oraz mniejszym prawdopodobieństwem wyznaczenia nieprawidłowej wartości głębi. W najkorzystniejszym przypadku, jeśli $dMin$ i $dMax$ dla bloku są równe, nie ma potrzeby estymacji głębi dla tego bloku.

Przed rozpoczęciem podziału rekurencyjnego, cała ramka głębi jest wstępnie dzielona na siatkę bloków o zadanym rozmiarze (Rys. 2A). Dla dalszego przetwarzania wymagane jest znalezienie zakresu głębi w każdym bloku. Kiedy różnica $dMax - dMin$ przekracza zadany próg podziału (ang. split threshold, domyślnie równy 2562 dla 16 bitowych map głębi), blok może zostać podzielony. Pozwala to na redukcję sumarycznej objętości bloku a tym samym rozłożenie procesu estymacji na mniejsze zakresy głębi. Jak wspomniano, obecne rozwiązanie oferuje 7 typów podziału bloku (Tab. 1). Wymagane jest więc wybranie podziału, który najefektywniej redukuje objętość bloku, a więc minimalizuje koszt (ang. cost volume – CV) dany wzorem:

$$CV_B = \sum_{S \in B} \left((dMax_S - dMin_S) + 1 \right) \cdot w_S \cdot h_S , \quad (1)$$

gdzie w_S i h_S są odpowiednio szerokością i wysokością podbloku S w bloku B. Opisany proces podziału jest powtarzany rekurencyjnie dla bloków utworzonych w poprzednich podziałach. Kiedy blok nie spełnia warunku do podzielenia lub jeden z wymiarów podbloku staje się mniejszy od ustalonego minimalnego rozmiaru, rekurencja kończy się. Rezultat podziału rekurencyjnego przedstawia Rys. 3.



Rys. 3. Mapa głębi podzielona rekurencyjnie

Ostatnim krokiem jest wykorzystanie informacji o spójności czasowej z map głębi. W tym celu odpowiadające sobie bloki są porównywane z blokami z poprzedniej ramki. Czasowe zróżnicowanie bloków jest wyrażone wzorem:

$$Z = \frac{\sum_{i=0}^{w_B-1} \sum_{j=0}^{h_B-1} \left| B_f(i,j) - B_{f-1}(i,j) \right|}{w_B \cdot h_B}, \quad (2)$$

gdzie f jest numerem ramki. Oczywiście zróżnicowanie Z nie jest liczone dla pierwszej ramki sekwencji.

Jeśli zróżnicowanie Z jest mniejsze od ustalonego progu pomijania (ang. skip threshold, domyślnie równy 2%), wysyłana jest flaga pominięcia estymacji głębi dla tego bloku. W celu zmniejszenia rozmiaru metadanych, flaga pominięcia wysyłana jest dla bloku nadrzędnego,

którego wszystkie podbloki są pomijane. Dodatkowo wartości $dMax$ i $dMin$ podlegają kwantyzacji.

3.2. Kompresja metadanych

Na metadane składają się wartości graniczne głębi oraz flagi sygnalizujące pomijanie estymacji i informujące o sposobie podziału bloku (Tab. 1). Jak można zauważyć, najprostszy rodzaj bloku jest sygnalizowany przez jeden bit o wartości równej zero, natomiast bloki pozwalające lepiej dostosować parametry do bardziej skomplikowanych struktur map głębi wymagają nawet do pięciu bitów sygnalizacyjnych. Podobieństwo zakresów głębi w sąsiednich blokach oraz zdefiniowane słowa kodowe stwarzają dogodne warunki do zakodowania metadanych przy użyciu kodowania kontekstowego CABAC [10], co w konsekwencji znacznie zmniejszy ich rozmiar. Dodanie nowych typów bloków oraz propozycja podziału rekurencyjnego wymusza aktualizację kodera metadanych o obsługę nowych słów kodowych i kontekstów.

Parametry geometrii są zorganizowane w formie drzewa czwórkowego, gdzie główny blok pełni funkcję korzenia, a ostatnie bloki, zawierające wartości $dMin$ i $dMax$ lub flagę pominięcia estymacji, stanowią liście. Każda z flag ma przypisany zbiór kontekstów reprezentujących prawdopodobieństwa wystąpienia symboli. W przypadku kodowania flagi pominięcia estymacji, kontekst jest wybierany na podstawie sumy wartości tychże flag z sąsiednich bloków (lewy i górny) i bloku z poprzedniej ramki, które są liśćmi drzewa (flaga pomięcia występuje tylko tam). Natomiast dla flag informujących o sposobie podziału bloku (a więc o strukturze drzewa) kontekst zależy tylko od wartości flag w bloku z poprzedniej ramki (znajdowany jest pierwszy blok o rozmiarze mniejszym lub równym kodowanemu). Na koniec, wartości $dMin$ i $dMax$ są kodowane w sposób predykcyjny, wykorzystując już zakodowane wartości w sąsiednim lewym i/lub górnym bloku.

4. BADANIA EKSPERYMENTALNE

4.1. Metodyka badań

Eksperymenty przeprowadzono według ustalonych warunków testowych (ang. Common Test Conditions) standardu MIV [21] przy wykorzystaniu potoku przetwarzania zdefiniowanego dla profilu MIV GA. Zbiór testowy zawierał 15 sekwencji w formacie MVD (sekwencja liczy 17 ramek). Z dostępnych w koderze map głębi wyekstrahowano parametry w 12 konfiguracjach, które następnie wraz z widokami przesłano do dekodera. Zdekodowane widoki i parametry geometrii wykorzystano do estymacji głębi metodą IVDE – Immersive Video Depth Estimation [14], która jest powszechnie wykorzystywana w eksperymentach związanych z MIV. Odtworzone mapy głębi i widoki wykorzystano do syntezy widoków wirtualnych, których jakość została oceniona miarą PSNR w odniesieniu do widoków oryginalnych.

W badaniach wykorzystano parametry dla 12 konfiguracji podziału bloków reprezentowanych w formacie $T - MAX - MIN$, gdzie MAX i MIN oznaczają odpowiednio maksymalny i minimalny rozmiar bloku, a T to jeden z dwóch rozważanych typów podziału bloków: Q przy podziale czwórkowym, a A przy wykorzystaniu wszystkich dostępnych podziałów ($Q + R$, R – podział prostokątny) (Tab. 1). Na przykład, $A - 64 - 8$ oznacza parametryzację z wykorzystaniem wszystkich typów podziału bloków, rozpoczynając od bloku o rozmiarze 64×64, a na 8×8 kończąc.

Dwa główne cele estymacji głębi po stronie dekodera z wykorzystaniem parametrów obejmują zwiększenie jakości syntezowanych widoków i skrócenie czasu dekodowania (w tym czasu potrzebnego na estymację głębi). W związku z powyższym, do oceny testowanych konfiguracji wzięto pod uwagę jakość syntezowanych widoków, prędkość bitową i czas estymacji głębi. Zaprezentowane w tabelach wyniki są średnimi wartościami wyznaczonymi po wszystkich zdekodowanych widokach testowanych sekwencji w pięciu różnych przepływnościach.

4.2. Wyniki

Analizując Tab. 2, można zauważyć, że mniejsze (lepiej dopasowane) bloki pozwalają na skrócenie czasu estymacji głębi. Jednakże, w przypadku użycia bardzo małych prostokątnych bloków czas estymacji może nieznacznie wzrosnąć, co ma związek z częstotliwością występowania flagi pomijania estymacji, której sygnalizacja następuje tylko dla odpowiadających sobie bloków w kolejnych ramkach. W przypadku podziału czwórkowego, pomimo niestabilności czasowej głębi, struktura bloków w sąsiednich ramkach może pozostawać niezmieniona, a w związku z tym, prawdopodobieństwo sygnalizacji pominięcia estymacji wzrasta przyczyniając się do skrócenia czasu estymacji map głębi.

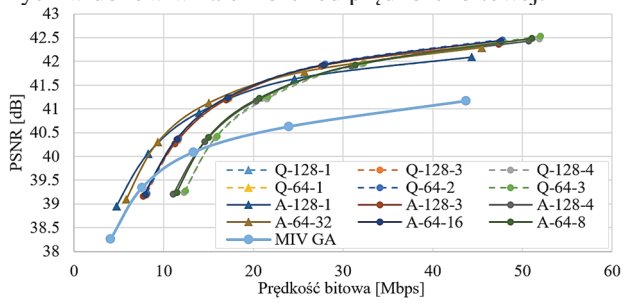
Tab. 2. Średni czas estymacji głębi w dekoderze [s]

Rozmiar bloku		**Typ podziału bloku T**	
MAX	MIN	Q	$A(Q+R)$
128	64[1]	4471,1	3731,3
	16	3939,5	3222,2
	8	3219,2	3885,9
64	32[1]	3880,4	3429,4
	16	3648,9	3873,7
	8	3558,7	3265,1
MIV GA[1]		12417,2	

[1] rozwiązanie wspierane w standardzie MIV [8]

W ostatnim wierszu Tab. 2 zaprezentowano wyniki uzyskane dla profilu MIV GA bez użycia parametryzacji głębi. Jak pokazano, przesłanie dodatkowych parametrów umożliwia znaczące przyspieszenie estymacji głębi, a więc całego procesu dekodowania.

Na Rys. 4 przedstawiono uśrednione krzywe RD uzyskane dla wszystkich testowanych konfiguracji kodeka MIV, które przedstawiają średnią jakość syntezowanych widoków w zależności od prędkości bitowej.



Rys. 4. Krzywe RD testowanych konfiguracji

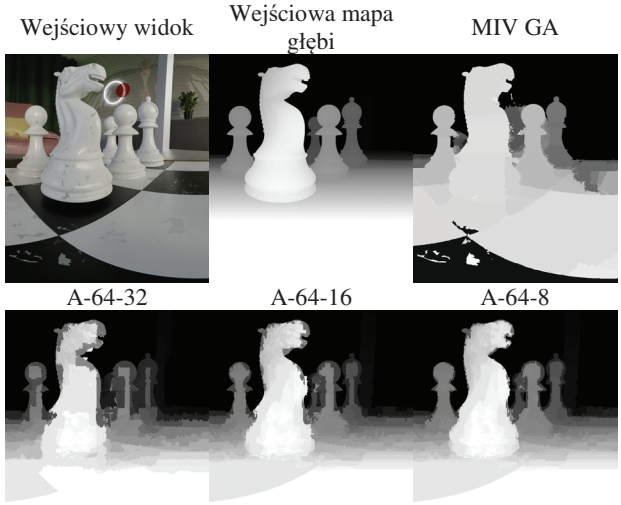
Wykorzystanie dowolnej konfiguracji do estymacji głębi znacznie poprawia jakość syntezowanych widoków w stosunku do MIV GA. Dokładniejsza parametryzacja powoduje wzrost prędkości bitowej metadanych (Tab. 3), znacznie przewyższając wartość dla MIV GA, jednakże zysk jakości oraz skrócenie czasu estymacji pozytywnie przeważają na korzyść prezentowanej metody.

Tab. 3. Średnie prędkości bitowe zakodowanych metadanych z parametrami [Mbps], wraz z ich procentowym udziałem w całym strumieniu

Rozmiar bloku		**Typ podziału bloku *T***	
MAX	*MIN*	Q	$A(Q+R)$
128	64[1]	0,656 (1,5%)	0,674 (1,5%)
	16	3,971 (8,3%)	3,651 (7,7%)
	8	8,171 (15,8%)	6,978 (13,7%)
64	32[1]	1,735 (3,8%)	1,775 (3,9%)
	16	4,040 (8,5%)	3,863 (8,1%)
	8	8,331 (16%)	7,375 (14,4%)
MIV GA[1]		0,0085 (0,02%)	

[1] rozwiązanie wspierane w standardzie MIV [8]

Jak można zauważyć na Rys. 5, wraz z dokładniejszą parametryzacją, szczegółowość estymowanej mapy głębi zwiększa się, a w związku z tym precyzyjniej odzwierciedla detale wejściowej mapy głębi i kształt obiektów w scenie. Ponadto, wykorzystanie parametrów pozwala wierniej odtworzyć wejściową mapę w stosunku do podejścia MIV GA, zwłaszcza w przestrzeniach pomiędzy obiektami i na płaskich powierzchniach.



Rys. 5. Fragment map głębi dla porównywanych konfiguracji i odpowiadający im widok wejściowy

5. PODSUMOWANIE

Zaprezentowana w niniejszym artykule metoda dokładniejszej parametryzacji geometrii sceny znacznie zwiększa wydajność estymacji głębi po stronie dekodera. Przeprowadzone eksperymenty pokazały, że pomimo zwiększenia ilości metadanych przesyłanych do dekodera, lepsze dopasowanie parametrów do struktury mapy głębi skutkuje zwiększeniem jakości wyznaczanej w dekoderze geometrii przy jednoczesnym skróceniu czasu estymacji. W związku z tym metoda ta idealnie nadaje się do zastosowań praktycznych, ponieważ dzięki niej można uzyskać wizję wszechogarniającą o lepszej jakości w porównaniu z metodą odniesienia.

PODZIĘKOWANIA

Praca finansowana ze środków przyznanych przez Ministerstwo Edukacji i Nauki.

LITERATURA

[1] Boyce J. M. *i in.* 2021. „MPEG Immersive Video Coding Standard". *Proc. IEEE,* 109 (9) : 1521–1536.
[2] Clare G. *i in*. 2021. „[MIV] Combination of m56626 and m56335 for Geometry Assistance SEI message". ISO/IEC JTC1/SC29/WG4 MPEG2021/M56950.
[3] Dziembowski A. *i in.* 2019. „Virtual View Synthesis for 3DoF+ Video". *Picture Coding Symposium*, 1–5.
[4] Fachada S. *i in*. 2018. „Depth image based view synthesis with multiple reference views for virtual reality". *3DTV-Conference.*
[5] Garus P. *i in*. 2021. „Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?". *IEEE T. on Circ. & Sys. for Vid. Technology.*
[6] Garus P. *i in*. 2019. „Bypassing Depth Maps Transmission For Immersive Video Coding". *PCS Conference*.
[7] He X. *i in*. 2020. „MV-GNN: Multi-View Graph Neural Network for Compression Artifacts Reduction". *IEEE Trans. on Image Processing*, 29 : 6829–6840.
[8] ISO/IEC 23090-12. „Information technology - Coded representation of immersive media - Part 12: MPEG Immersive video". ISO/IEC JTC 1/SC 29.
[9] Laga H. *i in*. 2020. „A Survey on Deep Learning Techniques for Stereo-based Depth Estimation", *IEEE Trans. on Pat. An. & Mach. Intell.* 44 (4) : 1738-1764.
[10] Marpe D. *i in*. 2003. „Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard". *IEEE Transactions on Circuits and Systems for Video Technology,* 13 (7) : 620–636.
[11] Mieloch D. *i in*. 2022. „Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video''. *IEEE T. on Circ. & Sys. for Vid. Technology.*
[12] Mieloch D. *i in*. 2021. „Depth Map Refinement for Immersive Video". *IEEE Access,* 9 : 10778–10788.
[13] Mieloch D. *i in*. 2021. „Point-to-Block Matching in Depth Estimation". *WSCG Conference*.
[14] Mieloch D. *i in*. 2020. „Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation". *IEEE Access,* 8: 5760–5776.
[15] Mueller K. *i in*. 2011. „3-D Video Representation Using Depth Maps". *Proc. of the IEEE,* 99 : 643–656.
[16] Stankiewicz O. *i in.* 2018. „A Free-Viewpoint Television System for Horizontal Virtual Navigation". *IEEE Transactions on Multimedia,* 20 (8) : 2182–2195.
[17] Stankiewicz O. *i in.* 2018. „Nowe media immersyjne". *Przegląd Telek. + Wiadomości Telekomunikacyjne,* 6.
[18] Szydełko B. *i in.* 2022. „Recursive block splitting in feature-driven decoder-side depth estimation". *ETRI Journal,* 44.
[19] Szydełko B. *i in.* 2021. „Rectangular blocks in encoder-derived features for decoder-side depth estimation". ISO/IEC JTC1/SC29/WG4 MPEG2021/M5635.
[20] Wien M. *i in.* 2019. „Standardization Status of Immersive Video Coding". *IEEE Journal on Emerging & Selected Topics in Circuits & Systems,* 9 (1).
[21] „Common Test Conditions for MPEG Immersive Video".ISO/IEC JTC1/SC29/WG4 MPEG2021/N005.