



Solução de rastreamento com três graus de liberdade para o Project North Star utilizando tecnologias abertas

Franklin Luiz do Nascimento Fracaro
Univ. Estadual do Oeste do Paraná
Foz do Iguaçu, Brasil
frafracaro@gmail.com

Fabiana Frata Furlan Peres
Univ. Estadual do Oeste do Paraná
Foz do Iguaçu, Brasil
fabiana.peres@unioeste.br

Claudio Roberto Marquetto Mauricio
Univ. Estadual do Oeste do Paraná
Foz do Iguaçu, Brasil
claudio.mauricio@unioeste.br

Abstract—Augmented reality is a lot more present in society nowadays, however, glasses like Hololens from Microsoft have an excessively prohibitive price, thus the open project with GLP 3.0 license, Project North Star, is an accessible alternative. The modularity of the project allows other technologies to be linked to it, amongst them, can be mentioned a tracking system based on a MIMU (Magnetic and Inertial Measurement Unit) that allows the user to move his head while using the device with three degrees of freedom. This work presents the implementation of a tracking system solution based on a MIMU sensor for one of the versions of Project North Star, assembled as described in Fracaro, Frata and Mauricio [1], using an Arduino Nano as support.

Resumo— A realidade aumentada está cada vez mais presente na sociedade; porém óculos como o Hololens da Microsoft possuem preços demasiadamente proibitivo. Desta forma o projeto aberto com licença GLP 3.0, Project North Star é uma alternativa acessível deste tipo de tecnologia. A modularidade do projeto permite que diversas outras tecnologias sejam associadas a ele. Dentre elas pode-se citar um sistema de rastreamento baseado em um MIMU (*Magnetic and Inertial Measurement Unit*) que permite o usuário se movimentar utilizando o dispositivo com três graus de liberdade. O presente trabalho apresenta a implementação de uma solução de sistema de rastreamento baseado em MIMU para uma das versões do Project North Star, montado conforme descrito em Fracaro, Frata e Mauricio [1], utilizando um Arduino Nano como apoio.

Palavras-chave—Realidade Aumentada; Rastreamento; Magnetic and Inertial Measurement Unit (MIMU).

I. INTRODUÇÃO

A utilização de *Head Mounted Displays* (HMDs) como dispositivos de visualização da Realidade Aumentada (RA) tende a não ser muito acessível em países em desenvolvimento como o Brasil devido ao preço dos dispositivos comerciais como o Hololens [2] e Magic Leap [3]. O surgimento de um projeto open source com licença GLP 3.0 chamado Project North Star gerou novas oportunidades.

Deve-se levar em consideração que a popularização de *smartphones* e *tablets* alavancou o desenvolvimento de pesquisas e a utilização da RA para dispositivos móveis [4]. A principal forma utilizada para aumentar a realidade em dispositivos móveis é o *video see-through*. Porém a utilização destes aparelhos, em muitas aplicações, dificulta a interação do usuário e apresenta um baixo campo de visão com

o processamento da cena, resolução e contraste da imagem real sendo diretamente impactados pelo hardware disponibilizado pelo dispositivo. Estas limitações geralmente tornam um HMD do tipo *video see-through* baseado em dispositivos móveis, inviável para algumas aplicações [5]. Os HMDs do tipo *optical see-through* também apresentam diversos problemas, como por exemplo, dificuldade em ocultar objetos e construção de versões de HMDs compactas [6]. No entanto, de acordo com Azuma [5], solucionando esses problemas, a abordagem *optical see-through* têm maior chance de tornar-se a principal abordagem de visualização utilizada em aplicações de RA voltadas ao consumidor.

Com a montagem da versão exiii do Project North Star [1], foi possível identificar diversas possibilidades para expandir e melhorar os óculos. Uma dessas possibilidades foi à inclusão de um sistema de rastreamento da pose (posição e orientação) da cabeça do usuário baseado em um sensor MIMU (*Magnetic and Inertial Measurement Unit*) e um Arduino Nano. Desta forma, este trabalho apresenta o contexto teórico e prático que permitiu a inclusão deste sensor no óculos montado anteriormente pelos autores.

Para apresentação das questões relacionadas a este trabalho definiu-se a seguinte estrutura: na Seção 2 são apresentados os pilares da fundamentação teórica; a Seção 3 descreve os materiais e métodos utilizados para o desenvolvimento da solução proposta; a Seção 4 detalha o problema que o sistema de rastreamento proposto aborda e como ele foi implementado; por fim na última seção são apresentadas as considerações finais.

II. FUNDAMENTAÇÃO TEÓRICA

A. Realidade Aumentada

Os requisitos básicos de um sistema de realidade aumentada, segundo Azuma [7] são: combinar conteúdo real e virtual; ser interativo em tempo real; e registrar em 3D os conteúdos virtuais. Estes requisitos não impõem limitações quanto às tecnologias utilizadas [8]. Entre os possíveis dispositivos que permitem o usuário visualizar o ambiente aumentado, os HMDs foram os mais comuns durante vários anos [9] na área de pesquisa e desenvolvimento. A principal vantagem deste tipo de dispositivo é a liberdade das mãos do usuário, já que o dispositivo é anexado a cabeça e dispensa que ele segure para perceber o ambiente aumentado. Os HMDs podem ser caracterizados em relação

a sua forma de visualização em duas abordagens principais: visão de câmera de vídeo (*video see-through*) e visão óptica direta (*optical see-through*) [7].

Os HMDs *optical see-through* (Fig. 1 (a)) utilizam combinadores óticos para apresentar as informações virtuais ao usuário. Isso pode ocorrer via projeção de conteúdos virtuais em lentes semi-translúcidas [10] ou diretamente na retina do usuário [11]. No *video see-through* (Fig. 1 (b)), a imagem do mundo real visualizada pelo usuário é capturada por uma ou mais câmeras, estas imagens são então processadas adicionando-se os objetos virtuais; o resultado é exibido no dispositivo de visualização [8].

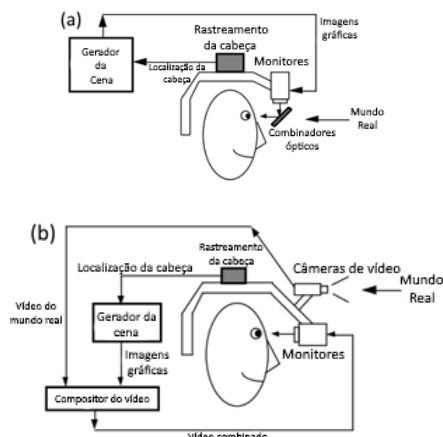


Fig. 1. *Optical See-through* (a) e *Video See-through* (b) [7]

Uma das maiores dificuldades da realidade aumentada utilizando a abordagem *see-through* é o alto custo dos dispositivos disponíveis comercialmente. Embora diversas propostas de dispositivos tecnológicos tenham surgido ao longo dos anos, as limitações quanto à forma de visualização, interação, poder de processamento e principalmente custo, tem impedido a disseminação desta tecnologia. De acordo com Azuma [5], solucionando esses problemas, esta abordagem tem maior chance de tornar-se a principal forma utilizada em aplicações voltadas ao consumidor. Nesse contexto os projetos abertos são de grande interesse visto que são tecnologias financeiramente mais acessíveis do que os dispositivos comerciais.

B. Project North Star

O Project North Star [12] é um projeto aberto com licença GNU General Public License 3.0 [13] de um dispositivo de realidade aumentada do tipo *optical see-through* de baixo custo, que utiliza impressão 3D e componentes eletrônicos básicos encontrados em lojas de equipamentos eletrônicos em sua composição [14]. O projeto foi disponibilizado pela então startup desenvolvedora do Leap Motion Controller [15] atual Ultraleap [16], em junho de 2018.

Em sua versão de teste antes de ser distribuído para o público (Fig. 2), o HMD utilizava duas telas LCD de 120Hz de 5.5" com resolução de 1440x2560. Cada olho visualizava conteúdo digital com aproximadamente 105° verticalmente por 75° horizontalmente com uma sobreposição estérea de 60%, resultando em um campo de visão de 105° por 105° [17].

A versão mais compacta disponibilizada para o público, utiliza duas telas LCD de 120Hz de 3.5" com resolução de

1600x1440, mantendo um campo de visão de mais de 100° porém utilizando uma versão industrial do Leap Motion, não disponível para pessoas físicas, que permite o rastreamento dos gestos do usuário em um campo de 180° por 180° em comparação ao campo de visão de 120° por 150° da versão comercial [14].



Fig. 2. Protótipo funcional do Project North Star [17]

Com a abertura do projeto para o público através da licença GPL 3.0 surgiram outras versões [18], [19], [20], [21] e atualizações no projeto original [22], [23]. As novas versões apresentam propostas para adaptar componentes mecânicos e de hardware além de permitir a utilização da versão comercial do Leap Motion Controller ao invés da versão industrial, o que acabou reduzindo o custo para a construção do dispositivo [1].

Parte dos softwares necessários, tais como para projetar os objetos virtuais, rastrear gestos do usuário e projetar conteúdo nas telas de LCD estão disponíveis no GitHub [24] e são dependentes da game engine Unity [25].

As partes mecânicas que compõem o dispositivo especificado no Project North Star podem ser vistas na Fig. 3, sendo elas: duas telas LCD direcionadas a duas lentes com revestimento espelhado que refletem 50% e transmitem 50% da luz. Essas lentes são responsáveis por combinar as imagens virtuais. Para intermediar o processo de interação entre os objetos virtuais e o usuário, é utilizado o Leap Motion Controller. Estes componentes são acoplados a uma estrutura plástica impressa em uma impressora 3D.

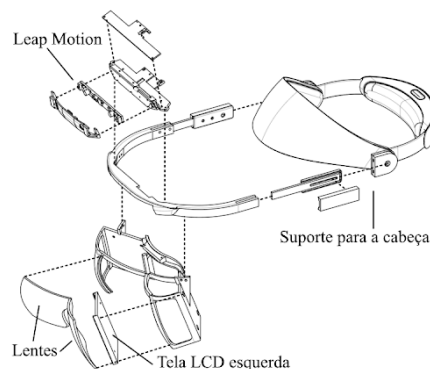


Fig. 3. Esquema do Project North Star [17]

C. Rastreamento

A base da realidade aumentada é a capacidade do sistema em identificar onde os elementos virtuais vão aparecer na cena [11], dependendo da aplicação e/ou tecnologia, a pose (posição e orientação) do usuário em relação a uma âncora no mundo real deve ser determinada para que os elementos virtuais sejam posicionados corretamente [8]. O processo de rastreamento (registro e rastreamento) pode abranger uma ou duas fases: Fase de

registro, responsável por determinar a pose do usuário em relação a uma âncora no mundo real, que pode ser um objeto, imagens ou sensores acoplados ao usuário ou objetos; Fase de rastreamento atualiza a pose do usuário em relação à pose anterior [8], [11].

Sistemas de realidade aumentada podem ser classificados quanto à sua forma de rastreamento, destacando-se a baseada em visão, baseada em sensores e híbrida [11]. Técnicas de rastreamento baseadas em visão atualmente utilizam métodos de processamento de imagens para calcular a pose da câmera relativa à objetos do mundo real [26], uma forma bastante comum de RA baseada em visão é a utilização de marcadores fiduciais, que geralmente são imagens com características específicas que são utilizadas como pontos de referência para os algoritmos que calculam a pose da câmera [11].

As técnicas de rastreamento baseado em sensores empregam um ou mais sensores, como por exemplo sensores magnéticos, acústicos, inerciais, ópticos e/ou mecânicos, para identificar a pose de um objeto ou pessoa [26]. O rastreamento inercial, por exemplo, utiliza sensores IMUs (*Inertial Measurement Units*) como acelerômetro e giroscópio para determinar a orientação, rotação e velocidade de um objeto. Estes sensores são suscetíveis à pequenos desvios de precisão em seus resultados com o passar do tempo, sendo necessário a aplicação de filtros para a redução destes desvios, o que pode levar à uma redução no tempo de resposta e atualização do sistema de rastreamento [8].

Sistemas híbridos combinam os dados de diversos sensores com técnicas de visão computacional, desta forma, diversos problemas encontrados em ambas as abordagens, como desvios de precisão de sensores inerciais, podem ser minimizados e ou até mesmo eliminados além de permitir uma melhora considerável na precisão do rastreamento [8], [11].

Outra propriedade importante das abordagens de rastreamento refere-se a quantos graus de liberdade (*Degrees of freedom* - DOF) é possível medir. A posição e orientação de um objeto podem ser especificadas utilizando seis graus de liberdade, três para translação nos eixos x, y e z e três para a rotação que ocorre nestes eixos, *pitch*, *roll* e *yaw*, com o *pitch* correspondendo a uma rotação no eixo *right-left*, o *roll* no eixo *back-forward* e *yaw* no eixo *up-down* respectivamente, como pode ser visto na Fig. 4. Levando em consideração HMDs, 3-DOF (três graus de liberdade) permite que um usuário olhe ao redor do mundo a partir de um ponto fixo. Com 6-DOF (seis graus de liberdade) o usuário poderá movimentar-se na cena como na vida real, isto é, não só a sua posição no mundo será rastreada como também as rotações realizada por sua cabeça [27].

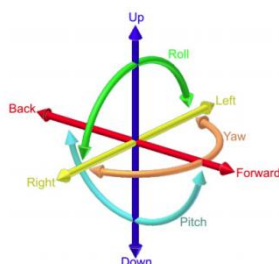


Fig. 4. Movimentação com seis graus de liberdade [27]

D. Fusão de sensores

Fusão de sensores ou AHRS (*Attitude and Heading Reference System*) são algoritmos utilizados para refinar dados e corrigir erros provenientes de MIMUs (*Magnetic and Inertial Measurement Units*), também conhecidos como MARG (*Magnetic, Angular Rate e Gravity*), através da leitura e união de dados de dois ou mais sensores. MIMUs são compostos de três sensores que podem ser utilizados para estimar a orientação e deslocamento de um corpo/objeto em tempo real: acelerômetro, giroscópio e magnetômetro [28],[29].

Os dados dos sensores de um MIMU podem ser utilizados sem nenhum tratamento, como por exemplo, o acelerômetro pode ser utilizado para computar a orientação, o magnetômetro oferece uma estimativa do ângulo do movimento de *yaw* e o acelerômetro oferece dados que podem ser utilizados para calcular mudanças na orientação [29]. Utilizando estes dados sem tratamento acaba resultando em problemas de acurácia que podem variar entre erros leves a graves, como ruídos eletrônicos intrínsecos ao próprio dispositivo, ruídos provocados por mudanças na temperatura, presença de objetos magnéticos que interferem com as leituras do magnetômetro, acumulação de erros provenientes do cálculo da mudança de orientação pelo acelerômetro, entre outros [30].

Um algoritmo AHRS geralmente pode ser dividido em duas partes: cálculo da orientação baseado nos dados de um giroscópio e cálculo da orientação baseado no acelerômetro e magnetômetro. Cada parte oferece uma estimativa da orientação; estes valores devem então ser fundidos, levando em consideração as peculiaridades de cada sensor utilizado como fonte para a coleta de dados, permitindo assim obter-se um resultado mais preciso da orientação [28].

Os algoritmos Madgwick e Mahony são bastante utilizados para realizar este processo de filtragem devido à seu baixo custo computacional. Madgwick utiliza uma representação em quaternião, número complexo com quatro dimensões, que permite que os dados de um acelerômetro e magnetômetro sejam utilizados em uma derivada analítica e em um algoritmo de otimização descida de gradiente para computar a orientação de um MIMU [28], [31]. Mahony é um tipo de filtro complementar que é capaz de estimar a pose de um corpo através da fusão da integração das medidas de um giroscópio e das medidas de direção de um acelerômetro. Através do cálculo de erros e a propagação do resultado da correção destes erros através de uma integral proporcional, é possível corrigir a velocidade angular medida pelos sensores, o que resulta em uma estimativa da pose mais confiável [28], [32]. Deve-se levar em consideração que estes dois filtros estimam corretamente a orientação apenas em situações ideais, onde não há uma aceleração significativa ou distúrbios magnéticos [28].

III. MATERIAIS E MÉTODOS

Um dos passos previsto no trabalho de Fracaro, Peres e Mauricio [1] para a evolução do óculos montado, estava a implementação de um IMU (*Inertial Measurement Unit*) para realizar o rastreamento da posição da cabeça do usuário.

Uma das versões mais atuais do Project North Star, Deck Mk.1 [20], faz uso do sensor de rastreamento Intel RealSense T261 [33], que realiza a junção dos dados de câmeras de profundidade e um IMU para obter a pose do usuário com seis graus de liberdade. No entanto, o preço do dispositivo é proibitivo. Kosaka [34] apresentou uma solução viável financeiramente, utilizando um IMU. Nesta solução ele faz a leitura dos dados do sensor de seis graus de liberdade (MPU-6050) através de um Arduino Nano, possibilitando ao usuário do óculos uma movimentação da cabeça com três graus de liberdade. No entanto, conforme Kosaka [34] descreve, a utilização de um IMU ao invés de um MIMU prejudicou o resultado final, visto que o erro cumulativo proveniente do acelerômetro e giroscópio fez com que o sistema tivesse de ser resetado após certo tempo de uso para que a pose voltasse a ser rastreada normalmente.

Para tentar minimizar os problemas de precisão que afetam o rastreamento com IMUs, utilizou-se o sensor MPU-9250 [35], um MIMU, que diferente do MPU-6050, um IMU, pois possui um magnetômetro que pode ser utilizado para melhorar os resultados provenientes do acelerômetro e giroscópio através da aplicação de um algoritmo de fusão de sensores como o Magdewick ou Mahony.

Foram utilizadas duas bibliotecas para auxiliar a comunicação entre o sensor, o Arduino e o óculos; a biblioteca MPU-9250 [36] e a Arduino Serial Command [37]. Essas bibliotecas foram adicionadas ao sketch da IDE Arduino [38].

A parte de software responsável por gerar as cenas nas telas do óculos está totalmente atrelada à game engine Unity [25] versão 2018.2 ou superior, sendo necessário também a instalação dos módulos do Leap Motion, sensor responsável pelo rastreamento das mãos do usuário, possibilitando a interação com objetos virtuais [39] além da IDE Visual Studio Code [40]. O sistema executado através do Unity é o responsável pelo registro e rastreamento da cabeça do usuário utilizando os dados do MIMU recebidos do Arduino Nano[41] como pode ser visto na Fig. 5.

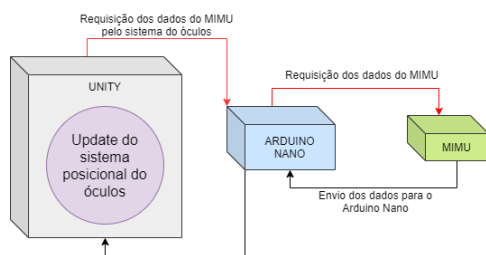


Fig. 5. Diagrama do sistema Unity-Arduino-MIMU

Para o desenvolvimento da solução, foram utilizados os seguintes materiais:

- Óculos de RA Project North Star [1];
- Unity 3D (versão 2018.4 ou superior);
- Visual Studio Code;
- Arduino IDE;
- Arduino Nano;
- MIMU (MPU-9250/6500);
- Um metro de cabos de fios par-traçados;

- Conector mini-USB de 2 metros;
- Duas mini protoboards (170 pontos);
- Biblioteca MPU-9250;
- Biblioteca Arduino Serial Command;

Além destes materiais foi utilizado um computador *desktop* com um processador i3 de 3,7 GHz, 8GB de memória DDR4 e uma placa de vídeo ATI Radeon RX550 de 4GB.

IV. SISTEMA DE RASTREAMENTO DO PROJECT NORTH STAR

O óculos de realidade aumentada montado conta com o sensor Leap Motion para a detecção da pose das mãos do usuário do óculos, porém este sistema não oferece nenhuma forma de atualizar a pose de objetos virtuais em relação ao usuário; sua principal função é ofertar uma forma de interação com estes objetos. Desta forma, a menos que o usuário movimente estes objetos manualmente durante um movimento de rotação, o óculos apresentará o comportamento conforme mostrado na Fig. 6, onde ao observar um objeto virtual no mundo real (Fig. 6 (a)) e realizar um movimento de rotação de 180° (*yaw*), o objeto virtual irá seguir o movimento (Fig. 6 (b)) pois não existe nenhum sistema rastreamento a pose deste usuário. A Fig. 6 (c) apresenta o comportamento esperado do objeto virtual em relação a um movimento de rotação caso um sistema de rastreamento estivesse presente no óculos, permitindo assim criar a ilusão de que este objeto está ancorado e realmente existe no mundo real.

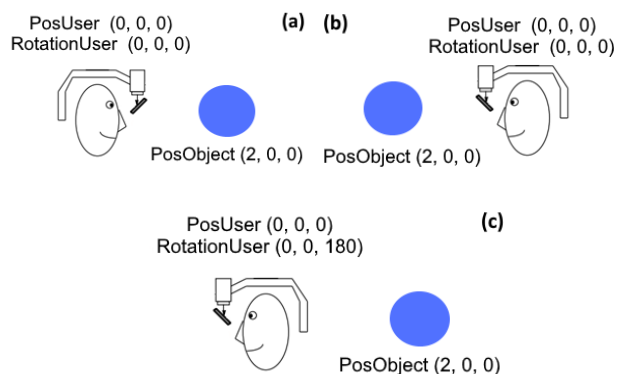


Fig. 6. Exemplo de visualização de um objeto virtual no mundo real: (a) posição inicial do usuário e do objeto virtual; (b) usuário rotacionado 180°, sem um sistema de rastreamento; (c) usuário rotacionado 180°, com um sistema de rastreamento.

A. Sistema de rastreamento implementado

Com os materiais descritos anteriormente na seção Materiais e Métodos, foi possível montar um circuito seguindo as informações oferecidas pelo *datasheet* dos componentes [35], [41], o que resultou no diagrama da Fig. 7. O MIMU foi acoplado no topo dos óculos, acima das lentes e do Leap Motion como pode ser visto na Fig. 8.

O MIMU utiliza apenas 3.3 Volts, porém, possui um adaptador interno que permite a utilização de uma fonte com 5 Volts, além disto, é possível escolher entre duas interfaces de comunicação, I²C (*Inter-Integrated Circuit*) e SPI (*Serial Peripheral Interface*), ambas síncronas dependentes de um sinal de clock compartilhado entre o dispositivo mestre e seu parente, desta forma, utilizando a interface I²C através das portas SCL (*Serial Clock*) e SDA (*Serial Data*) no

MIMU, conectadas às portas A5 (SCL) e A4 (SDA) do Arduino Nano, foi possível transmitir informações entre os dispositivos. Devido à interface de comunicação escolhida (I²C), o tamanho dos fios que conectam o MIMU ao Arduino Uno não deve ser longo, pois pode afetar os valores a serem coletados.

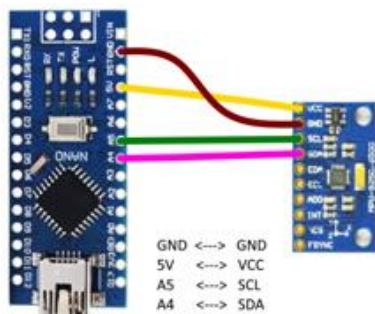


Fig. 7. Diagrama das conexões entre o Arduino e o MIMU

Para a captura dos dados do MIMU foi utilizado a biblioteca MPU-9250, pois ela oferece diversas funcionalidades além de simplificar aquisição de dados de sensores inerciais e magnéticos. Dentre estas funcionalidades estão dois filtros de fusão de sensores, Mahony e Madgwick. Optou-se pela utilização do filtro Madgwick, devido à ele levar em consideração os dados do magnetômetro. A biblioteca Arduino Serial Command foi utilizada para facilitar o recebimento e envio de mensagens entre o Arduino e o Unity, devido a ela oferecer wrappers que facilitam o parsing e tokenização de comandos enviados pela game engine ao Arduino. A Fig. 9 apresenta um exemplo de como esta comunicação pode ser feita. Estas funções foram adaptadas e unidas ao código de exemplo de coleta de dados de um MIMU da biblioteca MPU-9250 (MPU9250BasicAHRS.ino) de forma que o sistema resultante fosse capaz tanto de receber e enviar mensagens para o Unity quanto ler e melhorar a precisão dos valores lidos do MIMU através do filtro de Madgwick.

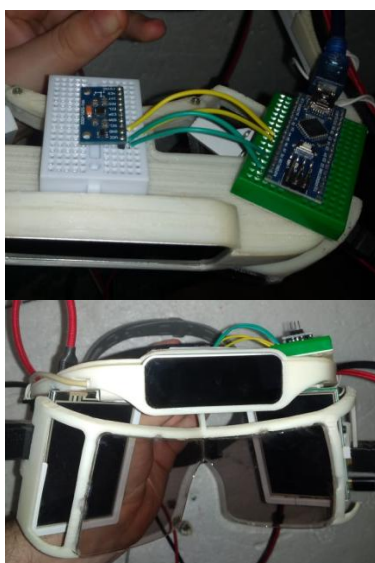


Fig. 8. Arduino e MIMU acoplados ao óculos

Além da adaptação do código apresentado na Fig. 9, outras modificações foram feitas no exemplo

“MPU9250BasicAHRS.ino”, dentre elas está a remoção de todo o código relacionado à displays de LCD além de todos os cálculos realizados após a chamada do filtro de Madgwick, pois este apresenta como resultado final os quatros valores que representam um quaternião que pode ser utilizado pela API do Unity para a representação de rotações.

O filtro Madgwick recebe os dados que foram lidos pelo sensor e retorna quatro valores de ponto flutuante que representam um quaternião com as informações da pose do usuário. Estes quatro valores são transformados em strings e concatenados com o símbolo “ ; ” sendo utilizado para a separação. A string resultante representa a mensagem que será enviada do Arduino para a game engine.

```
#include <SerialCommand.h>
#include <SoftwareSerial.h>

SerialCommand SCmd;

void setup() {
    Serial.begin(9600);

    while(!Serial);

    SCmd.addCommand("HELLO", respondCmd);
}

void loop() {
    SCmd.readSerial();
}

void respondCmd(const char *command) {
    Serial.println("WORLD");
    delay(100);
}
```

Fig. 9. Sistema de comunicação entre o Arduino e a game engine

A comunicação entre a game engine Unity 3D e o Arduino pode ser vista na Fig. 10, que apresenta os métodos utilizados durante este processo como o SendMsgToArduino e ReceiveMsgFromArduino, que são utilizados para enviar e receber as mensagens do Arduino. Todo o processo de comunicação é realizado dentro da co-rotina AsyncAccessArduino através da biblioteca padrão do C#, *Sytem.IO.Ports*, que oferece acesso à porta serial e permite o streaming de mensagens entre os dispositivos.

As mensagens recebidas pela game engine apresentam o seguinte formato: “valor1;valor2;valor3;valor4”, onde cada campo valorX representa uma parte de um quaternião. Esta mensagem passa por um processo de parsing e cada valor é transformado de string para ponto flutuante e adicionados a uma instância da classe Quaternion que é repassado para a classe HMD_Manager para ser aplicado como uma rotação no óculos através da co-rotina AsyncSendValuesToHMD_Manager.

```
public void SendMsgToArduino(string msg)...

public string ReceiveMsgFromArduino()...

public IEnumerator AsyncAccessArduino
(Action<string> results, Action fail = null,
float timeout = float.PositiveInfinity)...

public Quaternion ParseMsg(string msg)...

public IEnumerator AsyncSendValuesToHMD_Manager
(Quaternion valuesFromIMU)...
```

Fig. 10. Métodos responsáveis pela comunicação entre o Unity e o Arduino

Os valores do quaternião são adicionados pelo HMD_Manager ao transform do objeto que representa o óculos na cena da game engine através do método ApplyRotation que é chamado a cada vez que o HMD_Manager recebe uma nova mensagem do AsyncSendValuesToHMD_Manager, como pode ser visto na Fig. 11.

```
[SerializeField]
GameObject Oculos;

[SerializeField]
float smoothValue = 2.0f;

void ApplyRotation(Quaternion IMUValues)
{
    Oculos.transform.rotation = Quaternion.Slerp(
        Oculos.transform.rotation, IMUValues,
        Time.deltaTime * smoothValue);
}
```

Fig. 11. Trecho de código responsável por atualizar a rotação do óculos.

Como resultado o usuário do óculos poderá movimentar a cabeça livremente sem que os objetos virtuais repitam o mesmo movimento, como pode ser visto na Fig. 12, onde três cubos foram posicionados à frente do usuário, que inicialmente vê na lente esquerda os blocos verde e azul (Fig. 12 (a)) e na lente direita parte do bloco azul e o bloco vermelho, porém ao rodar a cabeça para à direita, o bloco verde desaparece do campo de visão da lente esquerda com parte do bloco azul e vermelho podendo ser observados nesta lente (Fig. 12 (b)).

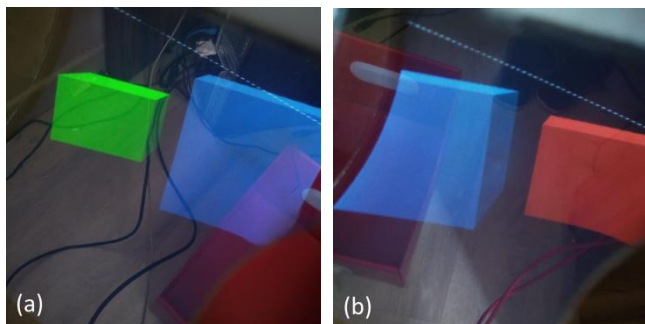


Fig. 12. Resultado de um movimento de rotação (yaw) em uma das lentes do óculos: objetos virtuais refletidos na lente esquerda (a) e objetos virtuais após a rotação na lente esquerda(b)

Levando em consideração o custo original de U\$ 195,00 para a construção da versão exiii do Project North Star reportado em Fracaro, Frata e Mauricio [1], os componentes para a montagem do sistema de rastreamento (Arduino Nano, MIMU, protoboards e cabos) acabam adicionando U\$ 27,00 a este valor, resultando em um custo total de U\$ 222,00.

V. CONSIDERAÇÕES FINAIS

O Project North Star, por ser um projeto aberto e de baixo custo, consegue oferecer oportunidade de acesso à dispositivos de realidade aumentada do tipo *optical see-through*, para um número maior de pessoas. Sua modularidade garante que diversas tecnologias possam ser combinadas a ele, permitindo que o óculos seja adaptado a realidade financeira de cada usuário.

A adição do rastreamento a partir dos dados de um MIMU permitiu que o requisito básico “registrar em 3D os conteúdos virtuais” de um sistema de realidade aumentada apresentado por Azuma [7] fosse incorporado aos óculos, visto que o rastreamento das mãos oferecido pelo Leap Motion funciona somente quando as mãos do usuário está dentro do alcance do sensor. Além disto, deve-se levar em consideração que o sistema implementado oferece uma alternativa de baixo custo para o rastreamento baseado em fusão de sensores.

Algumas partes do sistema de rastreamento apresentado podem ser melhoradas, podendo-se citar o sistema de troca de mensagens, que se apoia, em grande parte, em co-rotinas, que se fosse executado em sua própria thread, permitiria que alguns problemas em relação à taxa de atualização de rotações bruscas do óculos sejam minimizados.

Levando em consideração o projeto como um todo, é importante citar o seu imenso potencial para ser utilizado como uma ferramenta de ensino-aprendizagem, principalmente para o público universitário devido a grande quantidade de áreas da computação que são e podem ser abordadas com o óculos, podendo-se citar visão computacional, robótica, engenharia de software, algoritmos, inteligência artificial além da própria área de realidade aumentada.

REFERÊNCIAS

- [1] F. Fracaro, F. F. F. Peres, and C. R. M. Mauricio, “Montando o Project North Star: Um Dispositivo de Visualização de Baixo Custo Baseado em Visão Óptica Direta,” in XXI SIMPÓSIO DE REALIDADE VIRTUAL E AUMENTADA, 2019.
- [2] Microsoft, “Microsoft HoloLens | Mixed Reality Technology for Business”, Microsoft.com, 2020. [Online]. Available: <https://www.microsoft.com/en-us/hololens>. [Accessed: 20- Nov- 2020].
- [3] Magic Leap, “Magic Leap | Reality is just beginning”, MagicLeap.com, 2020. [Online]. Available: <https://www.magicleap.com/en-us>. [Accessed: 20- Nov- 2020].
- [4] man and J. Swan II, “A Systematic view of Usability Studies in Augmented Reality between 2005 and 2014”, 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), 2016.
- [5] R. Azuma, “The road to ubiquitous consumer augmented reality systems”, Human Behavior and Emerging Technologies, vol. 1, no. 1, pp.26-32, 2019.
- [6] R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, “Recent advances in augmented reality”, IEEE Computer Graphics and Applications, vol. 21, no. 6, pp. 34-47, 2001.
- [7] R. Azuma, “A Survey of Augmented Reality”, Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385, 1997.

- [8] M. Billinghurst, A. Clark and G. Lee, "A Survey of Augmented Reality", *Transactions and Trends in Human-Computer Interaction*, vol. 8, no.2-3, pp. 73-272, 2015.
- [9] O. Bimber and R. Raskar, "Modern approaches to augmented reality", *ACM SIGGRAPH 2005 Course on - SIGGRAPH '05*, 2005.
- [10] R. Roberto and V. Teichrieb, "Desenvolvimento de Sistema de Realidade Aumentada Projetiva com Aplicação em Educação", M.S. thesis, Universidade Federal de Pernambuco, 2012.
- [11] R. Tori and M. S. Hounsell, Eds. *Introdução a Realidade Virtual e Aumentada*. Porto Alegre: Editora SBC, 2020.
- [12] L. Motion, "Project North Star—Leap Motion Developer", Leap Motion Developer, 2019. [Online]. Available: <https://developer.leapmotion.com/northstar>. [Accessed: 20- Nov- 2020].
- [13] Free Software Foundation, "The GNU General Public License v3.0 - GNU Project - Free Software Foundation", Gnu.org, 2019. [Online]. Available: <https://www.gnu.org/licenses/gpl-3.0.html>. [Accessed: 20- Nov- 2020].
- [14] L. Motion, "Project North Star is Now Open Source - Leap Motion Blog", Leap Motion Blog, 2019. [Online]. Available: <http://blog.leapmotion.com/north-star-open-source/>. [Accessed: 20- Nov- 2020].
- [15] L. Motion, "Leap Motion", Leap Motion, 2019. [Online]. Available: <https://www.leapmotion.com/>. [Accessed: 20- Nov- 2020].
- [16] Ultraleap, "Digital worlds that feel human | Ultraleap", Ultraleap.com, 2020. [Online]. Available: <https://www.ultraleap.com/>. [Accessed: 20- Nov- 2020].
- [17] D. Holz and D. Holz, "Our Journey to the North Star", Leap Motion Blog, 2019. [Online]. Available: <http://blog.leapmotion.com/our-journey-to-the-north-star/>. [Accessed: 20- Nov- 2020].
- [18] "How to Build Project North Star Simplified Version", exiii | Haptic Wearable Technology for XR, 2019. [Online]. Available: https://exiii.jp/2018/07/25/project_north_star_en/. [Accessed: 20- Nov- 2020].
- [19] "How to Build Project North Star Simplified ver.2", exiii | Haptic Wearable Technology for XR, 2019. [Online]. Available: <https://exiii.jp/2018/12/07/project-north-star-simplified-v2-en/>. [Accessed: 20- Nov- 2020].
- [20] "Project North Star Headsets & Components | CombineReality", Combine Reality, 2020. [Online]. Available: <https://combinereality.com/>. [Accessed: 20- Nov- 2020].
- [21] "Triton", Leap-27, 2020. [Online]. Available: <https://leap-27.github.io/TritonProject-Guide/#>. [Accessed: 20- Nov- 2020].
- [22] F. Maurer, "Project North Star: Mechanical Update 1 - Leap Motion Blog", Leap Motion Blog, 2019. [Online]. Available: <http://blog.leapmotion.com/project-north-star-mechanical-update-1/>. [Accessed: 20- Nov- 2020].
- [23] F. Maurer, "Project North Star: Mechanical Update 3 - Leap Motion Blog", Leap Motion Blog, 2019. [Online]. Available: <http://blog.leapmotion.com/project-north-star-mechanical-update-3/>. [Accessed: 20- Nov- 2020].
- [24] "leapmotion/ProjectNorthStar", GitHub, 2019. [Online]. Available: <https://github.com/leapmotion/ProjectNorthStar>. [Accessed: 20- Nov- 2020].
- [25] U. Technologies, "Unity - Unity", Unity, 2020. [Online]. Available: <https://unity.com/>. [Accessed: 20- Nov- 2020].
- [26] F. Zhou, H. Duh and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Cambridge, 2008, pp. 193-202.
- [27] G. Koulouris, K. Akşit, M. Stengel, R. Mantiuk, K. Mania and C. Richardt, "Near-Eye Display and Tracking Technologies for Virtual and Augmented Reality", *Computer Graphics Forum*, vol. 38, no. 2, pp. 493-519, 2019.
- [28] E. Diaz, F. Müller, A. Jiménez and F. Zampella, "Evaluation of AHRS algorithms for inertial personal localization in industrial environments," 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, 2015, pp. 3412-3417.
- [29] M. Nazarahari , H. Rouhani, "40 Years of Sensor Fusion for Orientation Tracking via Magnetic and Inertial Measurement Units: Methods, Lessons Learned, and Future Challenges", *Information Fusion* (2020).
- [30] R. Bhardwaj, N. Kumar and V. Kumar, "Errors in micro-electro-mechanical systems inertial measurement and a review on present practices of error modelling", *Transactions of the Institute of Measurement and Control*, vol. 40, no. 1, 2017.
- [31] S. Madgwick, "An efficient orientation filter for inertial and inertial / magnetic sensor arrays". 2010.
- [32] R. Mahony, T. Hamel, J. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group". *IEEE Transactions on Automatic Control*, Institute of Electrical and Electronics Engineers, vol. 53, no. 5, pp.1203-1217.
- [33] I. Corporation, "Tracking Technology – Intel RealSense Depth and Tracking Cameras", Intel® RealSense™ Depth and Tracking Cameras, 2020.

- [Online]. Available: <https://www.intelrealsense.com/tracking/>. [Accessed: 20- Nov- 2020].
- [34] R. Kosaka, "Project North Star を 3DoF にする - ryokosaka blog", Ryokosaka.com, 2020. [Online]. Available: <https://ryokosaka.com/blog/181201.html>. [Accessed: 20- Nov- 2020].
- [35] InvenSense, "MPU-9250 | TDK", Invensense.tdk.com. [Online]. Available: <https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/>. [Accessed: 20- Nov- 2020].
- [36] K. Winer, "kriswiner/MPU9250", GitHub, 2020. [Online]. Available: <https://github.com/kriswiner/MPU9250>. [Accessed: 20- Nov- 2020].
- [37] S. Cogswell, "scogswell/ArduinoSerialCommand", GitHub, 2020. [Online]. Available: <https://github.com/scogswell/ArduinoSerialCommand>. [Accessed: 20- Nov- 2020].
- [38] Arduino, "Software | Arduino", Arduino, 2020. [Online]. Available: <https://www.arduino.cc/en/software>. [Accessed: 20- Nov- 2020].
- [39] L. Motion, "leapmotion/ProjectNorthStar - Software", GitHub, 2020. [Online]. Available: <https://github.com/leapmotion/ProjectNorthStar/tree/master/Software>. [Accessed: 20- Nov- 2020].
- [40] Microsoft, "Visual Studio Code - Code Editing Redefined", Code.visualstudio.com, 2020. [Online]. Available: <https://code.visualstudio.com/>. [Accessed: 20- Nov- 2020].
- [41] Arduino, "Arduino - Arduino Board Nano", Arduino.cc, 2020. [Online]. Available: <https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>. [Accessed: 20- Nov- 2020].