

WYKORZYSTANIE ROZSZERZONEJ RZECZYWISTOŚCI DO PRAKTYCZNEGO SZKOLENIA PERSONELU LOTNICZEGO

W artykule scharakteryzowano przebieg prac mających na celu stworzenie aplikacji mobilnej wykorzystującej technologię Mobile Augmented Reality wspomagającej proces szkolenia personelu obsługującego statki powietrzne w zakresie przeglądu przedlotowego samolotu. Przedstawiono wymagania jakim powinna odpowiadać aplikacja, dobór narzędzi programistycznych oraz mobilnych urządzeń obrazujących. W zakończeniu artykułu przedstawiono wyniki testów aplikacji na różnych urządzeniach mobilnych oraz wypływające z nich wnioski.

WSTĘP

Możliwości współczesnych systemów przetwarzania i wyświetlania obrazu pozwalają na wykorzystanie technologii „wirtualnej rzeczywistości” w coraz większej ilości zastosowań praktycznych. Odmianą „wirtualnej rzeczywistości” jest rozszerzona rzeczywistość (ang. Augmented Reality - AR), w której następuje połączenie rzeczywistych obrazów z wirtualnymi obiektami lub informacjami generowanymi przez jednostkę obliczeniową. Technologia ta wzbogaca ludzką percepcję i ułatwia zrozumienie skomplikowanych zjawisk.

Jednym z obszarów możliwych zastosowań technologii AR jest jej wykorzystanie w praktycznym szkoleniu personelu lotniczego obsługującego statki powietrzne (SP). Korzyści jakie potencjalnie może dać ta technologia zostały przedstawione w [2].

W artykule przedstawiono metodykę opracowania aplikacji wykorzystującej tę technologię, która wspomagałaby nauczanie i doskonalenia umiejętności praktycznych personelu lotniczego w zakresie wykonywania przeglądu przedlotowego przykładowego statku powietrznego.

Aplikacja została przygotowana dla samolotu Diamond DA20 Katana wykorzystywanym w Akademickim Ośrodku Szkolenia Lotniczego Wyższej Szkoły Oficerskiej Sił Powietrznych w Dęblinie.

Połączenie obrazu świata wirtualnego z rzeczywistym wymaga odpowiedniego oprogramowania i sprzętu [1]. Podstawowy system AR składa się z:

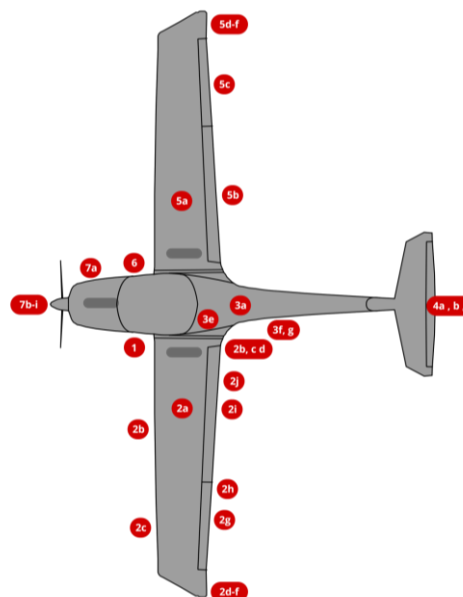
1. Sensorów zbierających informacje o otoczeniu takie jak własne położenie lub obraz świata zewnętrznego,
2. Aplikacji interpretującej zebrane przez sensory informacje,
3. Jednostki obliczeniowej mogąca przetworzyć algorytmy AR.
4. Urządzenia mobilnego, które pozwala na łączenie obrazów świata rzeczywistego z wirtualnym.

Do chwili obecnej opracowano wiele systemów AR wspomagających procesy obsługi maszyn i urządzeń oraz wspomaganie edukacji. Są to jednak systemy dość zamknięte i przeznaczone dla celów szkoleniowych czynności w niewielkim obszarowo środowisku i na małych rozmiarowo elementach. Przykładem może być nauczanie czynności dotyczących wymiany elementu elektronicznego na płycie głównej komputera, czy wymiana szczotek w silniku elektrycznym.

Podjęte przez autorów, i opisane w niniejszym artykule, zadanie dotyczy opracowania kursu szkolenia z zakresu czynności obsługi-

wych wykonywanych na statku powietrznym czyli na dużym obiekcie i obszernym środowisku.

Celem aplikacji byłoby wspomaganie procesu szkolenia studentów w zakresie przeprowadzenia i wykonania czynności obejmujących „Listę kontrolną normalnego użytkownika - przegląd przed lotem – przegląd wokół samolotu”. Zakres czynności obejmuje sprawdzenia wzrokowe mające na celu wykrycie ewentualnych uszkodzeń, zanieczyszczeń, pęknięć, odklejeń, nadmiernych luzów, niezabezpieczonych lub niewłaściwie zamocowanych elementów i ich stan ogólny. Składa się z 7 głównych punktów sprawdzeń i obejmujących sprawdzenia: lewego podwozia głównego, lewego skrzydła, kadłuba, usterzenia, prawego skrzydła, prawego podwozia głównego, nos [5].



Rys. 1. Etapy przeglądu wokół samolotu [opr. wł]

1. KONCEPCJA APLIKACJI WYKORZYSTUJĄCEJ TECHNOLOGIĘ ROZSZERZONEJ RZECZYWISTOŚCI

Aplikacja będzie wykorzystywała mobilną platformę (okulary, tablet lub smartfon) oraz znacznik, który będzie umieszczany w określonym miejscu sceny (rys. 2).



Rys. 2. Przykładowe umiejscowienie znacznika na SP [opr. wł]

Po uruchomieniu aplikacji należy zeskanować i rozpoznać kod znacznika (rys. 3.), który będzie punktem odniesienia dla modelu 3D i dalszych instrukcji wykonania przeglądu (rys. 4.).



Rys. 3. Działanie aplikacji - skanowanie kodu QR [opr. wł]



Rys. 4. Znacznik będący punktem odniesienia dla modelu 3D [opr. wł]

Syntetyczny obiekt AR w postaci modelu 3D samolotu zostanie naniesiony na obraz rzeczywistego SP (rys. 5.).



Rys. 5. Nałożenie modelu 3D (niebieskie kontury) na rzeczywisty obiekt oraz lokalizacja na SP pierwszego etapu sprawdzeń [opr. wł]

W celu wykonania aplikacji, której założenia przedstawiono powyżej zostały zrealizowane następujące przedsięwzięcia:

Część programistyczna:

- opracowanie aplikacji opartej na połączeniu narzędzi SDK (Software Development Kit) systemu mobilnego i SDK silnika rozpoznawania i śledzenia,
- opracowanie modelu 3D samolotu Diamond DA20,
- przygotowanie wykazu czynności wykonywanych przez szkolenego na samolocie Diamond DA20.

Część użytkowa:

- zainstalowanie aplikacji na mobilnej platformie, plik instalacyjny dostępny będzie na zewnętrznym serwerze,
- umieszczenie znacznika w określonym miejscu w stosunku do rzeczywistego samolotu,
- uruchomienie aplikacji i skierowanie aparatu mobilnej platformy na znacznik w celu inicjacji śledzenia,
- przeprowadzenie sprawdzeń według informacji wyświetlanych na telefonie.

Wymagania funkcjonalne aplikacji

Prezentowana aplikacja ma umożliwić przeprowadzić następujących procedur:

- lokalizacja i śledzenie wzorca, określanie położenia urządzenia względem odnośnika,
- renderowanie modelu 3D samolotu Diamond DA 20 w skali 1: 1 w czasie rzeczywistym,
- renderowanie animacji na modelu 3D w postaci sygnalizacji umiejscowienia danego elementu,
- wykonanie przez użytkownika procedur przeglądu przedstartowego,
- informowanie użytkownika o realizowanym etapie operacji,
- możliwość dodatkowej ręcznej kalibracji współrzędnych modelu w przypadku błędów współrzędnych śledzenia,
- możliwość wyboru skalowania modelu do celów edukacyjnych przez użytkownika,
- obrót modelu wokół własnej osi Z.

Możliwość renderingu modelu 3D oraz śledzenia położenia kamery w czasie rzeczywistym stanowi główną funkcję projektu. Aby zapewnić tą funkcjonalność należało na podstawie, uzyskanej w wyniku kwerendy literatury przedmiotu, wiedzy wybrać odpowiedni silnik śledzenia do przeprowadzenia tak skomplikowanej operacji.

Oprócz przedstawionych powyżej założeń funkcjonalnych konieczne było również określenie dodatkowych wymogów нефункциональных.

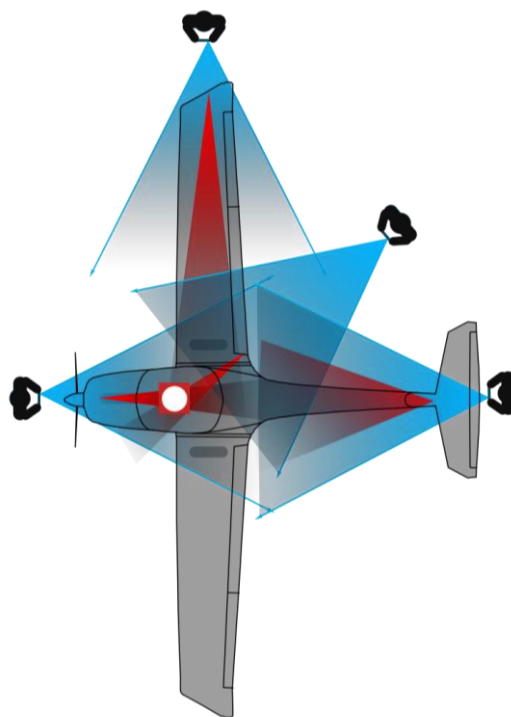
Założenia niefunkcjonalne

- Urządzenie mobilne powinno działać w systemie operacyjnym android 4.0.4. oraz być wyposażone w układ IMU (Inertial Measurement Unit – 3x akcelerometry, 3 x żyroskopy) i aparat fotograficzny,
- Aplikacja powinna być napisana w języku programistycznym Java oraz wykorzystywać technologie internetowe takie jak: Html, Css, JavaScript,
- Uniwersalna rozdzielczość pracy aplikacji powinna automatycznie dopasowywać się do rozdzielczości wykorzystywanego urządzenia mobilnego,
- Powinna istnieć możliwość korzystania z aplikacji w trybie offline.

Wybór silnika i technologii śledzenia

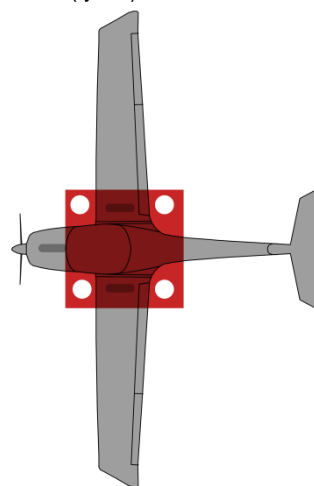
Ze względów praktycznych [2] jako urządzenia wyświetlające, które pozwala na łączenie obrazów świata rzeczywistego z wirtualnym wybrano inteligentne okulary Moverio BT-200. Z dostępnych rozwiązań śledzenia jedynie oprogramowanie Wikitude [8] oraz Vuforia [6, 7], posiadają implementacje i wsparcie dla tego typu okularów. Na podstawie przeprowadzonej analizy możliwości oraz dostępności dokumentacji dla obydwu pakietów zostało wybrane rozwiązanie pierwszego producenta. Dodatkowym atutem dla wyboru oprogramowania Wikitude były gotowe rozwiązania i przykłady programów, co umożliwiło dokładniejsze poznanie działania pakietu. Udostępniona nowa technologia śledzenia dużych scen w wersji beta dobrze się wpisywała do tworzonych założeń aplikacji. Jest jednak produktem odpłatnym, gdzie skrypty i algorytmy silnika stanowią własność intelektualną i nie są udostępnione. Do celów deweloperskich udostępniona jest wersja testowa, posiadająca wszystkie funkcje i działania jak pełnoprawny produkt. Wybrany został najnowszy SDK Wikitude 6.0 przy użyciu interfejsu Javascript API zawierający zintegrowany mechanizm renderowania 3D.

W roku 2016 Wikitude zaprezentował wersję betę technologii śledzenia SLAM dla dużych scen. Zastosowanie tej technologii umożliwiłoby stworzenie mapy statku powietrznego potrzebnej do śledzenia wokół dużego obiektu jakim jest samolot. Technologia w trakcie programowania została wycofana do testów i nie była wspierana przez producenta. Obecna technologia Instant Tracking SLAM ma jedynie zastosowanie dla małych scen. Określanie lokalizacji i lokalizacji jako początku układu AR realizowane jest za pomocą GPS. Przy wykorzystaniu tej technologii model obiektu jest wyświetlany naprzeciw użytkownika w odległości kilku metrów co nie pozwala na dokładne nałożenie siatki modelu 3D na realistyczny obiekt. Brak odniesienia wysokości powoduje, iż przybliżone dane wysokości powodują ułożenie modelu poza polem widzenia kamery. Najrozsądniejszym zatem rozwiązaniem okazało się śledzenie i rozpoznawanie za pomocą znacznika z zastosowaniem Extended Tracking. Podczas początkowej fazy prób znacznik zlokalizowano w najbardziej widocznym miejscu pod samolotem, w taki sposób, aby obchodząc samolot był choćby w części w polu widzenia kamery urządzenia (rys. 6).



Rys. 6. Martwe punkty pola widzenia kamery zaznaczone na czerwono [opr. wł]

Wstępne testy technologii Extended Tracking wykazały jej duże wady takie jak braki w trakcie uczenia się środowiska. Polegają one na gubieniu punktu środka współrzędnych czy nierozpoznawanie środowiska znacznika w strumieniu obrazu wideo. Również widoczność znacznika była bardzo słaba, ograniczana przez elementy samolotu takie jak skrzydła, podwozie i kadłub. Rozwiązaniem było zastosowanie znacznika, który umożliwiłby lepszą widoczność z każdej strony samolotu (rys. 7).



Rys. 7. Koncepcja znacznika pozwalający na widoczność z każdej strony [opr. wł]

Przygotowywanie modelu 3D i animacji

Przygotowanie modelu 3D samolotu pozwalającego na zaimplementowanie go do oprogramowania Wikitude wymagało poprawnego przejścia przez enkoder - 3D Encoder. Jest to aplikacja desktopowa dostępna na systemy Mac oraz Windows, która pozwala na optymalizację i konwersję plików z modelem 3D w formacie fbx do formatu wt3 obsługiwanego przez Wikitude SDK. W celu uzyskania możliwości bezbłędnej konwersji, model 3D samolotu musiał odpowiadać następującym wymaganiom:

- model statyczny (przygotowany na trójkątach),
- modele oświetlenia opracowane na podstawie modeli empirycznych Phong'a, Lambert'a, Blinn'a i przezroczystości,
- model wykonany zgodnie z metodą NURBS (ang. Non-Uniform Rational B-Spline)

W przypadku wystąpieniu błędów lub nieobsługiwanych funkcji (Multi tekstur czy Normal Mapping) oprogramowanie konwertujące wskaże je nam. Dodatkowe rozszerzenie obiektu o animacje jest możliwe za pomocą transformacji obiektu, natomiast animacje tekstur nie są obsługiwane.

Model 3D samolotu Diamond DA 20 został przygotowany w darmowym oprogramowaniu Blender wersja 2.78. Jest to powszechnie stosowane przez grafików komputerowych narzędzie służące do tworzenia i animacji modeli 3D. Model samolotu został oparty o efekt „wireframe” przedstawiający szkielet samolotu (rys. 8.). Pozwalało to w przypadku nałożeniu się modelu na sylwetkę samolotu jednoczesną widoczność obydwu. Zastosowanie wypełnionej teksturą wygenerowanego modelu 3D samolotu powodowałoby zakrycie w całości rzeczywistego samolotu, co uniemożliwiłoby wykonanie na nim jakichkolwiek czynności.

Model został przygotowany dla użytku mobilnego z uwzględnieniem faktu, że moc obliczeniowa urządzenia mobilnego jest ograniczona. Trójwymiarowa siatka modelu samolotu wykonana została zgodnie z wymaganiami „low-poly”, gdzie obiekt składa się z małej ilości wielokątów. Dzięki takiemu rozwiązaniu urządzenie mobilne jest w stanie generować trójwymiarowy model samolotu w czasie rzeczywistym



Rys. 8. Stworzony model 3D Diamond Da20 w programie Blender [opr. wł]

Głównym sposobem wskazującym położenie elementów na rzeczywistym samolocie w danym etapie przeglądu są nakładane na obraz rzeczywisty animacje. Animacja polega na transformacji obiektu do pełnowymiarowej strzałki i przesunięciu jej w kierunku danego elementu w czasie 2s. Każda z animacji musi być zdefiniowana oddzielnie. Powinna zostać wyodrębniona i indywidualnie adresowana tak by mogła zostać bezpośrednio wyzwalana przez aplikację (rys. 9.).



Rys. 9. Prezentacja po klatkowa przejścia wszystkich animacji „strzałek” na modelu 3D [opr. wł]

Architektura i interfejs aplikacji

Zestaw narzędzi Android SDK umożliwia tworzenie aplikacji na podstawie przygotowanego zestawu podstawowych narzędzi ADT Bundle [4], w skład których wchodzi:

- SDK Tools,
- Platform Tools,
- Zestaw bibliotek do najnowszej wersji Androida,
- Emulator oraz obraz systemu Android.

Aplikacja Java stworzona na platformę android, z wykorzystaniem bibliotek systemowych ma określoną ściśle strukturę. Nie musi zawierać wszystkich zawartych poniżej elementów, ale powinna zawierać, co najmniej jeden z nich:

- **Klasa Activity** – interfejs użytkownika,
- **Klasa Service** – dla komponentów działających okresowo lub w tle,
- **Klasa BroadcastReceiver** – komponent służący wymiany informacji pomiędzy aplikacjami,
- **Klasa ContentProvider** – komponent służący realizacji mechanizmu zapisu danych (plików, baz danych itp.).

Do stworzenia aplikacji zostało wykorzystane oprogramowanie Android Studio [3]. Jest to oficjalne wydanie Zintegrowanego Środowiska Programistycznego IDE dla systemu Android. Część kliencka platformy kompatybilna jest z Android API 17.

Finalnym produktem jest plik o rozszerzeniu apk. Jest to plik instalacyjny aplikacji na urządzenie mobilne z systemem android. Był on instalowany na różnych modelach urządzeń mobilnych w celu przeprowadzenia testów jego funkcjonalności.

2. TESTY APLIKACJI

W części testowej została sprawdzona poprawność działania aplikacji poprzez przeprowadzanie testów na różnych urządzeniach mobilnych. Działanie aplikacji oparte jest na interakcji użytkownika, z zewnętrznym środowiskiem jego działania. Nie jest one możliwe do odtworzenia środowiska w formacie cyfrowym, ani nie ma możliwości symulacji fizycznych sensorów znajdujących się w urządzeniach mobilnych. Do testów wykorzystano kilka modeli smartfonów oraz inteligentne okulary BT200 (tab. 1). Proces testów opierał się na sprawdzeniu działania następujących funkcji:

- rozpoznawanie i śledzenie wzorca,
- renderowanie modelu 3D,
- działanie wszystkich opcji i poprawność logiki aplikacji.

Tab. 1. Specyfikacja testowych urządzeń mobilnych.

Model	Samsung Galaxy J5(2016)	Samsung Galaxy TAB S2	Huawei Honor 7	Epson Movorio BT-200
Procesor	Qualcomm Snapdragon 410 8916 4 x 1,20 GHz	Samsung Exynos 7 Octa 5433 4x 1,9 GHz + 4x 1,3 GHz	Kirin 935 4 x 2,2 GHz	TI OMAP 4460 2 x 1,2 GHz
GPU	Adreno 306	Mali-T760 MP6	Mali-T628 MP4	-
RAM	2GB	3 GB	3 GB	1GB
Czujniki	3x akcelerometri	IMU	IMU	IMU
System	Android 6.0.1	Android 5.0	Android 5.0	Android 4.0.4

Głównym założeniem aplikacji było przeprowadzenie przeglądu przedstartowego samolotu z pomocą programu wspomagającego wykorzystującego technologię AR. Każdy etap operacji, którą szkolony personel miałby wykonać na samolocie zostałby wskazany na rzeczywistym obiekcie poprzez elementy świata AR na ekranie urządzenia. Naniesiony na rzeczywisty samolot jego model 3D

wykonany w postaci siatki byłby bazą do przygotowania kolejnych operacji.

Największą trudnością w realizacji tego zadania było śledzenie znacznika w trakcie przejścia użytkownika wokół samolotu. Na wstępie przeprowadzono testy przy wykorzystaniu prostego modelu 3D (rys.10.), sprawdzające możliwe kąty śledzenia oraz odległość od znacznika przy wykorzystaniu pełnej i częściowej widoczności znacznika (tab. 2.). Do testów wykorzystano telefon posiadający układ IMU oraz znacznik o rozmiarach kartki A4. W przeprowadzonych testach można było zauważyć wiele pojawiających się artefaktów, min. drganie elementów AR oraz złe określenie współrzędnych. Utrzymanie dobrych parametrów śledzenia znacznika przy odległościach większych niż 1m wymagało od użytkownika wykonywania bardzo powolnych ruchów. Znacznik na tych odległościach jest bardzo mały i wykrywany był jedynie częściowo, co przy gwałtownych ruchach powoduje iż automatyczna regulacja ostrości aparatu nie nadąża z ustawieniem nowej ogniskowej wskutek czego uzyskiwany obraz jest rozmyty, a algorytm nie był w stanie odczytać charakterystyk znacznika.

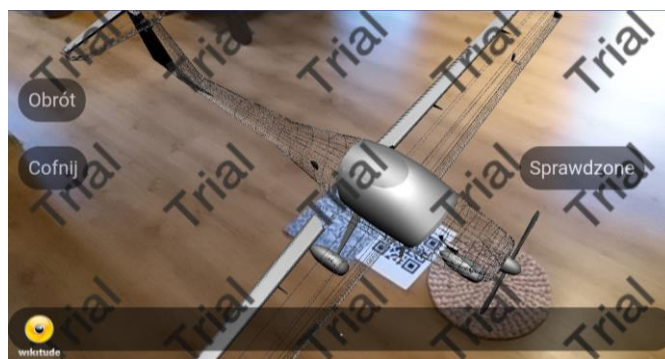
Tab. 2 Badania, jakości śledzenia i rozpoznawania znacznika za pomocą Samsung Galaxy J5

Odległość i kąt widzenia znacznika	l=1m $\alpha=0^\circ$	l=1m $\alpha=40^\circ$	l=1m $\alpha=80^\circ$	l=3m $\alpha=0^\circ$	l=3m $\alpha=40^\circ$	l=3m $\alpha=80^\circ$
Widoczność znacznika: 100%	bdb	bdb	db	db+	dst	ndst
Widoczność znacznika: 50%	bdb	bdb	dst	db	dst	brak

l – odległość do znacznika
 α – kąt widzenia znacznika

Poprawne renderowanie modelu 3D w czasie rzeczywistym wykorzystuje duże zasoby systemowe urządzenia. W przeprowadzonych testach renderowania, implementowano model 3D samolotu w skali 1:1, jako element AR. Po wykryciu znacznika inicjowany jest rendering modelu samolotu (rys.10.). Test polegał na wyświetleniu modelu w skali 1:1. Wykorzystano przy tym teście wszystkie dostępne urządzenia. W żadnym z nich nie udało się zainicjować renderingu modelu. Przyczyną jest tutaj prawdopodobnie zbyt słaba moc obliczeniowa niezbędna do wygenerowania modelu. Według utworzonych logów aplikacji, model w tym przypadku nie został nawet załadowany. W celu zmniejszenia obciążenia sprzętowego wprowadzono pogorszenie jakości nagrywania oraz uruchamianie aplikacji poprzez dedykowane aplikacje rezerwujące moc obliczeniową kosztem działania systemu. Najlepsze wyniki uzyskano w przypadku modelu w skali 1:7.

Drugą z możliwych przyczyn niepowodzenia jest ograniczenie oprogramowania Wikitude. W trakcie przeprowadzanych dodatkowych testów z wykorzystaniem innych modeli 3D, o mniejszej ilości poligonów i wymiarach zbliżonych do rzeczywistych wymiarów samolotu DA20 wystąpiły błędy inicjowania modelu występujące również przy inicjowaniu głównego modelu samolotu.



Rys. 10. Renderowany model w trakcie śledzenia znacznika oraz interfejs sterowania [opr. wl]

załadowany model w skali 1:7 do pamięci aplikacji pozwala na odtwarzanie przypisanych do niego animacji w etapach dla danych operacji sprawdzeń samolotu. Animacje polegają na wskazaniu w danej operacji elementu do sprawdzenia oraz dodatkowo możliwy jest obrót modelu wokół osi pionowej w przypadku utrudnionej widoczności tego elementu. Np. gdy model wyświetlany jest przez osobę na stole jednocześnie siedząc obok. Model w skalach do wartości nie przekraczających 1:7, które urządzenia są w stanie zainicjować działają bez żadnych problemów.

Aplikacja była również testowana z wykorzystaniem inteligentnych okularów Moverio-200. W celu inicjalizacji AR aplikacja przechodziła kalibrację 3D dla lewego i prawego oka. Niedoskonałość kalibracji powodowała bardzo szybkie męczenie wzroku uniemożliwiając długotrwałą pracę i stworzenie działającej aplikacji ze względu na konieczność ciągłych poprawek na urządzeniu nie było możliwe.

WNIOSKI

W wyniku przeprowadzonych prac nie udało się uzyskać założonych na wstępie pracy celów. Obecny poziom technologiczny posiadanych urządzeń nie umożliwia spełnienie wszystkich założeń aplikacji. Możliwe jest również, iż oprogramowanie zawiera ograniczenia nie pozwalające wykorzystać go na dużych modelach, których rozmiary przekraczają kilka metrów. Działa prawidłowo przy wykorzystaniu go w środowiskach gdzie zarówno modele 3D jak i obiekty AR są stosunkowo małych rozmiarów umożliwiając tworzenie gier czy wirtualnych reklam w świecie AR. Wybór dobrze zapowiadającego się oprogramowania okazał się nietrafiony. Możliwym jest, że stwierdzone podczas testów ograniczenia nie występują w innych środowiskach programistycznych takich jak Unity lub w przypadku zastosowania modeli CAD, gdzie nie jest wymagane kodowanie modelu przez dedykowane oprogramowanie.

Brak możliwości inicjacji modelu w skali 1:1 przekreśla wszystkie główne założenia aplikacji do prowadzenia kursów w tej formie. Brak niezbędnej mocy obliczeniowej testowanych urządzeń oraz możliwe ograniczenia oprogramowania nie pozwalają zastosować aplikacji w środowisku o dużych rozmiarach jaki stanowi nawet tak mały statek powietrzny jakim jest samolot Diamond DA20.

Brak sensorów takich jak czujnik głębokości oraz jakościowo lepszych wbudowanych kamer ograniczała możliwości śledzenia znacznika i środowiska, co wykazały testy podczas których zanotowano dużą ilość błędów związanych z nieprawidłowym położeniem wyświetlanego modelu.

W wyniku przeprowadzonych prac można stwierdzić, że deklarowana przez producenta oprogramowania otwartość aplikacji i dostępność jej na wszystkie urządzenia mobilne jest przesadzona. Ze względu na ogromną różnorodność urządzeń większość z nich nie będzie w stanie poprawnie uruchomić się i działać z aplikacją.

W celu ostatecznego określenia możliwości praktycznego wykorzystania stworzonej aplikacji do wspomagania przeglądu samolotu DA20 należałoby sprawdzić jej działanie na najnowszych modelach smartfonów i tabletów o większej mocy obliczeniowej oraz w systemie IOS. Dopiero to wykazałoby czy technologia z obecnym oprogramowaniem jest gotowa na tak wymagające wyzwanie. Pozytywny skutek mogłoby odnieść zastosowanie nowych technik śledzenia z wykorzystaniem drogich urządzeń wykorzystujących czujniki głębi oraz skali szarości.

BIBLIOGRAFIA

1. Kęsy M.: Poszerzona rzeczywistość w praktyce inżynierskiej oraz kształceniu technicznym, Edukacja-Technika-Informatyka, UR, 2016.
2. Rypulak A. (2016). Wykorzystanie rozszerzonej rzeczywistości do praktycznego szkolenia personelu lotniczego. TTS Technika Transportu szynowego, R. 23, nr 12, 381-385.
3. Stasiewicz A. (2015). Android Studio. Podstawy tworzenia aplikacji. Wydawnictwo Helion.
4. Android software development. Pozyskano z: www.wikipedia.org/wiki/Android_SDK
5. Instrukcja użytkownika w locie samolotu DA 20-C1, nr dokumentu: DA202-C1, 2010 r.
6. How to Use 3D Scan Object Recognition In Unity. Pozyskano z: <https://library.vuforia.com/articles/Solution/How-To-Use-Object-Recognition-in-Unity.html>
7. How to Use the Trackable Base Class. Pozyskano z: <https://library.vuforia.com/articles/Solution/How-To-Use-the-Trackable-Base-Class.html>

8. Wikitude Support. Pozyskano z <http://www.wikitude.com/external/doc/documentation/latest/android/>

Use of augmented reality environment for practical aviation staff training

The article describes the course of works aimed at creating a mobile application using Mobile Augmented Reality technology supporting the training process of personnel servicing aircraft in the field of pre-flight aircraft inspections. The requirements for the application, selection of programming tools and mobile imaging devices are presented. At the end of the article the results of tests of applications on various mobile devices and their conclusions are presented.

Autorzy:

dr inż. **Andrzej Rypulak** – Wyższa Szkoła Oficerska Sił Powietrznych w Dęblinie, Katedra Awioniki i Systemów Sterowania.

inż. **Sebastian Kuźmicz** – Wyższa Szkoła Oficerska Sił Powietrznych w Dęblinie.

JEL: L93 **DOI:** 10.24136/atest.2018.159

Data zgłoszenia: 2018.05.24 **Data akceptacji:** 2018.06.15