

**Zbigniew Gomółka, Bogusław
Twaróg, Bogdan Kwiatkowski, Ewa
Żesławska**

**Wirtualizacja obiektów w pasywnej
wizualizacji stereoskopowej =
Virtualization of Objects in Passive
Stereoscopic Visualisation**

Edukacja - Technika - Informatyka nr 1(19), 305-311

2017

Artykuł został opracowany do udostępnienia w internecie przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego. Artykuł jest umieszczony w kolekcji cyfrowej bazhum.muzhp.pl, gromadzącej zawartość polskich czasopism humanistycznych i społecznych.

Tekst jest udostępniony do wykorzystania w ramach
dozwolonego użytku.



**ZBIGNIEW GOMÓLKA¹, BOGUSŁAW TWARÓG²,
BOGDAN KWIATKOWSKI³, EWA ŻESŁAWSKA⁴**

Wirtualizacja obiektów w pasywnej wizualizacji stereoskopowej

Virtualization of Objects in Passive Stereoscopic Visualization

¹ Doktor, Uniwersytet Rzeszowski, Wydział Matematyczno-Przyrodniczy, Katedra Inżynierii Komputerowej, Polska

² Doktor inżynier, Uniwersytet Rzeszowski, Wydział Matematyczno-Przyrodniczy, Katedra Inżynierii Komputerowej, Polska

³ Doktor inżynier, Uniwersytet Rzeszowski, Wydział Matematyczno-Przyrodniczy, Katedra Inżynierii Komputerowej, Polska

⁴ Magister inżynier, Wyższa Szkoła Informatyki i Zarządzania w Rzeszowie, Wydział Informatyki Stosowanej, Katedra Zastosowań Systemów Informatycznych, Polska

Streszczenie

W artykule przedstawiono zagadnienia związane z wirtualizacją obiektów w pasywnej wizualizacji stereoskopowej. Wykorzystując środowisko Unity3D oraz platformę programistyczną C#, zrealizowano aplikację do pasywnej projekcji stereoskopowej. Przedstawiono zagadnienia związane ze stereoskopową projekcją 3D oraz etapy realizacji obrazu 3D.

Słowa kluczowe: technologia 3D, aktywny i pasywny system projekcji stereoskopowej, Unity3D, wirtualizacja obiektów

Abstract

The article presents issues related to virtualization of objects in passive stereoscopic visualization. Using the Unity3D environment and the C# programming platform, the application has been implemented for passive stereoscopic projection. The issues related to stereoscopic 3D projection and stages of 3D image presentation are presented.

Keywords: 3D technology, active and passive stereoscopic projection system, Unity3D, virtualization objects, computer generated images, stereoscopic 3D

Wstęp

Stereoskopia oraz wyświetlacze stereoskopowe stały się bardzo ważne dla wielu zastosowań, w tym do działania urządzeń zdalnych, obrazowania problemów natury medycznej, chirurgii, wizualizacji naukowych oraz projektowania komputerowego. Termin 3D występuje w 2 postaciach: w postaci generowanych

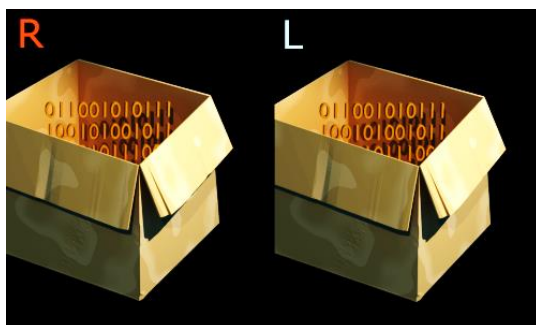
komputerowo obrazów graficznych (CGI – *computer generated images*), gdzie mamy do czynienia z wirtualnymi modelami obiektów 3D, oraz w postaci filmów stereoskopowych (s3D), gdzie obraz widziany przez odpowiednie okulary wydaje się wychodzić poza obszar ekranu. Zjawiska 3D czy stereoskopia odnoszą się do tego, jak nasze oczy i mózg postrzegają trzeci wymiar. Ludzkie oczy oddalone są od siebie o około 50–75 mm, dzięki czemu każde z osobna widzi delikatnie inny – przesunięty obraz. Możemy to zauważyć, trzymając palec przed naszą głową i na przemian zamykając oraz otwierając oczy. Takie 2 obrazy trafiają do naszego mózgu, w którym odbywają się złożone procesy geometryczne mające na celu zestawienie różnic między jednym a drugim okiem. Oparta na takiej zasadzie współczesna technologia 3D wykorzystuje specjalne okulary, które mają za zadanie dostarczyć naszym oczom różną perspektywę tego samego obrazu. Dla nas, a raczej dla naszego mózgu, cała matematyczna obróbka zsynchronizowania obrazów jest całkiem naturalna i prosta, natomiast zaprojektowanie kamery, która potrafi robić to samo z takim samym efektem, to coś trudniejszego (Zone, 2005).

Projekcja stereoskopowa 3D

Projekcję stereoskopową można zrealizować na różnych przykładach. W artykule proces ten przedstawiono, wykorzystując wirtualizację obiektów, czyli utworzenie wirtualnych kopii przedmiotów przy użyciu technologii 3D oraz zaprojektowanie odpowiedniej aplikacji pozwalającej wyświetlać te obiekty w formacie stereo *side-by-side*. Jako narzędzia programistyczne wykorzystano środowisko języka C# oraz silnik Unity3D do tworzenia interaktywnych multimediiów, gier i sekwencji wideo. Aby uzyskać obraz 3D, potrzebne były 2 wersje tej samej sceny sfilmowane z precyzyjnie odpowiednim kątem, tak jakby nasze oczy oglądały tę samą scenę. Obydwie kamery musiały poruszać się, śledzić i powiększać z tą samą prędkością, w przeciwnym wypadku obraz nie będzie synchroniczny. Zostały użyte do tego celu specjalne statywy pozwalające umieścić obok siebie 2 kamery, aby zapobiec rozbieżności. Kwestię tę można rozwiązać także, wykorzystując specjalne statywy lustrzane, gdzie obraz trafia do jednego obiektywu, a następnie odbijany jest przez lustro do innej kamery, która nagrywa drugi obraz (Sniderman, 2011).

Wszystkie stereoskopowe wyświetlacze pokazują prawy i lewy obraz zakodowany do wyświetlania na odbiorniku 2D i opierają się na systemie dekodującym, najczęściej w postaci określonego typu okularów, które filtrują obraz dla odpowiedniego oka. Różnica polega na sposobie kodowania i dekodowania obrazu (Sniderman, 2011). Zapisany obraz stereo może być wyświetlany w formacie: *mono*, *dualstream*, *side-by-side*, *ver/under*, *interlaced*, *anaglyph*, *frame-sequential*. Format *side-by-side* (rys. 1), czyli jeden obok drugiego, to 2 obrazy (jeden dla lewego, drugi dla prawego oka) ustawione obok siebie. W metodzie

side-by-side natywna rozdzielczość ekranu jest dzielona na połowę, bez zwiększania rozdzielczości na ekranie wyświetlane są 2 klatki – dla prawego i lewego oka jednocześnie. W praktyce powoduje to np., że rozdzielczość HD Ready, czyli 1280x720 pikseli, zmienia się poprzez 2 umieszczone obok siebie obrazy na rozdzielczość 640x720 pikseli, czyli spadek o 50% w poziomie. Taki sygnał po dotarciu do odbiornika zostaje rozdzielony na osobne klatki, obraz podniesiony do normalnej rozdzielczości, a ramki wyświetlone w wybranym systemie oferowanym przez urządzenie 3D.



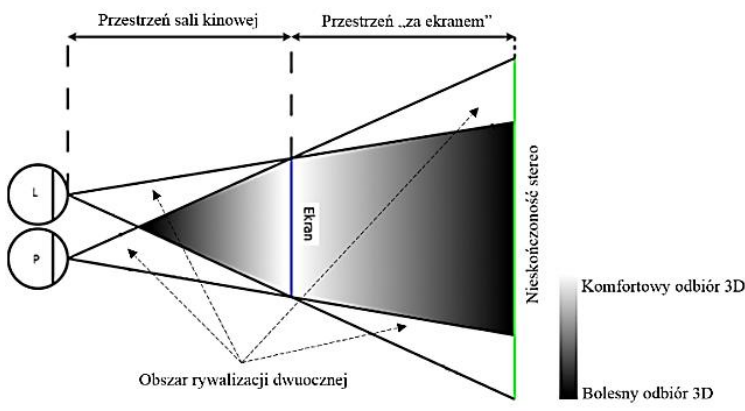
Rys. 1. Format zapisu stereo – *side-by-side*

Postępy w informatyce pozwalają obecnie na wyświetlanie obrazu 3D w szerokiej gamie produktów konsumenckich, jak laptopy, monitory, aparaty fotograficzne, kamery oraz wiele innych. Jednym z największych problemów technologii 3D jest zmniejszenie obrazu o połowę. Soczewkowe ekrany wysyłają pół ekranu do każdego oka, okulary migawkowe blokują jedno oko przed całością obrazu, a okulary polaryzacyjne dopuszczają tylko połowę światła dla każdego oka. Aby oglądać film jako jeden ciągły obraz, ludzkie oko potrzebuje 60 klatek na sekundę. Technologia 3D dzieli to na pół, umożliwiając każdemu oku oglądanie tylko 30 klatek, co może powodować migotanie czy mdłości. Aktualne systemy są w stanie zwiększyć liczbę klatek na sekundę dzięki np. częstotliwości odświeżania w telewizorach.

Wyróżnić można aktywny i pasywny system projekcji stereoskopowej. Popularną metodą systemu aktywnego jest *active 3D*, która wymaga użycia specjalnych okularów wyposażonych w przesłony LCD i kontrolowanych zależnie od producenta za pomocą Bluetootha, IR, czyli fali podczerwonych, lub WiFi, których praca zsynchronizowana jest z pracą wyświetlacza. System pasywny to system, w którym widz patrzy na ekran dzięki okularom zawierającym 2 odmiennie spolaryzowane filtry – dla oka lewego i dla prawego. Światło dla każdego z 2 wyświetlonych obrazów jest spolaryzowane i dociera do użytkownika przez odpowiedni filtr – lewy lub prawy.

Metody realizacji obrazu 3D

Jednym z ważniejszych pojęć w realizacji obrazu 3D jest *okno stereo* odnoszące się do fizycznej powierzchni wyświetlacza, np. ekranu telewizora, jako prawdziwego okna, które pozwala zobaczyć świat zewnętrzny. Obiekty w stereoskopowej scenie mogą występować za lub inaczej – na zewnątrz okna; mówimy wtedy o dodatniej paralaksie, „na” oknie – czyli płaszczyźnie ekranu z zerową paralaksą, lub między oknem a widzem z paralaksą ujemną. W ten sposób fotografowany temat będzie miał różne przesunięcia w poziomie (paralaksy) w zależności od jego głębokości w scenie. Paralaksa w filmie 3D jest wartością nadrzędną, bowiem jest nośnikiem głębi i decyduje o tym, czy efekt 3D w filmie jest dla nas atrakcyjny. W tym sposobie odległość między soczewkami jest zmienna, jak również kąt patrzenia kamery jest zmienny oraz odległość od najbliższego elementu decyduje o ustawieniu bazy stereoskopowej (Tseng, Anastassiou, 2014; Lincoln, 2011).



Rys. 2. Stereoskopowa strefa komfortu

Opracowanie odpowiednich ustawień 3D na potrzeby tworzenia projektu polega na adekwatnym skurczeniu świata rzeczywistego, tak aby go zmieścić wewnątrz przestrzeni trójwymiarowego ekranu. Tę przestrzeń tworzy interakcja naszych oczu z ekranem 2D, która może ulegać pewnej transformacji przy wykorzystaniu specjalnych efektów wizualnych. Trzy optyczne parametry pomagają w kontrolowaniu tego zabiegu: odległość międzysoczewkowa, konwergencja oczu oraz długość ogniskowej względem odległości od fotografowanego tematu. Po zgromadzeniu wszystkich reguł dotyczących percepcji stereoskopowej jesteśmy w stanie wyznaczyć pewną strefę komfortu określoną przez nasze oczy (rys. 2). Obiekt lub jego część znajdująca się w obszarach rywalizacji dwuocznzej jest widziana tylko przez jedno oko, co wywołuje zaburzenia postrzegania głębi (Mendiburu, 2009).

Projekt i realizacja aplikacji wizualizacyjnej

Zasadniczą sztuką poprawnie zrealizowanego projektu 3D jest umiejętność fotografii. Błędy popełnione na etapie zapisu obrazu są zmurą dalszej pracy, a narzędzia służące do eliminacji szkód, które na ogół pozwalają na ich naprawę w przypadku 2D, nie radzą sobie z obrazem 3D lub wymagają znacznie większego nakładu pracy. Przed przejściem do tworzenia zdjęć należy dobrać odpowiedni aparat wraz z obiektywem, dopasowując ogniskowe oraz przysłony, dostosowując światło do strefy komfortu i parametrów 3D. Ważnym krokiem jest wykonanie próbki zdjęć i sprawdzenie efektu 3D po wcześniejszym zakodowaniu obrazów do formatu obsługiwanego przez dostępny wyświetlacz 3D (Gomółka, Żesławska, Twaróg, Bolanowski, 2015, s. 430–433).

W pierwszym etapie realizacji aplikacji wykonano wirtualizację obiektów, która polegała na wykonaniu zdjęć trójwymiarowych n -wybrany obiektom, a następnie zrealizowano ich wirtualne repliki wewnątrz silnika środowiska Unity. Kolejnym etapem była wizualizacja obiektów – implementacja aplikacji pozwalającej na wyświetlanie wcześniej utworzonych obiektów metodą pasywnego stereo. Do wirtualizacji obiektów w pasywnej wizualizacji stereoskopowej wybrano następujące przedmioty: kamerę RED dragon, drzewko bonsai, wentylator podłogowy, drewnianą rzeźbę kiwi, koszyk, maskotkę bobra, maskotkę prosiaczka.

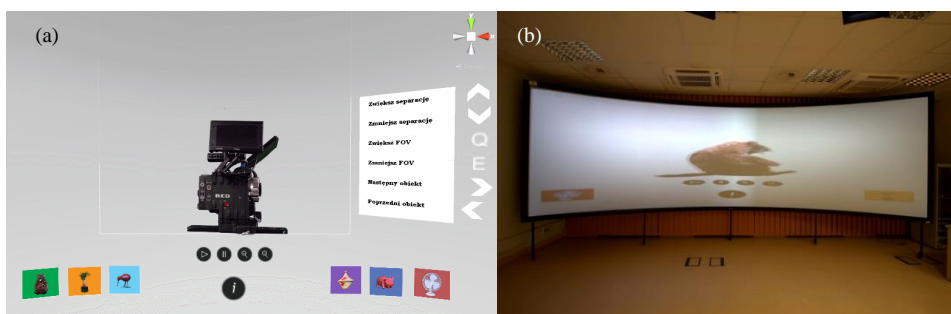


Rys. 3. Wizualizacja wykorzystanej techniki fotografii 3D

W celu zbudowania imitacji tych obiektów w scenie wirtualnej potrzebne było wykonanie 72 ujęć osobno dla lewego oka oraz prawego, co dla 7 obiektów daje łączną liczbę 1008 zdjęć. Efekt stereoskopii osiągnięty został dzięki aparatowi zamocowanemu na specjalnym statywie z ruchomą szyną (rys. 3); baza stereo wynosiła 6,5 cm.

Najważniejszą częścią w tworzeniu sceny pozwalającej wyświetlać obiekty stereoskopowo jest zaprojektowanie i zrealizowanie wirtualnego rigu (statywu)

z podwójnymi kamerami. W tym celu dodano do sceny sześćcian i rozciągnięto go wzdłuż osi X, formując rodzaj szyny. Utworzono jedną kamerę i zamocowano na środku szyny (kamera dla lewego oka), a drugą kamerę, dla prawego oka, zduplikowano z poprzedniej. Następnym krokiem było ustawienie obu kamerom odpowiednich warstw, które mają wyświetlać, odpowiednio warstwę *left* dla lewego oka i warstwę *right* dla prawego, oraz w ustawieniach komponentu dostosowano parametr *target eye* w podobny sposób. Uwzględniając, że zdjęcia są już wykonane w stereo, nie ma potrzeby rozsuwania kamer w scenie w celu uzyskania efektu 3D. Używając środowiska Unity3D oraz języka C#, zaimplementowano aplikację do wirtualizacji obiektów w pasywnej wizualizacji stereoskopowej (rys. 4a).



Rys. 4. Aplikacja do wirtualizacji obiektów (a), pasywna projekcja stereoskopowa (b)

Zrealizowaną aplikację testowano na 2 stanowiskach do projekcji 3D. Pierwsze pozytywne testy przeprowadzono na kompatybilnym, aktywnym telewizorze 3D wyposażonym w standard HDMI 1.4. Kolejne testy przeprowadzone były w Laboratorium Badań Kognitywistycznych 3D Uniwersytetu Rzeszowskiego. Testy przeprowadzono na stanowisku do pasywnej projekcji 3D. Składa się ono z 4 zsynchronizowanych projektorów, które realizowały zadanie projekcji stereoskopowej w trybie pasywnym, i również one zakończyły się powodzeniem i prawidłowym wyświetleniem obrazu z aplikacji. W celu zmaksymalizowania efektu 3D metodą doświadczalną wybrane zostały odpowiednie wartości parametrów separacji oczu oraz zakresu pola widzenia kamery. Rysunek 4b przedstawia obiekt z poprawnie dobranymi parametrami – obraz lewego i prawego oka zachodzi na siebie na środku ekranu, w wyniku czego otrzymujemy właściwy efekt 3D.

Podsumowanie

Dzięki wykorzystaniu profesjonalnego narzędzia, jakim jest środowisko graficzne Unity, silnik deweloperski, który zapewnia bogate ramy tworzenia aplikacji 2D i 3D lub gier, zaprojektowano i zaimplementowano złożoną aplikację

wirtualizacji obiektów w pasywnej wizualizacji stereoskopowej. W praktyce zrealizowane przedsięwzięcie jest pomostem pomiędzy implementacją świata wirtualnej a rozszerzonej rzeczywistości. Bardzo ważnym problemem w projekcie okazał się czas przetwarzania i wyświetlania projekcji graficznej, dla optymalizacji którego będzie w przyszłości zaproponowany wydajny algorytm w równoległej architekturze obliczeniowej.

Literatura

- Gomółka, Z., Żesławska, E., Twaróg, B., Bolanowski, M. (2015). Restitution of 3D Scenery with Coherent and Structured Light Scanner Technologies. *Measurement Automation Monitoring*, 9, 430–433.
- Lincoln, T. (2011). *Exercises in Three Dimensions*. Pobrane z: <http://www.lincolntom.com/pages/exercises%20in%203d.html> (28.06.2017).
- Mendiburu, B. (2009). *3d Film Making*. Burlington, MA: Focal Press.
- Sniderman, Z. (2011). *How Does 3D Technolohg Work?* Mashable. Pobrane z: http://mashable.com/2011/02/07/how-does-3d-work/#15O8B._g8Pqw (28.06.2017).
- Tseng, B., Anastassiou, D. (2014). *Compatible Video Coding of Stereoscopic Sequences Using MPEG-2's Scalability and Interlaced Structure*. University of Columbia.
- Zone, R. (2005). *3D Filmmakers: Interviews with 3D Cinematographers*. Toronto, Oxford: Scarecrow Press.