

# Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

---

## Projektová dokumentácia

*Vozidlo s obsluhou pomocou zariadenia virtuálnej reality*

**Číslo tímu:**

5

**Členovia tímu:**

Matejov Erik, Bc.  
Habovštiak Martin, Bc.  
Morvay Tomáš, Bc.  
Prágai Albert, Bc.  
Protuš Peter, Bc.  
Siro Miroslav, Bc.  
Trybulová Natália, Bc.

**Akademický rok:**

2014/2015

**Študijný program:**

Počítačové a komunikačné systémy a siete

**Študijný odbor:**

Počítačové inžinierstvo

# Anotácia

---

Projekt sa zaoberá možnosťou ovládania vozidla pomocou zariadenia virtuálnej reality. V projekte budú analyzované rôzne techniky potrebné pri neskoršom návrhu a implementácii. Jedná sa najmä o analyzovanie možností výroby vozidla, bezdrôtového ovládania a prenosu videa a spracovania videa na počítači do vhodného formátu pre zariadenie virtuálnej reality.

Cieľom projektu je vytvoriť model ovládaného vozidla s možnosťou zaznamenania, prenosu a reprezentácie prenosu 3D obrazu. Toto vozidlo bude využiteľné v mnohých oblastiach. Ide napríklad o odoberanie vzoriek z kontaminovaných území, detonácie náloží, hľadanie ľudí v troskách a mnoho ďalších. Práve zariadenie virtuálnej reality zlepší operátorovi vozidla orientáciu v priestore, zjednoduší ovládanie kamier a umožní mu sprostredkovať 3D obraz neprístupného miesta tak, ako keby sa tam reálne nachádzal.

Výsledkom projektu bude vytvorený model vozidla, ovládanie jeho pohybu, smerovanie kamier a spôsoby zaznamenávania, prenosu a reprezentácie 3D video obsahu.

# Obsah

---

<b>1</b>	<b>Úvod .....</b>	<b>1</b>
<b>2</b>	<b>Cieľ projektu .....</b>	<b>2</b>
2.1	Využitie.....	2
2.1.1	Veda .....	2
2.1.2	Vojenská technika .....	2
2.1.3	Zábavný priemysel.....	2
2.1.4	Záchranárske operácie.....	2
2.1.5	Vzdelávanie.....	2
<b>3</b>	<b>Analýza.....</b>	<b>3</b>
3.1	Kompresia a dekompresia videa .....	3
3.1.1	Motion JPEG.....	3
3.1.2	MPEG-4 a H.264.....	3
3.1.3	Multiview Video Coding .....	5
3.1.4	3D Video .....	5
3.1.5	Porovnanie hardvérovej a softvérovej kompresie .....	6
3.2	Prenos videa .....	6
3.2.1	Bluetooth.....	7
3.2.2	WiFi .....	7
3.2.3	HSDPA .....	8
3.2.4	LTE .....	8
3.3	Hardvér .....	8
3.3.1	Raspberry Pi.....	8
3.3.2	Arduino .....	9
3.3.3	PWM.....	9
3.3.4	DC motor .....	9
3.3.5	Servomotor.....	9
3.3.6	Krokový motor .....	9
3.3.7	H-mostík .....	9
3.3.8	Flyback diódy.....	10
3.4	Ovládanie .....	10
3.4.1	Vstup z gamepadu .....	11
3.5	Zariadenie virtuálnej reality .....	11

3.5.1	Oculus Rift .....	11
3.5.2	Alternatívy k zaradeniu Oculus Rift .....	12
3.6	Oculus SDK .....	13
3.6.1	Odporúčané kroky pri integrácii LibOVR .....	13
3.6.2	Senzory a sledovanie polohy hlavy .....	13
3.6.3	Renderovanie skreslenia .....	15
3.6.4	OpenGL vs. Direct3D .....	15
3.7	OpenGL (Open Graphics Library) .....	16
3.8	Softvérové spracovanie videa .....	18
3.8.1	OpenCV (Open Source Computer Vision) .....	18
3.8.2	AForge.Net.....	19
3.8.3	IVT (Integrating Vision Toolkit) .....	19
3.9	Existujúce riešenia .....	19
<b>4</b>	<b>Specifikacia požiadaviek.....</b>	<b>20</b>
4.1	Funkcionálne požiadavky .....	20
4.2	Nefunkcionálne požiadavky.....	20
4.2.1	Hardvérové požiadavky .....	20
4.2.2	Softvérové požiadavky.....	21
<b>5</b>	<b>Návrh.....</b>	<b>22</b>
5.1	Architektúra systému .....	22
5.2	Kompresia dát .....	22
5.3	Prenos videa .....	22
5.4	Hardvér .....	23
5.4.1	Riadenie vozidla.....	23
5.4.2	Pohon vozidla a pohyb kamery .....	23
5.4.3	Schéma zapojenia.....	24
5.5	Ovládanie .....	24
5.5.1	Spracovanie vstupu z gamepadu .....	25
5.5.2	Príprava dát pre vozidlo .....	25
5.5.3	Posielanie dát vozidlu .....	26
5.6	Zariadenie virtuálnej reality .....	26
5.7	Oculus SDK .....	26
5.7.1	Renderovanie skreslenia .....	26
5.7.2	OpenGL vs. Direct3D .....	26
5.8	Softvérové spracovanie videa .....	27

5.8.1	Priebeh spracovania obrazu na centrálnej stanici.....	27
5.9	Voľba implementačného prostredia .....	27
<b>6</b>	<b>Zhodnotenie .....</b>	<b>28</b>
	<b>Zoznam použitej literatúry .....</b>	<b>29</b>

# 1 Úvod

---

Koncept virtuálnej reality je známy už niekoľko desiatok rokov a to aj napriek tomu, že verejnosť sa oň začala viac zaujímať až po roku 1990. Už v roku 1962 postavil *Morton Heilig* prototyp známy ako *Sensorama*, ktorý umožňoval pri premietaní filmu vnímať okrem obrazu a zvuku aj vôňu. No od vtedy ubehlo veľa rokov a technika pokročila míľovými krokmi vpred. V súčasnosti sa virtuálna realita interpretuje ako počítačovo simulované prostredie, ktoré simuluje fyzickú prítomnosť na miestach v reálnom alebo vytvorenom svete.

Práve simulácia prítomnosti v reálnom svete môže mať široké využitie a je významné sa touto oblasťou zaoberať. Nami navrhovaný prístup premieta obraz reálneho sveta do zariadenia virtuálnej reality pomocou kamier z vozidla ovládaného na diaľku. Tento prístup má množstvo výhod, ktoré ho predurčujú byť úspešným v rôznych oblastiach.

V projekte plánujeme vytvoriť samonapájací model vozidla, ovládaný na diaľku. Toto vozidlo bude navrhnuté a skonštruované našim tímom tak, aby splnilo požiadavky potrebné pre ďalšie riešenie projektu a bolo ho možné okamžite použiť v reálnych situáciách.

Jednou z motivácií nášho tímu je možnosť prísť do kontaktu s modernými a perspektívnymi technológiami. Napríklad zariadenie *Oculus Rift*, ktoré plánujeme použiť, momentálne ešte nie je oficiálne uvedené na trhu. Na porovnanie technológií plánujeme vytvoriť riešenie aj pomocou zariadenia *Samsung Gear VR* a *Samsung Galaxy Note 4*. Zaujímavé je taktiež spojenie virtuálnej reality s hmotným svetom.

V neposlednom rade budú pri riešení projektu nadobudnuté mnohé cenné skúsenosti. Tie sa určite uplatnia nielen v ďalšom štúdiu, ale aj v neskoršej praxi. Tento projekt bude rovnako prínosom pre fakultu ako aj pre budúce generácie študentov, ktorí by mali záujem pokračovať vo výskume v oblasti virtuálnej reality, alebo dokonca rozšírenej reality. Taktiež plánujeme v prípade získania grantu zakúpiť zariadenia, ktoré sa budú môcť na fakulte používať na rôzne výskumné účely aj po nasledujúce roky.

S projektom sa v rámci propagácie fakulty plánujeme predstaviť na študentskej vedeckej konferencii *IIT SRC*. Takisto sme už prihlásení na súťaž *TP Cup*, ktorá predstavuje výzvu pre budúcich IKT profesionálov, ktorí si chcú zmerať svoje schopnosti s ostatnými a súčasne vytvoriť a „predať“ použiteľné riešenia.

## 2 Cieľ projektu

---

Cieľom projektu je vytvoriť vozidlo s obsluhou pomocou zariadenia virtuálnej reality. Na vozidle budú umiestnené kamery použité na zaznamenávanie 3D videa. Ovládané budú podľa polohy hlavy. Video sa bude prenášať do počítača kde spracuje pre zariadenie virtuálnej reality. Pohyb vozidla bude riadený pomocou gamepadu.

Operátor vozidla bude mať nasadené zariadenie virtuálnej reality v ktorom sa premieta obraz z kamier na vozidle. Gamepadom bude ovládať pohyb vozidla a pohybmi hlavy smerovanie kamier. Cieľom je umožniť operátorovi vozidla lepšiu orientáciu v priestore, zjednodušiť ovládanie kamier a sprostredkovať mu 3D obraz neprístupného miesta tak, ako keby sa tam reálne nachádzal.

### 2.1 Využitie

#### 2.1.1 Veda

Diaľkovo ovládané vozidlá majú rôzne vedecké využitie, ako napríklad práca v kontaminovaných prostrediach, práca hlboko v oceáne alebo prieskum vesmíru. Polícia ich používa na detekciu bômb alebo chemikálií pomocou špeciálnych senzorov. Väčšina sond vyslaných k iným planétam slnečnej sústavy bola ovládaná na diaľku, hoci niektoré boli už čiastočne autonómne. V súčasnosti dokonca pracujú vedci z *Carnegie Mellon University* na projekte, v ktorom sa pokúsia pomocou diaľkovo ovládaného vozidla a zariadenia *Oculus Rift* pristáť na Mesiaci.

#### 2.1.2 Vojenská technika

Vojenské využitie sa datuje už do prvej polovice 20. storočia. V oboch svetových vojnách sa používali diaľkovo ovládané lode, tanky a lietadlá z jednoduchého dôvodu, aby znížili straty na životoch. V súčasnosti sa používajú z rovnakých dôvodov. Nebezpečenstvo je minimalizované na operátora vozidla, ktorý ho môže ovládať relatívne z bezpečného miesta. Mnoho policajných oddelení a vojakov využíva vozidlá na zneškodnenie alebo odpálenie náloží alebo mín.

#### 2.1.3 Zábavný priemysel

Pretekárske videohry patria medzi najobľúbenejšie na svete. Oveľa lepšie je však mať model auta a jazdiť v reálnom prostredí. Pomocou zariadenia virtuálnej reality sa virtuálny a skutočný svet dokážu spojiť. Výsledkom je lepších zážitok z ovládania a celkovo z jazdy.

#### 2.1.4 Záchranárske operácie

V prípade zemetrasení sa vozidlo dokáže dostať do štrbín v troskách, čo sa dá využiť na hľadanie ľudí. Alebo v prípade ekologických katastrof sa vozidlo môže využívať na prehľadávanie a mapovanie kontaminovaných území. So zariadením virtuálnej reality bude u operátora vozidla výrazne zlepšená orientácia v priestore, zjednodušené ovládanie kamier a umožnený 3D pohľad priamo z daného miesta.

#### 2.1.5 Vzdelávanie

Výsledky projektu sa dajú priamo využiť na zlepšovanie vzdelávacieho procesu na fakultách informatického zamerania. Pre FIIT STU sú to konkrétne nasledovné existujúce predmety:

- Interakcia človeka s počítačom
- Počítačová grafika
- Pri riešení bakalárskych, diplomových a dizertačných prác

## 3 Analýza

---

### 3.1 Kompresia a dekompresia videa

Video kompresia sa stará o redukovanie a odstránenie nadbytočných dát, čoho výsledkom je digitálne video, ktoré môže byť efektívne poslané cez sieť alebo uložené na disku. Efektívna kompresia dát môže, ale aj nemusí vplývať na výsledný vizuálny efekt. Existujú rôzne techniky pre kompresiu, v tomto dokumente spomenieme 3 najznámejšie: *Motion JPEG*, *MPEG-4 Part 2 (MPEG-4)* a *H.264*.

Proces kompresie zahŕňa aplikovanie algoritmu, ktorý zo zdrojového videa vytvorí komprimovaný súbor, ktorý je následne pripravený na prenos alebo na uloženie na disk. V prípade prehrávania komprimovaného videa je potrebný algoritmus na dekompresiu, vďaka ktorému bude obsah rovnaký ako pôvodne zdrojové video. Tieto dva algoritmy ktoré navzájom spolupracujú sa nazývajú **video kodek** (*codec = encoder/decoder*). Video kodeky rôznych štandardov nie sú obvykle vzájomne kompatibilné. Jednotlivé štandardy pre kompresiu používajú rôzne metódy na redukciu dát a tým pádom sa líšia v rýchlosti, kvalite alebo latencii [1]. Pri vyberaní správneho kodeku sa obvykle stanovujú štyri ciele [2]:

1. Veľkosť komprimovaného súboru.
2. Rýchlosť kompresie.
3. Rýchlosť dekompresie.
4. Kvalita finálneho snímku.

Takéto štandardy je možné rozdeliť do dvoch typov: štandardy pre kompresiu obrazu (*M-JPEG*) a štandardy pre kompresiu videa (*MPEG-4* a *H.264*).

#### 3.1.1 *Motion JPEG*

Kompresia obrazu používa tzv. vnútro-rámcovú (*intraframe*) technológiu kódovania. Dáta sú zredukované v rámci obrazového rámcu odstránením nepotrebných informácií, ktoré nemusia byť viditeľné pre ľudské oko. Snímky sú skomprimované ako jednotlivé obrázky vo formáte *JPEG*. Na obrázku číslo 1 je znázornený princíp formátu *Motion JPEG*. Tri snímky (na obrázku) sú kódované a odoslané ako samostatné unikátne snímky (*I-frame*) nezávisle na sebe [1].



Obrázok 1: Princíp kódovania *Motion JPEG* [1]

#### 3.1.2 *MPEG-4* a *H.264*

Algoritmy ako *MPEG-4* a *H.264* používajú tzv. medzirámcové (*interframe*) predvídanie na redukciu dát medzi sériami snímok. Princíp spočíva v porovnaní snímky s referenčnou snímkou a iba pixely, ktoré sa zmenili od referenčnej snímky budú kódované. Tým pádom sa znižuje počet pixelov, ktoré sú kódované a následne odoslané. Pri zobrazení takejto zakódovanej sekvencie, sa obrázky javia ako



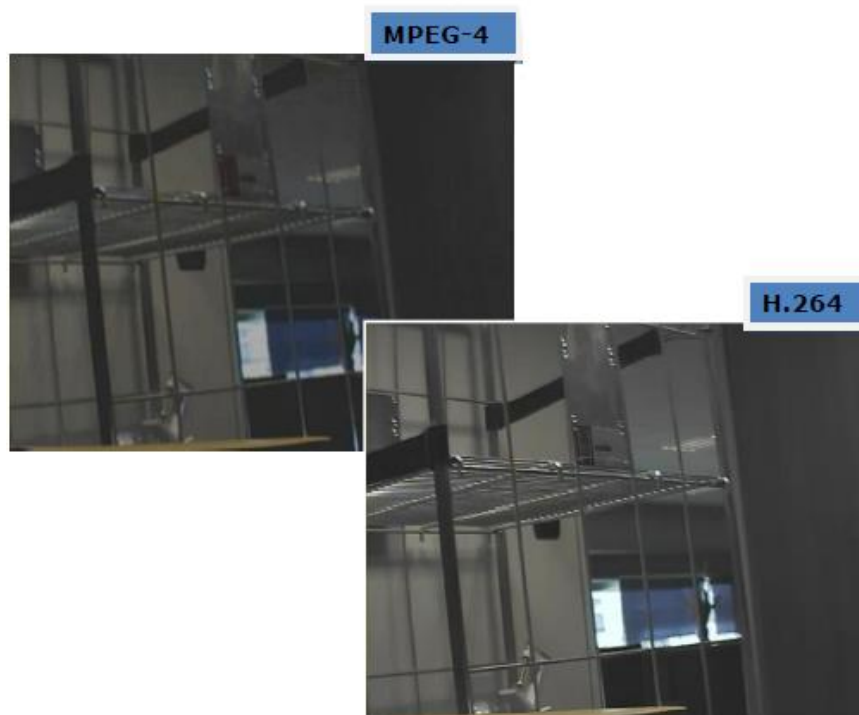
pôvodné video. Na obrázku číslo 2 je zobrazený princíp takéhoto kódovania. Zelenou farbou je zobrazená časť, ktorá sa posiela a žltou prerušovanou čiarou, to čo sa neposiela. Iba prvá snímka (tzv. *I-frame*) je zakódovaná v plnom rozsahu. V nasledujúcich dvoch snímkach (tzv. *P-frame*) sa zakódujú pomocou pohybových vektorov iba pohyblivé časti, čím sa znižuje množstvo odoslaných dát [1].



Obrázok 2: princíp kódovania MPEG-4 a H.264 [1]

### **Rozdiel medzi MPEG-4 a H.264**

*H.264* je najnovší *MPEG* štandard, ktorý má vyspelejšie metódy od základného štandardu *MPEG-4*. Jednou z výhod *H.264* je vysoká rýchlosť kompresie. V porovnaní s *MPEG-4* je až 2x rýchlejší. Ďalšie výhody sú napríklad kvalitnejší obraz, plynulejšie prehrávanie alebo nízka bitová rýchlosť potrebná na sieťový prenos. Na obrázku číslo 3 je znázornené porovnanie týchto dvoch štandardov, kde vidno, že obraz s kompresiou *H.264* je ostrejší a lepšej kvality [3].



Obrázok 3: Porovnanie MPEG-4 a H.264 [3]

### 3.1.3 Multiview Video Coding

**Multiview Video Coding (MVC)** je rozšírením štandardu *H.264*. Toto rozšírenie má vlastnosť efektívneho kódovania sekvenčných dát zachytených súčasne z niekoľkých kamier.

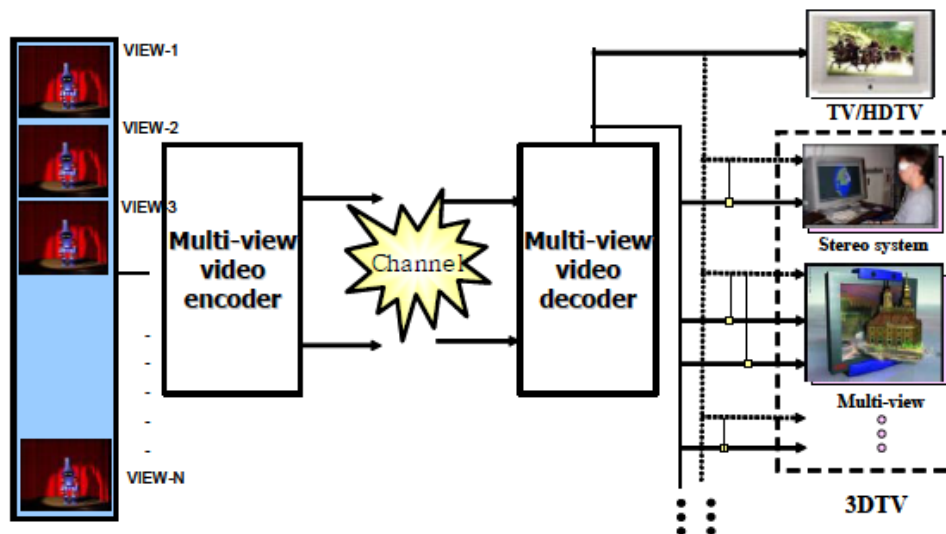
Charakteristika *MVC*:

- Použitie pri kódovaní stereoskopického videa (dve zobrazenia).
- Spätne kompatibilný s *H.264/AVC*, ktorý pri dekódovaní *MVC* videa ignoruje informácie o druhom zobrazení.
- Využitie časovej a medzi pohľadovej (*inter-view*) závislosti.

Na obrázku číslo 4 je zobrazený princíp *MVC*. *MVC* kóder používa časovú predikciu (predvídanie), kde sekvencia po sebe idúcich snímok je kódovaná ako:

- Intra snímok (*I-frame*) – snímok, ktorého všetky bloky sú nezávisle kódované a môžu odkazovať iba na aktuálny snímok.
- Prediktívne snímky (*P-frame*) – snímky, ktorých bloky môžu odkazovať na jeden snímok.
- Bi-prediktívne snímky (*B-frame*) – snímky, ktorých bloky môžu odkazovať na dva snímky.

Množstvo snímok sa prerobí do jedného toku, ktorý sa následne zakóduje a pošle kanálom. Na druhej strane sa takýto tok dát dekóduje a následne je rozdelený do jednotlivých pohľadov. Pôvodné zvonuusporiadanie a multiplexovanie sa vykoná tak, aby sa minimalizovalo využitie pamäte [4; 5].



Obrázok 4: Multiview Video Coding [4]

Pre *MVC* stále nie je podpora otvoreného zdrojového kódu. V dnešnej dobe zatiaľ dekódery *H.264* v knižniciach ako *FFmpeg* alebo *Libav* jednoducho ignoruje dodatočné informácie o druhom pohľade [6].

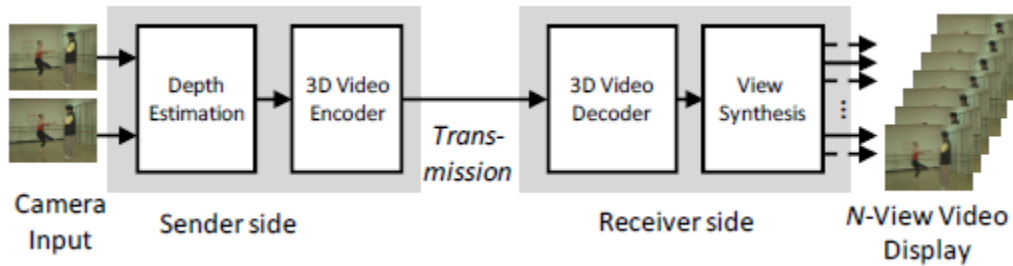
### 3.1.4 3D Video

Podpora pre viac pohľadové zobrazenia (autoskopické zobrazenia) z obmedzeného počtu vstupných pohľadov.

Charakteristika:

- Generovanie syntetických pohľadov použitím videa a hĺbky.

- Minimálne pre jedno video jedna hĺbková mapa.
- Potrebne technológie ako odhad hĺbky, kódovanie hĺbky, pohľadová syntéza.



Obrázok 5: 3D video systém [7]

### 3.1.5 Porovnanie hardvérovej a softvérovej kompresie

Vďaka špeciálnemu hardvéru je **hardvérová kompresia** oveľa rýchlejšia v porovnaní so softvérovou, ktorá používa na túto prácu univerzálny procesor. Napríklad v prípade kompresie za behu je možné komprimovať dáta rýchlosťou, ktorá je rovná, alebo väčšia v porovnaní s rýchlosťou pamäťových médií. Hardvérová kompresia nepridá žiadnu záťaž procesoru, všetky operácie sa robia v rámci vlastného špeciálneho hardvéru. **Softvérová kompresia** môže znížiť výkon počítača pri intenzívnom používaní. Takýto problém môže nastať pri komprimovaní veľkého množstva dát, pri používaní počítača v rovnakom čase [8].

Ďalšie porovnania:

- Hardvérová kompresia je drahšia ako softvérová.
- Softvérová kompresia pridáva prácu pre hostiteľa.
- Softvérová kompresia má výhodu pokiaľ ide o dlhodobé uskladnenie dát.

## 3.2 Prenos videa

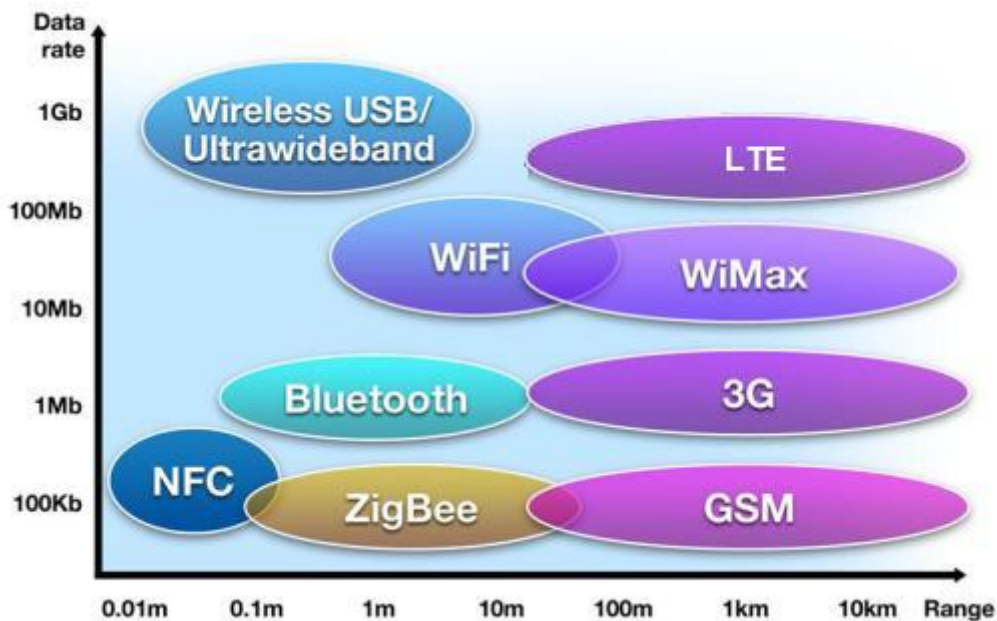
Vďaka neustálemu vývoju získavajú bezdrôtové technológie čoraz väčšie uplatnenie v súčasnom svete a dostávajú sa aj do oblastí, v ktorých ich nasadenie bolo ešte ťažko predstaviteľné. Okrem internetu, multimédií a hlasových služieb sú čoraz viac nasadzované aj v priemysle.

Ich prospech záleží na týchto výhodách:

- úspora nákladov,
- rozšírené možnosti ovládania,
- vyššia spoľahlivosť technológií.

V poslednej dobe sa pomer ceny snímača ku cene vedenia zmenil natoľko, že je možné ušetriť až 70% pôvodných nákladov použitím bezdrôtovej technológie namiesto káblových trás a s nimi súvisiacich elektroinštalčných konštrukcií. [9].

Na obrázku 6 je porovnanie najpoužívanejších bezdrôtových technológií prenosu dát v súčasnosti. Dostupné technológie z hľadiska dátových prenosov pokrývajú široké spektrum aplikácií, od jednoduchého *ZigBee* s rýchlosťou iba pár stoviek bitov za sekundu, až po *Bluetooth*, *WiFi*, alebo *3G* a *LTE*.



Obrázok 6: Prehľad súčasných bezdrôtových technológií [10]

Technológie ako *NFC*, *ZigBee* alebo *GSM*, *EDGE* sú teoreticky vhodné na bezdrôtové riadenie zariadení, ale kvôli ich malým prenosovým rýchlostiam sú nevhodné na prenos väčších dát ako napríklad video stream. Pri jednotlivých technológiách treba dbať aj na oneskorenie ktoré môže znemožniť presné riadenie vozidla.

### 3.2.1 Bluetooth

*Bluetooth* je proprietárny otvorený štandard pre bezdrôtovú komunikáciu, prepájajúci dve alebo viac elektronických zariadení, ako napr. mobilný telefón, osobný počítač a bezdrôtové slúchadlá. Každé zariadenie má svoju jedinečnú adresu (*BD\_ADDR* – *BlueTooth Device Address*) pomocou ktorej je identifikované.

*Bluetooth* je definovaný štandardom *IEEE 802.15.1* a je zaradený medzi osobné počítačové siete. Existuje vo viacerých verziách, práve aktuálna je verzia 4.1, ktorou je vybavená väčšina predávaných zariadení napr. televízie, tablety a mobilné telefóny.

Tento štandard pracuje v ISM pásme 2,4 GHz a v ideálnych podmienkach v najväčšej výkonnej úrovni (100 mW) vie komunikovať do vzdialenosti až 100 m s rýchlosťou 24 Mbit/s [11].

### 3.2.2 WiFi

*WiFi* je súbor štandardov pre bezdrôtové LAN siete. Pôvodne bola navrhnutá pre bezdrôtové zariadenia a lokálne siete, ale dnes sa častejšie používa na pripojenie k internetu pomocou prístupových bodov.

*WiFi* je definovaná štandardom *IEEE 802.11* vo viacerých verziách. Aktuálne najpoužívanejší je štandard *802.11n*, ktorý podobne ako *Bluetooth* pracuje na frekvencií 2,4 GHz a poskytuje rýchlosť prenosu dát 300 Mbit/s do vzdialenosti 100 m. Je spätne kompatibilný s verziami *b*, *g*, *n*. Staršia verzia pracuje na frekvencií 5 GHz a má znížený dosah. Na ochranu dát poskytuje rôzne šifrovacie algoritmy ako napr. *WEP* alebo novší *WPA2* [12].

### 3.2.3 HSDPA

Nadstavba siete 3G, inak nazývané aj **HSDPA (High-Speed Downlink Packet Access)** je bezdrôtová technológia, ktorá umožňuje dátové prenosy v mobilných sieťach. Je založená na inováciách 3G, vďaka ktorým dosahuje rýchlejšie reakcie na zmenu kvality signálu a nižšie oneskorenie.

Dokáže pracovať s rýchlosťou až do 14,4 Mbit/s. Problémom je, že na Slovensku je iba približne 60% pokrytie 3G siete a jej rýchlosť je príliš malá na streamovanie videí. Na ovládanie vozidla by sa teoreticky dala používať, ale mierne oneskorenie môže spôsobiť problémy [13].

### 3.2.4 LTE

Za najmodernejšiu technológiu pre mobilné internetové pripojenie je považovaná technológia **LTE (Long Term Evolution)**. Vďaka nízkemu oneskoreniu a vysokým dosahovaným rýchlostiam, až do 300 Mbit/s, umožňuje LTE sieť aj bezproblémový prenos videa v reálnom čase alebo online prenos obsahu vo vysokej HD kvalite, či 3D videoobsahu.

Problém s touto technológiou spočíva v tom, že zatiaľ na Slovensku nie je veľmi rozšírená. Slovak Telecom síce poskytuje 40% pokrytie populácie, ale to v skutočnosti zahŕňa iba veľké mestá a nie veľkú plochu územia, čo by znamenalo značné ohraničenia na konkrétne lokality [14; 15].

	ZigBee	WiFi	Bluetooth	3G	LTE
<b>Rýchlosť prenosu dát</b>	20, 40, 250 kb/s	11-300 Mbit/s	24Mbit/s	14,4 Mbit/s	300 Mbit/s
<b>Dosah</b>	10-100m	50-100m	< 100m	< 200km	< 100km
<b>Topológia</b>	Ad-hoc, peer to peer, star, mesh	Point to hub	Ad-hoc, veľmi malé siete	n/a	n/a
<b>Operačná frekvencia</b>	2.4GHz celosvetovo	2,4 GHz a 5 GHz	2.4GHz	873 MHz, 1,9 GHz	700 MHz – 2,6 GHz
<b>Spotreba energie</b>	Veľmi nízka	Vysoká	Vysoká	Nízka	Nízka

Tabuľka 1: Porovnanie bezdrôtových technológií

## 3.3 Hardvér

### 3.3.1 Raspberry Pi

**Raspberry Pi** (tiež skrátene nazývané *RPi*) je miniatúrny počítač veľkosti kreditnej karty, ktorý vyvinula nezisková organizácia *Raspberry Pi foundation*. Bol vyvinutý na účely vzdelávania, avšak našiel si veľmi veľa fanúšikov aj medzi bežnou populáciou.

*RPi* je postavený na architektúre *ARM*, vďaka čomu má nízku spotrebu elektrickej energie (približne 1-2W). Podporuje niekoľko operačných systémov založených na Linuxe, pričom odporúčaný je *Raspbian*. *Raspbian* je v podstate *Debian* optimalizovaný pre *Raspberry Pi*. Operačný systém spolu s ostatným softvérom sú uložené na (*micro*) *SD* karte.

K *RPi* sa dajú pripojiť periférie pomocou štandardných rozhraní *USB*, *HDMI*, *S-Video*, *Audio jack* (3.5mm) a v prípade modelov *B* a *B+* aj ethernet (*RJ-45*). Okrem týchto rozhraní má vyvedených niekoľko *GPIO* (*general purpose input/output*) pinov, na ktorých je možné priamo ovládať logickú úroveň napätia. To je užitočné napr. na ovládanie motorov [16].

### 3.3.2 *Arduino*

**Arduino** je vývojová platforma založená na architektúre *AVR* (procesory *ATMEGA*). Na rozdiel od *Raspberry Pi* má menší výkon a iba *GPIO* piny. Nemá operačný systém, takže program beží v reálnom čase [17].

### 3.3.3 *PWM*

**Modulovanie šírkou pulzu** (*pulse width modulation*) je technika riadenia výkonu (alebo inej veličiny) pomocou signálu s pevnou frekvenciou a variabilnými dĺžkami pulzov a medzier [18].

### 3.3.4 *DC motor*

DC motor je najjednoduchší typ motora. Neobsahuje nijakú dodatočnú elektroniku ani prevodovku. Má dva vstupy, ktoré ho napájajú a zároveň riadia. Zmena smeru sa vykonáva zmenou polarít, výkon závisí od napätia. Rýchlosť je možné ovládať pomocou *PWM* [19; 20].

### 3.3.5 *Servomotor*

Základom **servomotora** je DC motor, avšak obsahuje aj prevodovku a elektroniku. Je navrhnutý tak, aby bolo možné presne riadiť jeho polohu. Z toho dôvodu je vhodný na rôzne robotické ramená a podobné účely. Jeho nevýhodou je, že ním nie je možné urobiť plnú (360°) otáčku.

Výhodou je plynulý a rýchly pohyb. Taktiež nestratí informáciu o svojej polohe pri prerušení dodávky energie. (snímanie polohy sa vykonáva mechanicky pomocou potenciometra) [21].

### 3.3.6 *Krokový motor*

**Krokový motor** je podobný DC motoru, avšak na rozdiel od neho má až tri vinutia, ktoré sú statory (magnety sú rotory). Dá sa ovládať rovnako presne ako servomotor, avšak jeho pohyb je pomalší, a náročnejší na riadenie. Je vhodný na tlačiarne a podobné zariadenia [22; 23].

### 3.3.7 *H-mostík*

DC motory nie je možné pripojiť priamo k riadiacej logike. Z toho dôvodu sa na obojsmerné riadenie používajú **H-mostíky**. Tie sú zložené zo štyroch zosilňujúcich tranzistorov v tvare písmena "H". Zapojenie je navrhnuté tak, aby bolo možné meniť polaritu výstupného napätia.

Veľa H-mostíkov sa dá zaobstarať vo forme integrovaného obvodu spolu s prídavnou logikou. Prídavná logika umožňuje najmä ochrániť H mostík pred nevhodnou kombináciou vstupov, ktorá by spôsobila skrat a mostík zničila. Okrem toho môže poskytovať ďalšie funkcie ako napr. "enable" vstup ktorý môže vypnúť motor bez ohľadu na ostatné vstupy.

Veľa integrovaných obvodov s H-mostíkmi obsahuje dva H-mostíky. Tzn. je možné jedným integrovaným obvodom ovládať až dva motory [20].

### 3.3.8 Flyback diódy

Pri odpojení napätia z cievky nastáva elektromagnetická indukcia, ktorá vytvára napätie na cievke pôsobiace proti zmene. Toto napätie môže byť veľmi vysoké a môže zničiť zosilňovací tranzistor alebo H mostík. Na vyriešenie tohto problému sa používajú “flyback diódy”.

Ide o bežné diódy, ktoré sú zapojené v závernom smere vzhľadom na bežnú funkciu obvodu. Za normálnych okolností teda nimi neprechádza elektrický prúd. V prípade, že sa na cievke motora indukuje napätie opačnej polarítity, diódy sa otvoria a skratujú napätie indukované cievkou [24].

## 3.4 Ovládanie

Ovládanie vozidla je možné rôznymi spôsobmi. Najjednoduchší spôsob je pomocou klávesnice, alebo myši. Avšak, vzhľadom na to, že video z vozidla sa bude zobrazovať pomocou virtuálnej reality, je žiaduce, aby operátor mal väčší komfort a aby mal k dispozícii intuitívnejšie ovládanie ako je klávesnica, alebo myš. Operátor nebude vidieť na ovládanie vozidla a preto je dôležité, aby bolo ovládanie tak jednoduché, aby sa dalo realizovať iba pomocou hmatu.

Existuje viacero možností, ako toto dosiahnuť. Veľmi pohodlnou, lacnou a jednoduchou metódou je použitie ovládača pre herné konzoly. Existuje mnoho druhov ovládačov s rôznou funkcionalitou. Príkladom môžu byť rôzne joysticky, volanty, alebo gamepady. Napríklad na obrázku 7 je zobrazený častou používaný gamepad pre konzolu Xbox 360.



Obrázok 7: Gamepad pre Xbox 360 [25]

Tento gamepad obsahuje 2 páčky a niekoľko tlačidiel. Okrem toho obsahuje aj 2 špeciálne tlačidlá, ktoré rozlišujú úroveň ich stlačenia. Pre základné ovládanie vozidla (zrýchľovanie, spomaľovanie a zatáčanie)

postačuje 1 páčka. Tento gamepad je pripojiteľný k PC pomocou USB konektoru, ale existujú aj bezdrôtové varianty.

Z dôvodu, že výpočtový výkon PC je vyšší, ako výkon počítača na palube vozidla, je vhodné, aby sa čo najväčšia časť výpočtov vykonávala na strane PC. Na strane vozidla by sa malo vykonávať iba nevyhnutné minimum výpočtov. Ďalšou požiadavkou na program spracovávajúci vstup z gamepadu je jeho multiplatformovosť. Je dôležité, aby sa vozidlo dalo ovládať bez ohľadu na to, aký operačný systém beží na PC, ktorý ho riadi.

### 3.4.1 Vstup z gamepadu

Signály prichádzajúce z gamepadu je nutné zachytiť a rozpoznať. Existuje viacero alternatív ako to dosiahnuť. Najdiskutovanejšie z nich sú knižnice *XNA* a *SDL*.

Knižnica **XNA 4.0** pracuje v prostredí *.NET* a je určená pre programovací jazyk *C#*. Používa sa pre vývoj hier pre prostredie *MS Windows*. Pre *Linux* však tiež existuje podobná knižnica s názvom *MonoGame*. Táto knižnica obsahuje aj množstvo grafických funkcií. Poskytuje jednoduché rozhranie, ktoré umožňuje spracovávať vstup z gamepadu.

Druhou možnosťou je knižnica **SDL** (*Simple DirectMedia Layer*) verzie 2.0.3. Táto knižnica umožňuje spracovávanie vstupu z rôznych vstupných zariadení. Používa sa hlavne pri vývoji hier a iných multimediálnych aplikácií. Obsahuje aj grafické, alebo audio funkcie. Je napísaná v jazyku *C* a funguje na všetkých najpoužívanejších platformách. Táto knižnica poskytuje wrapper pre funkcie špecifické pre daný operačný systém.

*XNA* a *SDL* majú mnohé vlastnosti podobné, ale v mnohom sa aj líšia. Sú určené pre podobné účely. *C#* je jazyk vyššej úrovne a preto práca s *XNA* môže byť o dosť jednoduchšia a rýchlejšia, ako s knižnicou *SDL*, ktorá je napísaná v jazyku *C*. Na druhej strane *SDL* pracuje na nižšej úrovni a preto z pohľadu rýchlosti je výkonnejšia. *SDL* taktiež pracuje s priamo funkciami operačného systému, pre ktoré poskytuje wrapper a bola písaná s cieľom byť multiplatformová. Preto je *SDL* vhodnejšia pre použitie v multiplatformových aplikáciách, ako *XNA*, kde je prenesenie na rôzne platformy problematickejšie. Pri *SDL*, na rozdiel od *XNA* nie je potrebné pre používateľa danej aplikácie stiahnuť nič navyše, pretože aplikácie sú už skompilované pre architektúru, ktorú užívateľ používa, zatiaľ čo pri *XNA* je nutné si stiahnuť *.NET Framework*. [26; 27; 28; 29].

## 3.5 Zariadenie virtuálnej reality

**Virtuálna realita** označovaná ako VR je počítačovo simulované prostredie, ktoré simuluje fyzickú prítomnosť vo virtuálnom svete.

### 3.5.1 Oculus Rift

*Oculus Rift* je zariadenie virtuálnej reality, ktoré je vyvíjané spoločnosťou *Oculus VR*, vtedy ešte ako nezávislá spoločnosť. Celý projekt začal vďaka podpore cez crowdfundingovú platformu *Kickstarter*. Zákaznícka verzia tohto zariadenia má predpokladaný dátum vydania niekedy v roku 2015. Do dnešného dňa firma vydala dve vývojárske verzie (development kits) *DK1* v roku 2012 a *DK2* v roku 2014. V marci roku 2014 bola spoločnosť *Oculus VR* kúpená Facebook-om.

Súčasná verzia *DK2* obsahuje pentile AMOLED displej, rozlíšenie 960 x 1080 pixlov pre každé oko. Použitý displej je rovnaký ako v smartfónoch *Samsung Galaxy Note 3*. Jeho obnovovacia frekvencia je 75Hz a nominálny horizontálny zorný uhol je 100°. Pre snímanie relatívnej a absolútnej polohy sa



používajú gyroskop, akcelerometer, magnetometer a externá kamera snímajúca infračervené body na „okuliaroch“. Tieto umožňujú snímať *6DoF (6 Degrees of Freedom)* - 6 smerov voľnosti.

*Oculus Rift* oficiálne podporuje platformy *Windows*, *OS X* a *Linux*. Taktiež sa pracuje na integrácii s hernými enginmi *Unity* a *Unreal Engine*. Cena vývojárskej verzie *Oculus DK2* je 350 amerických dolárov [30; 31; 32].

### 3.5.2 Alternatívy k zaradeniu *Oculus Rift*

#### **Sony Morpheus**

Riešenie od Sony je navrhnuté pre hernú konzolu *PlayStation 4*. Obsahuje displej s rozlíšením 1080p a používa novú 3D audio technológiu pre vytvorenie priestorového zvuku, ktorý zodpovedá polohe hlavy. Horizontálny zorný uhol zobrazenia je 90°. Na trh má byť uvedený po roku 2014. Umožňuje sledovanie 6DoF [33; 34; 35].

#### **Vrvana Totem**

*Vrvana Totem* (alebo tiež *True Player Gear Totem*) je zariadené virtuálne reality, ktoré sa zameriava na širokú podporu rôznych platforiem. Medzi podporované patria PC (*Windows*, *Mac*, *Linux*) ale aj herné konzoly *Playstation (3, 4)*, *Xbox (One, 360)*. Taktiež sú podporované herné enginy *Unreal Engine 4*, *Unity*, *Havoc* a *CryEngine*.

Zariadenie obsahuje OLED displej s rozlíšením 1080p s obnovovacou frekvenciou 75 Hz. Zorný uhol zobrazenia je 90°. Zabudované sú 2 kamery (1080p, 120 Hz), gyroskop, akcelerometer a magnetometer. Zvukový výstup poskytuje binaurálny priestorový zvuk. Hardvérovo sa urýchľuje sledovanie pohybu. Výhodou je hardvérovo urýchľované skreslenie obrazu potrebné pre šošovky. Vstavaná je aj podpora pre emuláciu vstupných zariadení (napr. myš alebo gamepad) [36; 37].

V septembri 2014 bola na crowdfundingovej platforme *Kickstarter* spustená zbierka pre tento projekt. Zbierka bola mesiac po jej spustení zrušená a spustil sa predpredaj na stránke spoločnosti *Vrvana*. Odhadovaný dátum rozosielenia prvých 1000 zariadení je október 2015. Odhadovaná cena je 450 amerických dolárov [36; 38].

#### **Gear VR**

Zariadenie **Gear VR** vzniklo spoluprácou spoločností *Samsung* a *Oculus VR*. Zariadenie neobsahuje zabudovaný displej. Smartfón *Samsung Galaxy Note 4* upevnený vo vnútri *Gear VR* je použitý na zobrazovanie a taktiež ako zdroj obrazu. Okrem pripojenia na smartfón prostredníctvom mini USB konektoru riešenie nevyžaduje žiadne iné externé zariadenia, a preto sa jedná o nezávislý systém, ktorý je ľahko prenosný.

Zariadenie má zabudovaný gyroskop, akcelerometer a magnetometer. Zorný uhol je 96°. Rozlíšenie obrazovky vkladaneho smartfónu je 2560x1440 pixlov. Zvukový výstup smartfónu poskytuje priestorový 3D zvuk prostredníctvom slúchadiel [39; 40].

Cena zariadenia *Gear VR* by mala byť 199 amerických dolárov. Jediným podporovaným smartfónom použiteľným s týmto zariadením je však *Samsung Galaxy Note 4*, ktorého cena sa pohybuje na úrovni 750 eur [41].

#### **Avegant Glyph**

Ďalší projekt, ktorý podobne ako viaceré iných v tejto oblasti začal vďaka crowdfundingovej platforme *Kickstarter* je **Avegant Glyph**. Od ostatných riešení sa odlišuje v tom, že miesto jedného veľkého

displeja používa technológiu *Virtual Retina Display*, pomocou ktorej premieta obraz priamo na sieťnicu oka. Výsledkom je ostrý a čistý obraz. Väčšina podobných zariadení je navrhnutá pre úplné ponorenie do virtuálnej reality doma. *Glyph* má byť naproti tomu určený pre použitie v reálnom svete. V prípade, že sa nepoužíva, vyzerá ako štýlové slúchadlá a pri používaní neprekryje celé zorné pole používateľa.

Pre univerzálne pripojenie je použitý HDMI kábel. Displej pozostáva z matice mikrozkadiel, ktoré odrážajú cez optiku obraz na sieťnicu oka. Zobrazovacie rozlíšenie je 1280x720 pixelov na jedno oko a obnovovacia frekvencia je 120 Hz. Diagonálny zorný uhol je 45°. Pomocou senzorov sa sleduje poloha v 6DoF.

V súčasnosti prebieha vývoj. Cena pri predobjednaní je 499 amerických dolárov a dodanie prvého finálneho produktu sa odhaduje na začiatok roku 2015 [42; 43].

### **Google Cardboard**

*Google Cardboard* je minimalistické riešenie, ktorého cieľom je umožniť vytvorenie zariadenia virtuálnej reality z akéhokoľvek smartfónu a čo najlacnejšie. Riešenie pozostáva z kartónovej konštrukcie zomknutej na koncoch magnetom a upevnenými šošovkami, ktoré zakrivujú obraz z displeja smartfónu. Ten je upevnený v kartóne gumičkou a suchým zipsom. Voliteľne je možné pridať *NFC* tag.

Spoločnosť *Google* vytvorila experimentálny balíček SDK pre jednoduchý vývoj nástrojov virtuálnej reality. Pre použitie tohto „zariadenia“ je postačuje smartfón s prehliadačom podporujúcim technológiu *WebGL* (napr. *Chrome*). Toto riešenie je dostupné za 2,55 amerických dolárov s poštovným zdarma [44; 45; 46].

## **3.6 Oculus SDK**

Pre zlepšenie integrácie zariadenia *Oculus Rift* do hier a aplikácií poskytuje spoločnosť *Oculus VR* balíček nástrojov **SDK** (*Software Development Kit*). Oficiálne sú podporované platformy *Microsoft Windows*, *OS X* a čoskoro pribudne aj podpora pre *Linux*. Hlavnou časťou SDK sú zdrojové kódy a binárne súbory knižnice *LibOVR*. Taktiež obsahuje dokumentáciu, príklady a nástroje pre vývojárov. API pre použitie SDK je vytvorené v jazykoch *C/C++* [47].

### **3.6.1 Odporúčané kroky pri integrácii LibOVR**

*Oculus SDK* bolo navrhnuté s cieľom umožniť čo najľahšiu integráciu. Pri integrácii podpory zariadení *Oculus* do nových aplikácií je odporúčané dodržiavať nasledujúce kroky [47]:

1. Inicializácia *LibOVR*.
2. Enumerácie zariadení *Oculus*, vytvorenie objektu **ovrHmd** a spustenie senzorových vstupov.
3. Integrácia sledovania polohy hlavy do zobrazovacej a pohybovej časti aplikácie.
4. Inicializácia renderovania pre **HMD okuliare** (*Head-mounted display*).
5. Modifikácia renderovania snímok pre integráciu podpory HDM a správneho časovania snímok.
6. Prispôbenie používateľského rozhrania pre správne fungovanie v zariadení virtuálnej reality.

### **3.6.2 Senzory a sledovanie polohy hlavy**

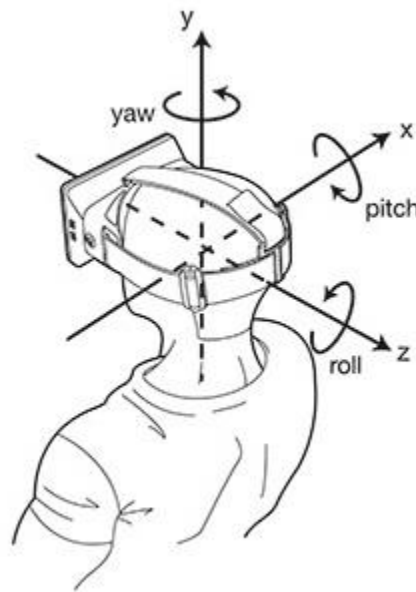
Zariadenie *Oculus Rift* obsahuje viacero nízkoúrovňových senzorov, z ktorých sú prijímané tieto surové dáta [47]:

- Akcelerometer
- Gyroskop
- Magnetometer
- Teplomer

- Čas odčítania hodnôt
- Vo verzii DK2 bola pridaná externá kamera, ktorá sníma infračervené vysielacie na zariadení pre zlepšenie sledovania polohy.

Prostredníctvom API sú dispozícii aj tieto surové dáta zo senzorov. Bežne sa však používajú údaje, ktoré určí API na základe surových dát zo senzorov použitím zlučovania, korekcie a predikcie. Výsledkom je pozícia používateľa zariadenia Oculus Rift.

Hodnota pozície je určená pre špecifikovaný bod v čase, typicky zodpovedajúci času v budúcnosti, kedy sa snímka renderovaním zobrazí na obrazovke. Toto umožňuje zlepšiť zážitok pri používaní zariadenia Oculus Rift. Pozícia získaná pomocou API obsahuje **šest' stupňov** voľnosti (**6DoF - six Degrees of Freedom**). 6DoF pozostáva z 3D polohového vektora a orientačného quaterniona. Orientácia je hlásená ako rotácia v pravorukom súradnicovom systéme (viď. obrázok 8) [47].



Obrázok 8: Súradnicový systém zariadenia Oculus Rift [47]

Ako je možné vidieť na obrázku, súradnicový systém používa nasledovné definície jednotlivých osí [47]:

- Y je kladná v smere nahor
- X je kladná v smere doprava
- Z je kladná v smere dozadu

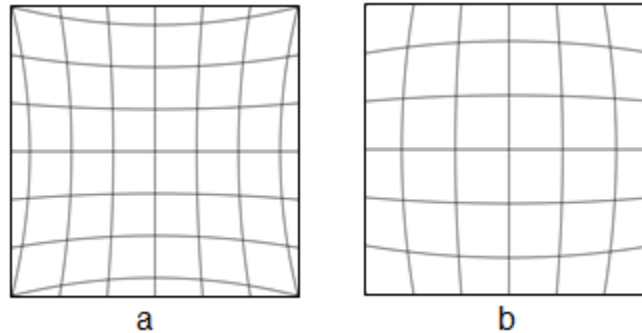
Rotácia je uchovávaná ako jednotkový quaternion, ale taktiež môže byť prevedená do formy vybočenie-sklon-náklon (yaw-pitch-roll). Pozitívna rotácia je v smere hodinových ručičiek pri pohľade zo záporného smeru každej osi. Komponenty rotácie sú [47]:

- **Sklon (pitch)** – rotácia okolo osi X, kladná pri naklonení dohora.
- **Vybočenie (yaw)** – rotácia okolo osi Y, kladná pri otočení doľava.
- **Náklon (roll)** – rotácia okolo osi Z, kladná pri naklonení doľava v rovine XY.

### Renderovanie na Oculus Rift

Oculus Rift vyžaduje rozdelenie obrazu na 2 polovice a použitie každej z nich pre jedno oko. Aby poskytl väčší zorný uhol a zlepšili ponorenie do virtuálnej reality. Šošovky v zariadení približujú obraz. Tento proces však podstatne skresľuje obraz. Preto je na obraz pre každé oko potrebné pred zobrazením aplikovať **skreslenie**, ktoré tento nedostatok šošoviek kompenzuje [47].

Šošovky do obrazu vnášajú **poduškové skreslenie** (*pincushion distortion*). Pri tomto skreslení zväčšenie obrazu rastie so vzdialenosťou od optickej osi (obrázok 9a). Toto skreslenie sa pri renderovaní kompenzuje **súdkovým skreslením** (*barrel distortion*). Pri súdkovom skreslení zväčšenie klesá so vzdialenosťou od optickej osi (obrázok 9b) [48].



Obrázok 9: Porovnanie poduškového (a) a súdkového (b) skreslenia [47]

Štandardné *Oculus C API* poskytuje dve možnosti vytvorenia tohto skreslenia: SDK renderovanie skreslenia a klientske renderovanie skreslenia. V oboch prípadoch môže aplikácia renderovať pohľad na oddelené textúry alebo na jednu spoločnú. API podporuje obe najčastejšie podporované grafické aplikačné rozhrania: *Direct3D* aj *OpenGL*.

### 3.6.3 Renderovanie skreslenia

Skreslenie je možné renderovať dvoma spôsobmi: SDK renderovaním skreslenia a klientskym renderovaním skreslenia. SDK renderovanie sa má stať preferovaným prístupom v budúcich verziách SDK.

- Pri **SDK renderovaní skreslenia** sa knižnica stará o časovanie, renderovanie skreslenia a výmenu buffer-ov. Aby bolo toto umožnené, vývojár poskytne rozhraniu API nízkoúrovňové ukazovatele na zariadenie a textúry a pridá do cyklu na vykreslenie snímky (**frame**) volanie funkcií `ovrHmd_BeginFrame` a `ovrHmd_EndFrame`. Pri tomto riešení nie je nutné mať znalosti o shaderoch [47].
- Pri **klientskom renderovaní skreslenia** musí byť skreslenie renderované aplikačným kódom. Skreslenie je zakódované v mesh vertexe (v starších verziách to bolo použitím explicitnej funkcie v pixel shaderi). SDK taktiež poskytuje explicitné funkcie pre časovanie a predikciu [47].

### 3.6.4 OpenGL vs. Direct3D

Renderovanie obrazu na *Oculus Rift* sa dá zabezpečiť použitím grafického API. Na výber sú dve možnosti. *OpenGL* alebo *Direct3D*. Hlavný rozdiel je v kompatibilitate. Zatiaľ čo *Direct3D* je podporovaný len na *Microsoft Windows*, *OpenGL* je otvorený štandard a podporuje rôzne platformy. Obe sú implementované v grafickom ovládači. Rozdiel je ale v tom, že *Direct3D* implementuje API v runtime (dodávaný *Microsoftom*), ktorý následne komunikuje s nízkoúrovňovým rozhraním grafického ovládača.

Pri *OpenGL* každý výrobca implementuje celé API do ovládača. Toto znamená, že niektoré funkcie sa môžu správať inak na grafických kartách od iných výrobcov.

Proprietárne *Direct3D* je oficiálne podporované len na *Microsoft Windows*, vrátane verzií použitých v *Xbox* konzole a *Sega Dreamcast*. Boli vytvorené rôzne reimplementácie tretími stranami ako napríklad *Wine*, tento proces je ale náročný pre nutnosť reverse engineeringu.

*OpenGL* je implementovaný na rôznych platformách ako *Microsoft Windows*, *Mac OS X*, *GNU/Linux*. *Nintendo* a *Sony* vyvinuli vlastné knižnice podobné *OpenGL*. Podskupina *OpenGL* je použitá aj v *Androide*, *BlackBerry*, *iOS* a *Symbian* vo forme *OpenGL ES (OpenGL for Embedded Systems)*.

Okrem drobných funkcionálnych rozdielov, ktoré boli za uplynulú dobu vyriešené, poskytujú tieto API skoro rovnakú úroveň funkcií [49].

### 3.7 OpenGL (Open Graphics Library)

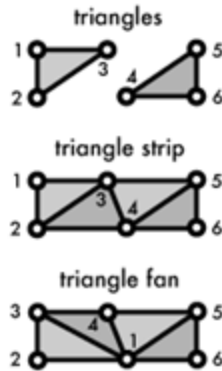
*OpenGL* je priemyselný štandard špecifikujúci viacplatformové rozhranie (API) ku grafickým kartám. Slúži na tvorbu aplikácií, ktoré pracujú najmä s trojrozmernou počítačovou grafikou prekresľovanou v reálnom čase. Používa sa pri tvorbe hier, CAD programov, aplikácií virtuálnej reality, vedecké vizualizácie a iné. *OpenGL* bolo vyvinuté firmou *Silicon Graphic Inc.* a vydané v roku 1992.

Toto API je definované ako súbor funkcií, ktoré sa volajú klientskym programom. Tieto funkcie sú podobné jazyku *C*, sú však jazykovo nezávislé.

Od prvých dní 3D grafiky sú trojuholníky základom grafických scén. Moderné grafické karty vedia tento fakt zamaskovať, avšak stále sú to len trojuholníky. **Grafické potrubie (*pipeline*)** zobrazené na obrázku 12, ktoré *OpenGL* implementuje znázorňuje, že program vyplní *OpenGL* spravované vyrovnávacie pamäte s poľami bodov (*vertices*). Tieto body sú premietnuté na obrazový priestor, spojené do trojuholníkov a rasterizované do fragmentu o veľkosti pixelu. Následne sú týmto fragmentom priradené farebné hodnoty a sú vykreslené do *framebuffer-u*. Moderné grafické karty získavajú svoju výhodu v tom, že nahradia kroky premietania na obrazový priestor a priradenie farebných hodnôt programami, ktoré sú nazývané *shadery* [50].

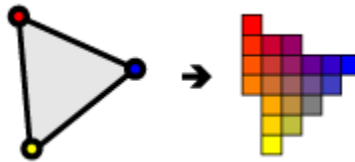
**Renderovanie** sa začne v jednej alebo viacerých **vyrovnávacích pamätiach s bodmi (*vertex buffers*)**, ktoré sú vyplnené atribútmi daných bodov. Tieto sú použité ako vstupy do *shader-a* bodov. Najčastejšie atribúty obsahujú pozíciu bodu v 3D priestore a súradnice textúry. Súbor takýchto vyrovnávacích pamätí sa nazýva **poľo bodov (*vertex array*)**.

GPU (grafická karta) začne čítaním každého bodu z poľa bodov, ktorý potom prejde cez **shader bodov (*vertex shader*)**. Shader bodov minimálne vypočíta pozíciu bodu na zobrazovacej ploche. GPU následne premietne body tak, že vytvoria trojuholník. Body môžu byť zobrazené do viacerých elementov. Buď sa zoberú každé tri a vytvoria samostatný trojuholník alebo sa vytvorí trojuholníkový pásik tak, že posledné dva body z každého trojuholníka sú prvé dva ďalšieho obrázok 10.



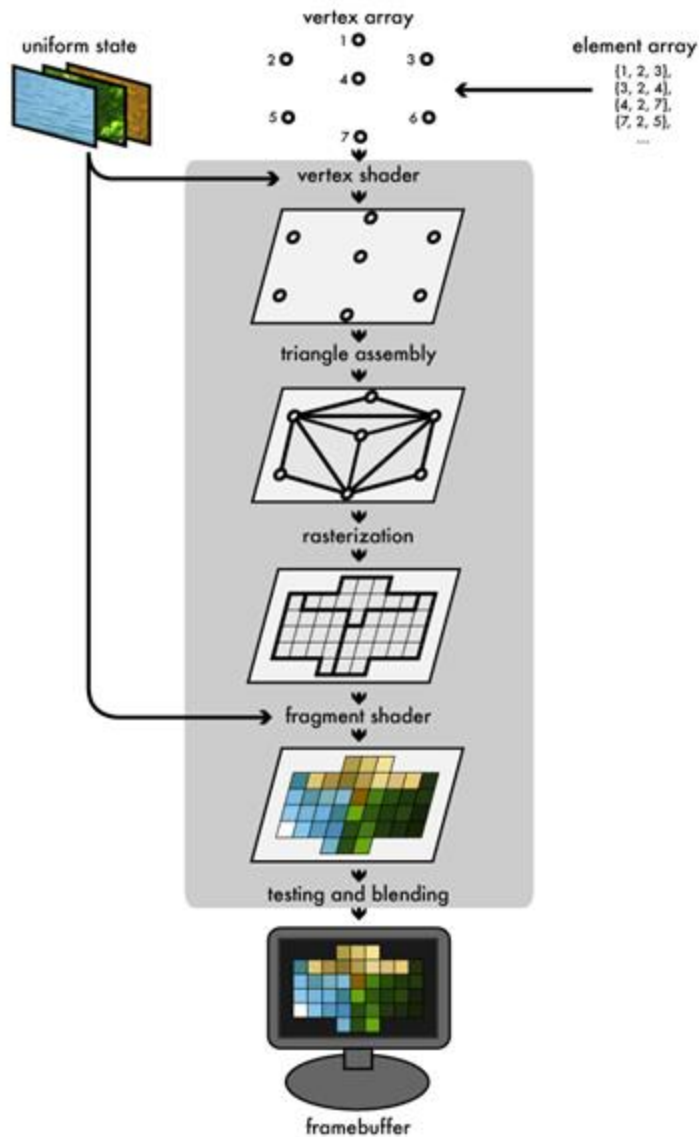
Obrázok 10: Spôsob zobrazenia bodov [50]

**Rasterizácia** zoberie každý trojuholník, oreže časti, ktoré sú mimo obrazovky a rozdelí zvyšné viditeľné časti na fragmenty o veľkosti pixelu. Pri použití farieb zo shadera bodov sa fragmenty trojuholníka zafarbia tak, aby bol prechod medzi farbami postupný obrázok 11.



Obrázok 11: Rasterizácia [50]

Tieto fragmenty potom prejdú cez ďalší program nazývaný **fragment shader**. Výstupom je farba a hĺbka, ktoré sú potom vykreslené do **framebuffera**. Najčastejšie použitie je na mapovanie textúr a nasvietenie. Keďže sa toto vykonáva pre každý pixel zvlášť, dokážeme vytvoriť zaujímavé efekty, tak isto je to ale najnáročnejšia časť tohto potrubia [50].



Obrázok 12: OpenGL potrubie (pipeline) [50]

## 3.8 Softvérové spracovanie videa

### 3.8.1 OpenCV (Open Source Computer Vision)

**OpenCV** je knižnica s otvoreným kódom napísaná v jazyku C/C++ orientovaná na počítačové videnie a strojové učenie. Je navrhnutá s ohľadom na výpočtovú efektívnosť so silným zameraním na aplikácie v reálnom čase. Bola vyvinutá vývojovým centrom *Intel Russia* a dnes je podporovaná výskumným laboratóriom a technologickým inkubátorom *Willow Garage*. Táto knižnica je multiplatformová (podporuje *Windows*, *Linux*, *Mac OS*, *iOS*, *Android*) a sústreďuje sa hlavne na počítačové videnie v reálnom čase.

V dnešnej dobe má táto knižnica množstvo algoritmov na spracovania obrázkov a videa, ktoré zefektívňujú vývoj. Niektoré z možností knižnice *OpenCV* sú rozpoznanie tváre, gest, sledovanie objektov, rozšírená realita. Pre podporu týchto funkcií taktiež obsahuje viacero knižníc pre štatistické

strojové učenie. Knižnica využíva výhody inštrukcií *MMX* a *SSE*. Taktiež sa aktívne vyvíja rozhranie pre technológie *CUDA* a *OpenCL*.

S knižnicou sa dodávajú aj knižnice tretích strán, ktoré sú najčastejšie zamerané na zlepšenie podpory rôznych formátov, kontajnerov a kodekov alebo na zvýšenie efektivity knižnice. Príkladom takejto knižnice je **FFmpeg**, ktorá je zameraná na spracovanie multimediálnych dát [51; 52].

### **3.8.2 AForge.Net**

*AForge.Net* je knižnica s otvoreným zdrojovým kódom v jazyku *C#* navrhnutá pre výskumníkov v oblasti počítačového videnia a umelej inteligencie. Pôvodne bola vyvinutá pre *.NET framework*. Obsahuje funkcie pre spracovanie obrázkov, neurónové siete, všeobecné algoritmy fuzzy logiku, strojové učenie a robotiku. Knižnica obsahuje aj množstvo ukážok, ktoré demonštrujú jej funkcionality [53].

### **3.8.3 IVT (Integrating Vision Toolkit)**

*IVT* je nezávislá multiplatformová knižnica s otvoreným zdrojovým kódom napísaná v jazyku *C++* zameraná na počítačové videnie s objektovo-orientovanou architektúrou. Ponúka jednoduché rozhranie pre prácu s kamerami a všeobecný model kamery ako aj implementáciu funkcií a štruktúr na spracovanie obrazu a matematických údajov. Taktiež umožňuje multiplatformový prístup pre vytváranie používateľského rozhrania [54].

## **3.9 Existujúce riešenia**

Pri hľadaní podobných riešení sa nepodarilo nájsť rovnaké existujúce riešenie vypracované na Slovensku. Avšak využitím virtuálnej reality sa už niektoré univerzity začali zaoberať, napríklad Filozofická fakulta Univerzity Komenského v Bratislave. Konkrétne sa zamerali na sociálnu prezenciu, ktorá vzniká pri vnímaní iných osôb v spoločnom virtuálnom priestore. Pomocou merania osobnej zóny, ktorú participanti zanechali pri interakcii s počítačom vytvorenými reprezentáciami ľudí, v plne imerznom virtuálnom prostredí, navodenom prototypom HMD okuliarov pre virtuálnu realitu, sa snažili preukázať, že vďaka podvedomým mechanizmom sa správali participanti k umelým osobám, ako k skutočným ľuďom [55].

Vo svete však podobné riešenia už existujú. Pre zaujímavosť jedno z nich používa úplne iné technológie, ako plánujeme využiť v našom projekte. Tvorca použil iba jednu kameru a tak dokázal vytvoriť iba 2D video. Na prenos videa nepoužil *WiFi* sieť ale vysielal video na frekvencii 5,8GHz. Na ovládanie vozidla použil vysieláč *Futaba T8UAP*. Hlavnou podobnosťou s našim riešením je ovládanie pohybu kamery pomocou polohy hlavy. Oveľa viac existujúcich riešení existuje pre zariadenia virtuálnej reality v spojení s bezpilotnými lietadlami [56].



## 4 Specifikácia požiadaviek

---

V tejto kapitole sú uvedené požiadavky pre realizáciu navrhovaného riešenia (funkcionálne, nefunkcionálne, hardvérové, softvérové). Princípom projektu je použitie zariadenia virtuálnej reality operátorom vozidla, pričom sa v tomto zariadení bude premietat' obraz z kamier umiestnených na vozidle. Vozidlo sa bude ovládať gamepadom.

### 4.1 Funkcionálne požiadavky

- smerovanie kamier podľa polohy hlavy
- pohyb vozidla pomocou gamepad zariadenia
- zaznamenávanie 3D video obsahu vnoreným systémom
- prenos video obsahu z vozidla do PC
- reprezentácia 3D video obsahu v PC
- bezdrôtová komunikácia medzi vozidlom a PC

### 4.2 Nefunkcionálne požiadavky

- bezpečnosť - zabezpečenie bezdrôtového spojenia
- multiplatformovosť - funkčnosť na viacerých operačných systémoch
- energetická efektívnosť - potrebné zabezpečenie z dôvodu napájania z bateriek
- rýchla odozva - vozidlo bude rýchlo reagovať na vstupy používateľa
- spoľahlivosť - vozidlo bude presne reagovať na signály od operátora

#### 4.2.1 *Hardvérové požiadavky*

V tejto podkapitole sú zhrnuté a opísané hardvérové požiadavky pre tento projekt.

##### *Raspberry Pi B+*

- riadenie vozidla

##### *IP kamera*

- min. rozlíšenie 1080x960
- video kompresia: H.264
- podpora sieťového protokolu UDP
- min. frame rate 25fps

##### *Oculus Rift DK2*

- Zariadenie virtuálnej reality vrátane potrebného príslušenstva

##### *mikroSD karta (16 GB)*

- úložisko dát pre Raspberry Pi

##### *Servo motory*

- otáčanie kamier
- 3 ks

##### *DC motory*

- pohyb vozidla

- 2 ks

#### ***Doska plošného spoja***

- zapojenie podľa schémy v kapitole 5.4.3.

#### ***Xbox 360 ovládač***

- potrebný na ovládanie vozidla

#### ***Centrálna stanica***

- počítač s minimálnymi požiadavkami:
  - Intel Core i7
  - RAM 8GB,
  - NVIDIA GeForce GTX 760 3GB

### **4.2.2 Softvérové požiadavky**

V tejto časti dokumentu sú zhrnuté technológie a programovacie jazyky potrebné pre tento projekt.

#### ***Operačný systém***

- Windows 7 alebo novší

#### ***Vývojové prostredie***

- primárny programovací jazyk C/C++

#### ***Oculus SDK***

- knižnica integráciu zariadenia *Oculus Rift*

#### ***OpenGL***

- renderovanie obrazu

#### ***OpenCV***

- spracovanie videa
- dekodovanie videa (súčasť *FFmpeg*)

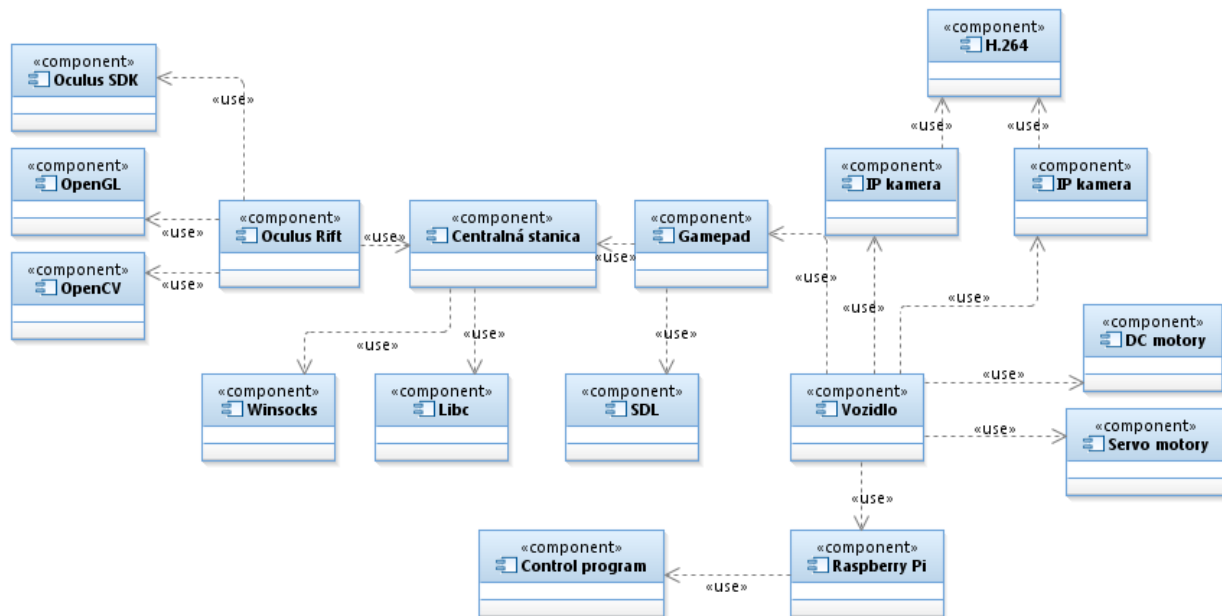
#### ***SDL***

- spracovanie vstupov z ovládacieho zariadenia

## 5 Návrh

Táto kapitola obsahuje návrh vozidla ovládaného pomocou zariadenia virtuálnej reality, vrátane diagramu komponentov jazyka *UML*.

### 5.1 Architektúra systému



Obrázok 13: Diagram komponentov jazyka *UML*

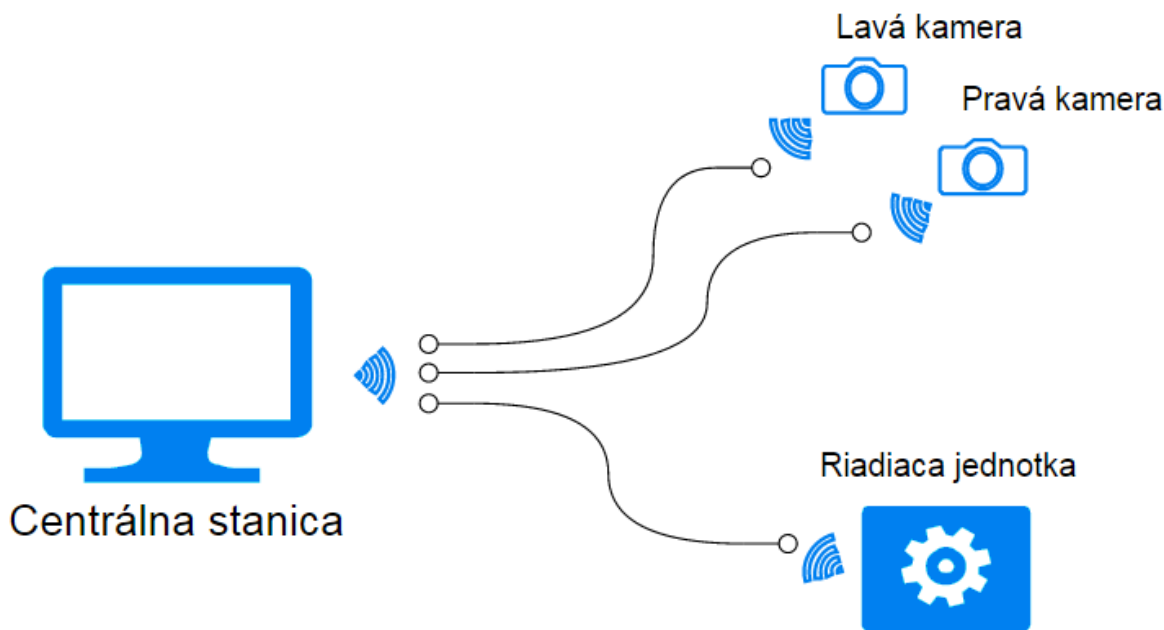
### 5.2 Kompresia dát

V časti analýzy sme porovnávali štandardy pre kompresiu *Motion JPEG*, *MPEG-4* a *H.264* ako aj softvérovú a hardvérovú kompresiu. Rozhodli sme sa použiť komprimačný algoritmus **H.264**, pretože poskytuje vyspelejšie metódy, ako napríklad rýchlejšiu kompresiu. *Multiview Video Coding* nemá podporu otvoreného zdrojového kódu, preto nie je možné toto rozšírenie štandardu *H.264* využiť. Rýchlosť štandardu *H.264* hrá dôležitú úlohu pri vývoji nášho projektu. Na základe tohto rozhodnutia sme si zvolili IP kamery, ktoré podporujú takýto typ kompresie. Zvolením kamery sme dali prednosť hardvérovej kompresie pred softvérovou. Kamera má vlastný špeciálny hardvér na kompresiu, vďaka čomu je hardvérová kompresia oveľa rýchlejšia od tej softvérovej. Pri výbere kamery bolo základnou požiadavkou minimálne rozlíšenie 960x1080 pixlov, pretože zariadenie *Oculus Rift* požaduje na jedno okolo presne také rozlíšenie.

### 5.3 Prenos videa

Na prenos videa sme sa rozhodli použiť bezdrôtovú technológiu **WiFi**, štandard *802.11n*. Po podrobnej analýze a porovnaní rôznych bezdrôtových technológií sme zistili, že pre náš projekt má táto technológia najlepšie parametre. Poskytuje najrýchlejší prenos dát spomedzi porovnaných, rýchlosť až do 300Mbit/s v ideálnych podmienkach. Má dostatočne dlhý dosah, 100 m pri ad-hoc riadení, čo zabezpečuje pohodlné riadenie aj z väčšej vzdialenosti. Vozidlo sa dá ľahko pripojiť aj na internet, a to pomocou lokálne

dostupných prístupových bodov. Tým pádom je možné riadiť vozidlo cez internet. V tomto prípade ale musíme rátať s väčšou odozvou zariadenia. Ďalšou výhodou je aj možnosť šifrovania, s ktorou môžeme zabezpečiť naše vozidlo proti nežiadaným útokom. Pomocou *WiFi* vieme ľahko prepojiť IP kamery, riadiacu jednotku vozidla a centrálnu riadiacu stanicu. Jedinou nevýhodou je väčšia spotreba energie ako pri ostatných technológiách, čo nám môže skrátiť dobu použitia vozidla pri jednom nabití, alebo zvýšiť váhu celého vozidla použitím väčšej batérie.



Obrázok 14: Návrh bezdrôtového prenosu dát

## 5.4 Hardvér

### 5.4.1 Riadenie vozidla

Na riadenie vozidla sme sa rozhodli použiť **Raspberry Pi**. Vyberali sme medzi ním a populárnou platformou *Arduino*. Obe sú veľmi obľúbené a preto sme sa rozhodovali hlavne medzi nimi.

Keďže sme sa na prenos videa rozhodli použiť *WiFi*, chceli sme rovnakú technológiu použiť aj na ovládanie, aby bol vývoj jednoduchší. Na použitie *WiFi* je *Raspberry Pi* vhodnejšie ako *Arduino*, pretože obsahuje operačný systém, ktorý veľmi uľahčuje vývoj.

### 5.4.2 Pohon vozidla a pohyb kamery

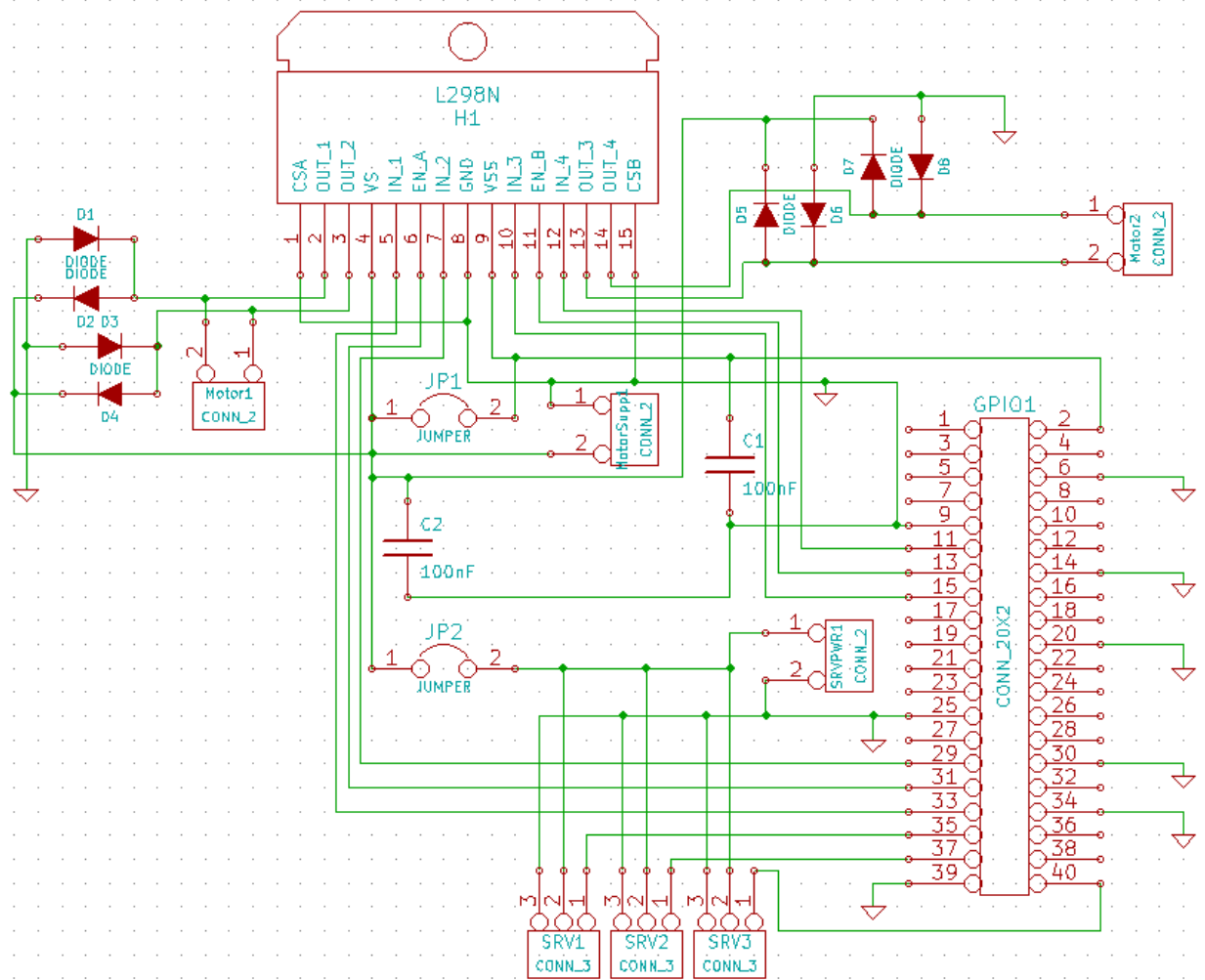
Na pohon vozidla sme sa rozhodli použiť obyčajné 5V DC motory a na pohyb kamier modelárske servomotory. Ďalšou alternatívou, ktorú sme zvažovali boli krokové motory. Tie sme ale vylúčili najmä kvôli tomu, že na pohon potrebné nie sú (obyčajné DC motory sú lacnejšie) a na pohyb kamier sa viac hodia servá kvôli tomu, že dokážu lepšie kopírovať prudké pohyby hlavy.

Servomotory budú pripojené cez ich štandardné rozhranie priamo na GPIO piny *Raspberry Pi* (napájanie možno bude osobitne). DC motory budú ovládané dvojitým H mostíkom L298N, ktorý má dobré vlastnosti, dostupnosť aj cenu.

### 5.4.3 Schéma zapojenia

**H** mostík zvládne prúd 2A (v špičke až 3A) pri max 46V. Pred indukovaným napätím bude chránený ôsmimi flyback diódami *1N4007*. Tie sú veľmi lacné a zároveň veľmi odolné, čo môže byť vhodné, ak sa rozhodneme použiť výkonnejšie motory (v prípade, že 5V nebude stačiť).

Aby sme umožnili neskôr zmeniť zdroj napätia motorov bude  $V_S$  prepojený s  $V_{SS}$  pomocou jumpera. Servomotory sa tiež budú dať napájať z osobitného zdroja. Oddelenie bude realizované opäť pomocou jumpera.



Obrázok 15: Návrh schémy zapojenia

## 5.5 Ovládanie

Z dôvodu ľahkej dostupnosti bude použitý gamepad pre hernú konzolu **Xbox 360**. Na ovládanie pohybu vozidla bude použitá pravá páčka. Druhá páčka a zvyšné tlačidlá môžu byť v budúcnosti použité pri ovládaní možných pokročilejších funkcií vozidla.

Ovládač pomocou USB rozhrania pripojíme k PC. Tam sú signály z ovládača spracované a upravené pre potreby vozidla a pomocou protokolu **UDP** odoslané k prijímaču na strane vozidla. IP adresu vozidla je nutné vopred poznať.

Na základe analýzy sme dospeli k názoru, že použitie *SDL* je vhodnejšie ako *XNA* na splnenie požiadaviek, ktoré sú na program kladené

### 5.5.1 *Spracovanie vstupu z gamepadu*

Pre spracovanie vstupu z gamepadu použijeme knižnicu *SDL* a funkcie, ktoré obsahuje.

Vytvoríme funkciu, ktorá cyklicky zisťuje pozíciu pravej páčky. Táto pozícia je definovaná pomocou hodnôt osí X a Y. V strede majú obe dve hodnoty 0. Y rastie smerom zhora dole a X rastie smerom zľava doprava. Hodnoty osí bývajú v rozsahu od *SHRT\_MIN* do *SHRT\_MAX*. To býva typicky od -32767 do 32767. Tieto hodnoty program prepočítava na percentá, kde páčka keď je natočená úplne napravo má na osi X hodnotu 100 a keď je natočená úplne naľavo má zase -100. Keď je natočená úplne dole, Y má hodnotu 100 a keď je zase úplne hore, je to -100. Ostatné tlačidlá môžu byť použité pre rozširujúce funkcie.

Veľmi malé hodnoty osí X a Y (<13) sa automaticky nahrádzajú hodnotou 0. Týmto sa zabraňuje tomu, aby drobné výchylky páčky od stredu, spôsobené napríklad chvením ruky držiacej páčku, alebo nepresnosťou mechaniky páčky neovplyvňovali činnosť vozidla [57].

### 5.5.2 *Príprava dát pre vozidlo*

Keďže cieľom je čo najviac výpočtov vykonať na riadiacom PC a nechať čo najmenej na počítač na palube vozidla, vstupy z gamepadu sa hneď po spracovaní prepočítajú na dáta, ktoré potrebuje prijať ovládač motorov v vozidlu.

Hodnota osi Y bude invertovaná. Kladná hodnota teda bude znamenať, že vozidlo má zrýchľovať. Páčka naklonená do strany znamená, že auto má zatáčať daným smerom. Ďalšie výpočty zohľadňujú spôsob, akým je vozidlo riadené.

Vozidlo je poháňané 2 motormi. Keď má vozidlo ísť rovno, oba motory ho ťahajú rovnakou rýchlosťou vpred. Keď má vozidlo zatáčať doľava, ľavý motor spomaľuje, prípadne začne cúvať a zároveň pravý motor zrýchľuje, potenciálne až po jeho maximálnu rýchlosť. Zatáčanie doprava je riešené obdobne.

Os Y určuje, akú majú mať oba motory rýchlosť. Hodnota 100 znamená, že motor má ísť na 100% jeho maximálnej rýchlosti v smere dopredu, -100 zase, že má ísť na 100% jeho maximálnej rýchlosti v smere dozadu. Hodnota Y sa použije rovnako pre motory a určí im, akou rýchlosťou majú ísť. Zatáčanie určuje os X. Tá nadobúda záporné hodnoty pri naklonení doľava a kladné pri naklonení doprava. Ich pripočítaním/odpočítaním od hodnoty Y dostaneme výslednú hodnotu, ktorú nastavíme motoru.

Pri zatáčaní doprava sa pravý motor spomalí a to podľa toho, ako veľmi je páčka naklonená doprava. Takže od hodnoty, ktorú pravý motor dostane od osi Y sa odčíta hodnota, ktorú dostane od osi X. Teda výsledná hodnota, ktorá sa pošle pravému motorčeku sa počíta nasledovne:

$\text{Pravý motor} = \text{os Y} - \text{os X}$
--

Pri zatáčaní doľava sa zase spomaľuje ľavý motorček v závislosti od naklonenia páčky doľava. Pri naklonení doľava má os X zápornú hodnotu a preto ju od hodnoty osi Y nebudeme odčítat', ale naopak pripočítavať k nej. Výsledná hodnota ľavého motorčeka teda bude:

Ľavý motor = os Y + os X
--------------------------

Takýmto spôsobom sa pripravia dáta pre počítač bežiaci na palube vozidla a ten potom nie je zaťažovaný týmito výpočtami.

### 5.5.3 Posielanie dát vozidlu

Takto pripravené dáta sa posielajú vozidlu pomocou protokolu **UDP**. Samotné dáta sa snažíme prenášať čo najúspornejším spôsobom. Aplikačný protokol má iba 2B. Prvý bajt obsahuje hodnotu, ktorá je určená pre ľavý motor. Druhý bajt zase obsahuje hodnotu, ktorá je určená pre pravý motor. Oba bajty môžu obsahovať hodnoty od **-100** do **+100**. Protokol je rozšíriteľný o ďalšie bajty, ktoré by určili činnosti ďalších, pokročilejších funkcií vozidla.

Posielanie dát bude realizované pomocou socketov a štandardných siet'ových knižníc operačných systémov *Windows* a *Linux*. Pre *Windows* to je *Winsock* a pre *Linux* je to *Libc*. Štruktúry a funkcie definované v týchto knižniciach umožňujú posielanie dát po sieti pomocou jazyka *C*. UDP správa s dátami je posielaná 100- krát za sekundu, keďže takáto frekvencia umožňuje vozidlu dostatočne rýchlo reagovať na nové dáta a zároveň ešte príliš nezaťažuje sieť [58; 59].

## 5.6 Zariadenie virtuálnej reality

Po analýze známych existujúcich zariadení virtuálnej reality sme dospeli k záveru, že zariadenie **Oculus Rift** bude najlepšou voľbou.

Zariadenia, ktoré používajú uchytený smartfón ako displej sme vylúčili, pretože takéto riešenie vnáša do prenosu obrazu veľké oneskorenie. Toto by malo nepriaznivý dopad na ponorenie do virtuálnej reality. Zo zvyšných zariadení, ktoré majú priamo pripojený vlastný displej, je momentálne dostupné len zariadenie *Oculus Rift* a to vo vývojárskej verzii *DK2*. Taktiež je medzi nimi cenovo najdostupnejšie.

Ak by ostatné zariadenia boli dostupné, mohli by poskytnúť iné výhody. Napríklad zariadenie *Vrvana Totem* má zabudovanú hardvérovú kompenzáciu skreslenia šošoviek, čo by uľahčilo implementáciu.

## 5.7 Oculus SDK

Pri integrácii zariadenia *Oculus Rift* bude použitý balíček nástrojov *Oculus SDK*. Pri integrácii sa bude postupovať najmä podľa odporúčaných krokov z dokumentácie dostupnej v SDK. Pomocou API sa bude zisťovať pozícia hlavy. Kombináciou predikcie pohybu hlavy implementovanej v SDK a manipulácie obrazu z kamery sa zlepší odozva na pohyby hlavy. SDK taktiež poskytuje funkcie pre časovanie a renderovanie obrazu.

### 5.7.1 Renderovanie skreslenia

Renderovanie skreslenia je možné robiť dvoma spôsobmi, buď použitím *Oculus SDK (LibOVR)* funkcií alebo priamo v aplikácii. Rozhodli sme sa pre **renderovanie pomocou SDK**, keďže sa preferuje tento spôsob riešenia. Tak isto je toto riešenie viac optimalizované, keďže sa používajú priamo funkcie SDK. Zjednoduší sa aj implementácia, pretože nie je potrebné samostatne riešiť mesh vertex, časovanie a predikciu.

### 5.7.2 OpenGL vs. Direct3D

Možnosti renderovania na zariadenie *Oculus Rift* sú dve: *OpenGL* a *Direct3D*. Po analýze sme sa rozhodli pre **OpenGL** z toho dôvodu, že je to otvorený štandard a funguje na rôznych platformách. *Direct3D* je podporovaný len na systémoch Microsoft Windows.

Toto API sa použije na renderovanie obrazov do *OpenGL* textúry, na ktorú sa aplikuje skreslenie. Na toto skreslenie sa využívajú funkcie SDK.

## 5.8 Softvérové spracovanie videa

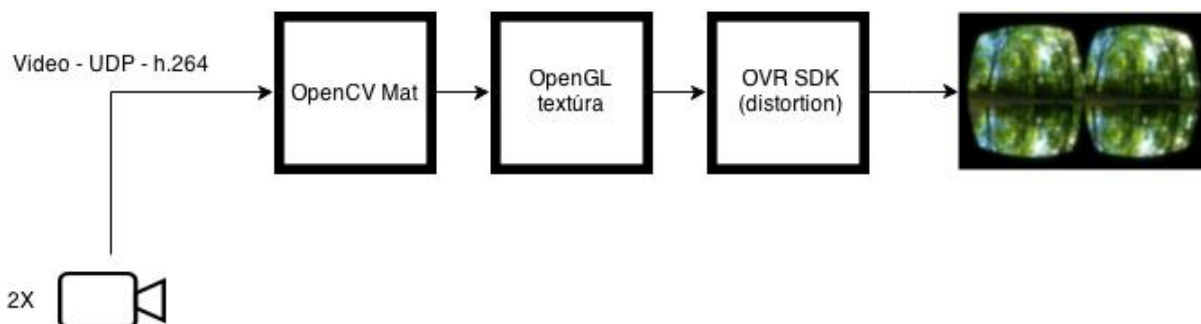
Pre spracovanie videa bude použitá multiplatformová knižnica **OpenCV**. Výhodou tejto knižnice je efektívnosť a jej silné zameranie na aplikácie v reálnom čase. Knižnica je široko používaná a preto by vyriešenie potencionálnych problémov mohlo byť urýchlené využitím podpory komunity.

Jej prednosťou je taktiež integrácia knižnice *FFmpeg*, ktorú plánujeme použiť na softvérové dekodovanie videa prijatého z vozidla.

*OpenCV* obsahuje množstvo algoritmov na spracovania obrázkov a videa, ktoré zefektívňujú vývoj. Niektoré z možností knižnice *OpenCV* sú rozpoznávanie tváre, gest, sledovanie objektov, rozšírená realita. Tieto funkcie sa môžu využiť pri rozširovaní nášho riešenia v budúcnosti.

### 5.8.1 Priebeh spracovania obrazu na centrálnej stanici

Na obrázku číslo 16 vidíme zjednodušený diagram spracovania obrazu z kamier. Video sa načíta zo streamu do *OpenCV*, následne sa presunie do *OpenGL* textúr. Ďalej sa na textúru aplikuje skreslenie. Tu sa použijú funkcie SDK ako časovanie a výmena buffer-ov.



Obrázok 16: Diagram spracovania obrazu

## 5.9 Voľba implementačného prostredia

Pre zvýšenie výpočtovej efektivity, priamu podporu všetkými použitými technológiami a zabezpečenie prenositeľnosti sme si pre implementáciu softvérovej časti na centrálnej stanici zvolili jazyk *C/C++*. Keďže balíček *Oculus SDK* je primárne vyvíjaný pre operačný systém *Microsoft Windows*, riešenie bude vyvíjané na tejto platforme. Pri vývoji však bude dbať na prenositeľnosť riešenia, najmä na platformu *Linux*. Podpora tejto platformy bude závisieť hlavne na podpore platformy balíčka *Oculus SDK* (pre túto platformu má byť k dispozícii čoskoro).



## 6 Zhodnotenie

---

V projekte sa podarilo identifikovať a analyzovať jednotlivé podúlohy potrebné pre vytvorenie vozidla s obsluhou pomocou zariadenia virtuálnej reality. Boli preskúmané dostupné algoritmy kompresie a dekompresie videa. Porovnali sa technológie, ktoré umožňujú bezdrôtový pres dát. Boli opísané možnosti pre riadenie a pohyb vozidla. Ďalej sa analyzovalo ovládanie vozidla pomocou ovládačov. Taktiež boli opísané známe zariadenia virtuálnej reality. Analýze sme bližšie podrobili zariadenie Oculus Rift a jeho integráciu do aplikácií. Preskúmali sa aj technológie používané na spracovanie videa a renderovanie grafického obsahu.

Bolo zvolené implementačné prostredie a vytvorila sa špecifikácia požiadaviek. Taktiež bol vytvorený návrh pre jednotlivé súčasti riešenia.

## Zoznam použitej literatúry

---

- [1]. Axis Communications. Video compression. [Online] [Dátum: 27. 10 2014.] [http://www.axis.com/products/video/about\\_networkvideo/compression.htm](http://www.axis.com/products/video/about_networkvideo/compression.htm) .
- [2]. **JORDAN, L.** Choosing the Best Video Codec. [Online] [Dátum: 27. 10 2014.] <http://www.larryjordan.biz/technique-choosing-the-best-videocodec/>.
- [3]. **Velleman.** The difference between MPEG-4 and H.264. [Online] [Dátum: 27. 10 2014.] [http://www.velleman.eu/downloads/3/h264\\_vs\\_mpeg4\\_en.pdf](http://www.velleman.eu/downloads/3/h264_vs_mpeg4_en.pdf).
- [4]. **Ohm, Jens-Rainer.** MPEG Developments in Multi-view Video Coding and 3D Video. [Online] [Dátum: 27. 10 2014.] [https://tech.ebu.ch/docs/events/3dtv09/presentations/ebu\\_3dtv09\\_ohm.pdf](https://tech.ebu.ch/docs/events/3dtv09/presentations/ebu_3dtv09_ohm.pdf).
- [5]. **Britz, Jochen.** Optimized implementation of an MVC decoder. [Online] [Dátum: 5. 11 2014.] [http://www.nt.uni-saarland.de/fileadmin/file\\_uploads/theses/master/Optimized\\_implementation\\_of\\_a\\_MVC\\_decoder.pdf](http://www.nt.uni-saarland.de/fileadmin/file_uploads/theses/master/Optimized_implementation_of_a_MVC_decoder.pdf).
- [6]. **Barbato, Luca.** The road to MVC. [Online] [Dátum: 6. 11 2014.] [http://blogs.gentoo.org/lu\\_zero/2014/04/04/the-road-to-mvc/](http://blogs.gentoo.org/lu_zero/2014/04/04/the-road-to-mvc/).
- [7]. **Karsten Müller, Philipp Merkle, Gerhard Tech, and Thomas Wiegand.** 3D Video formats and coding methods. [Online] [Dátum: 3. 11 2014.] <http://iphome.hhi.de/wiegand/assets/pdfs/ICIP10-3DV-formats.pdf> .
- [8]. Difference between Hardware Compression and Software Compression. *differencebetween.net*. [Online] [Dátum: 27. 10 2014.] <http://www.differencebetween.net/technology/difference-between-hardware-compression-and-software-compression/>.
- [9]. **Ďurovský, Ing. František.** NOVÉ BEZDRÔTOVÉ TECHNOLOGIE V ROBOTIKE. [Online] [Dátum: 4. 11 2014.] <http://www.sjf.tuke.sk/transferinovacii/pages/archiv/transfer/25-2013/pdf/031-034.pdf>.
- [10]. Zjednotí NFC všetky bezdrôtové technológie? *Elektronické súčiastky SOS electronic*. [Online] [Dátum: 28. 10 2014.] <http://www.sos.sk/?str=1260>.
- [11]. Bluetooth. *Wikipedia*. [Online] [Dátum: 28. 10 2014.] <http://en.wikipedia.org/wiki/Bluetooth>.
- [12]. WiFi. *Wikipedia*. [Online] [Dátum: 4. 11 20014.] <http://en.wikipedia.org/wiki/Wi-Fi>.
- [13]. Základné informácie o technológii – HSDPA. *Fony.sk*. [Online] [Dátum: 4. 11 2014.] <http://www.fony.sk/clanky/355->.
- [14]. Čo je LTE? *Technologia LTE - eStránky.sk*. [Online] [Dátum: 4. 11 2014.] <http://www.zpsfuska2011.estranky.sk/>.
- [15]. Všetko o 4G/LTE: história vo svete a príchod na Slovensko. *MojAndroid.sk*. [Online] [Dátum: ] 4. <https://www.mojandroid.sk/tema-tyzdna-4g-lte/>.
- [16]. Raspberry Pi. [Online] [Dátum: 5. 11 2014.] <http://www.raspberrypi.org/>.
- [17]. Arduino - Home. [Online] [Dátum: 5. 11 2014.] <http://arduino.cc/>.

- [18]. **Barr, Michael.** Introduction to Pulse Width Modulation (PWM). *Barr Group*. [Online] [Dátum: 5. 11 2014.] <http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>.
- [19]. DC Series Motors: High Starting Torque but No Load Operation III - Advies. *Ohio Electric Motors*. [Online] [Dátum: 5. 11 2014.] <http://www.ohioelectricmotors.com/dc-series-motors-high-starting-torque-but-no-load-operation-ill-advised-595>.
- [20]. **M. A. Laughton, D.F. Warne.** Electrical Engineer's Reference Book. [Online] [Dátum: 5. 11 2014.] [http://books.google.sk/books?id=5jOblzV5eZ8C&pg=SA19-PA4&dq=shunt+dc+motor&hl=en&ei=Zzy4TunfBKHd0QGAs\\_SrCg&sa=X&oi=book\\_result&ct=result&redir\\_esc=y#v=onepage&q&f=false](http://books.google.sk/books?id=5jOblzV5eZ8C&pg=SA19-PA4&dq=shunt+dc+motor&hl=en&ei=Zzy4TunfBKHd0QGAs_SrCg&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q&f=false).
- [21]. Hobby Servo. *Princeton University*. [Online] [Dátum: 5. 11 2014.] <https://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>.
- [22]. **Liptak, Bela G.** Instrument Engineers' Handbook, Fourth Edition, Volume Two: Process Control and Optimization. [Online] [Dátum: 5. 11 2014.] [http://books.google.co.uk/books?id=TxKynbyaIAMC&dq=Instrument+Engineers%27+Handbook&pg=P1&ots=jvrdPR7wxJ&sig=1hOUpQQDQH\\_8drYjW1yPvOcJSYI&hl=en&sa=X&oi=book\\_result&resnum=1&ct=result#v=onepage&q=Instrument%20Engineers'%20Handbook&f=false](http://books.google.co.uk/books?id=TxKynbyaIAMC&dq=Instrument+Engineers%27+Handbook&pg=P1&ots=jvrdPR7wxJ&sig=1hOUpQQDQH_8drYjW1yPvOcJSYI&hl=en&sa=X&oi=book_result&resnum=1&ct=result#v=onepage&q=Instrument%20Engineers'%20Handbook&f=false).
- [23]. Microstepping: Myths and Realities. *MICROMO*. [Online] [Dátum: 5. 11 2014.] <http://www.micromo.com/microstepping-myths-and-realities>.
- [24]. Relay Technical Notes. *American Zettler, Inc.* [Online] [Dátum: 5. 11 2014.] [http://www.azettler.com/pdfs/technical\\_notes.pdf](http://www.azettler.com/pdfs/technical_notes.pdf).
- [25]. Xbox 360 Controller for Windows. *Microsoft Hardware*. [Online] [Dátum: 5. 11 2014.] <http://www.microsoft.com/hardware/en-us/p/xbox-360-controller-for-windows>.
- [26]. About SDL. [Online] [Dátum: 5. 11 2014.] <https://www.libsdl.org/>.
- [27]. XNA 3.1 vs. SDL. *Stack Overflow*. [Online] [Dátum: 5. 11 2014.] <http://stackoverflow.com/questions/4141213/xna-3-1-vs-sdl>.
- [28]. Simple DirectMedia Layer. *Wikipedia*. [Online] [Dátum: 5. 11 2014.] [http://en.wikipedia.org/wiki/Simple\\_DirectMedia\\_Layer](http://en.wikipedia.org/wiki/Simple_DirectMedia_Layer).
- [29]. Microsoft XNA. *Wikipedia*. [Online] [Dátum: 5. 11 2014.] [http://en.wikipedia.org/wiki/Microsoft\\_XNA](http://en.wikipedia.org/wiki/Microsoft_XNA).
- [30]. The All New Oculus Rift Development Kit 2 (DK2) Virtual Reality Headset. *Oculus Rift*. [Online] [Dátum: 2. 10 2014.] <http://www.oculus.com/dk2/>.
- [31]. **Oculus.** Oculus Rift: Step Into the Game. *Kickstarter*. [Online] [Dátum: 2. 11 2014.] <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>.
- [32]. Oculus Rift. *Wikipedia*. [Online] [Dátum: 2. 11 2014.] [http://en.wikipedia.org/wiki/Oculus\\_Rift](http://en.wikipedia.org/wiki/Oculus_Rift).
- [33]. **Puga, Lauren.** Culture Five Alternatives to Oculus Rift. *GameSkinny*. [Online] [Dátum: 2. 11 2014.] <http://www.gameskinny.com/40hx8/five-alternatives-to-oculus-rift/>.

- [34]. PS4's Project Morpheus VR device won't launch in 2014, \$1000 price point unlikely. *Gamespot*. [Online] [Dátum: 2. 11 2014.] <http://www.gamespot.com/articles/ps4-s-project-morpheus-vr-device-won-t-launch-in-2014-1000-price-point-unlikely/1100-6418418/>.
- [35]. Project Morpheus (virtual reality). *Wikipedia*. [Online] [Dátum: 2. 11 2014.] [http://en.wikipedia.org/wiki/Project\\_Morpheus\\_\(virtual\\_reality\)](http://en.wikipedia.org/wiki/Project_Morpheus_(virtual_reality)).
- [36]. Vrvana Totem. *Vrvana*. [Online] [Dátum: 2. 11 2014.] <http://www.vrvana.com/>.
- [37]. **Vrvana**. Totem: the premium, full-featured Virtual Reality headset. *Kickstarter*. [Online] [Dátum: 2. 11 2014.] <https://www.kickstarter.com/projects/2091603040/totem-the-premium-full-featured-virtual-reality-he>.
- [38]. VRVana Cancel Totem Kickstarter Campaign, Pre-Orders Now Available. *Road To VR*. [Online] [Dátum: 2. 11 2014.] <http://www.roadtovr.com/vrvana-cancel-totem-kickstarter-campaign-pre-orders-now-available-instead/>.
- [39]. Samsung Gear VR. *Samsung*. [Online] [Dátum: 2. 11 2014.] [http://www.samsung.com/global/microsite/gearvr/gearvr\\_specs.html](http://www.samsung.com/global/microsite/gearvr/gearvr_specs.html).
- [40]. Gear VR. *Wikipedia*. [Online] [Dátum: 2. 11 2014.] [http://en.wikipedia.org/wiki/Gear\\_VR](http://en.wikipedia.org/wiki/Gear_VR).
- [41]. Poznáme cenu headsetu Samsung Gear VR. *zive.sk*. [Online] [Dátum: 2. 11 2014.] <http://www.zive.sk/clanok/98486/pozname-cenu-headsetu-samsung-gear-vr>.
- [42]. Avegant Glyph. *Avegant*. [Online] [Dátum: 2. 11 2014.] <http://avegant.com/>.
- [43]. Glyph: A Mobile Personal Theater With Built In Premium Audio. *Kickstarter*. [Online] [Dátum: 2. 11 2014.] <https://www.kickstarter.com/projects/avegantglyph/a-mobile-personal-theater-with-built-in-premium-au>.
- [44]. Cardboard. *Google*. [Online] [Dátum: 2. 11 2014.] <https://cardboard.withgoogle.com/>.
- [45]. Cardboard. *Google Developers*. [Online] [Dátum: 2. 11 2014.] <https://developers.google.com/cardboard/overview>.
- [46]. TIP: Kúpte si virtuálne okuliare Google Cardboard za 2,55€ s poštovým zdarma. *AndroidPortal.sk*. [Online] [Dátum: 2. 11 2014.] <http://androidportal.zoznam.sk/2014/09/tip-google-cardboard-2-55-eur-postovne-zdarma/>.
- [47]. Oculus Developer Guide, SDK Version 0.4. *Oculus VR, LLC*. [Online] 23. 10 2014.
- [48]. Distortion (optics). *Wikipedia*. [Online] [Dátum: 3. 11 2014.] [http://en.wikipedia.org/wiki/Distortion\\_\(optics\)](http://en.wikipedia.org/wiki/Distortion_(optics)).
- [49]. OpenGL 3 & DirectX 11: The War Is Over. *Tom's Hardware*. [Online] [Dátum: 30. 10 2014.] <http://www.tomshardware.com/reviews/opengl-directx,2019.html> .
- [50]. An intro to modern OpenGL. *Durian Software*. [Online] [Dátum: 30. 10 2014.] <http://duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Chapter-1:-The-Graphics-Pipeline.html>.
- [51]. About. *OpenCV*. [Online] [Dátum: 4. 11 2014.] <http://opencv.org/about.html>.
- [52]. OpenCV. *Wikipedia*. [Online] [Dátum: 4. 11 2014.] <http://en.wikipedia.org/wiki/OpenCV>.

- [53]. AForge.NET :: Framework. [Online] [Dátum: 5. 11 2014.] <http://www.aforgenet.com/framework/>.
- [54]. Integrating Vision Toolkit. [Online] [Dátum: 5. 11 2014.] <http://ivt.sourceforge.net/>.
- [55]. Metaverzum Sociálna prezencia v plne imerzných virtuálnych prostrediach. [Online] [Dátum: 1. 11 2014.] [http://psych.fss.muni.cz/files/remiar\\_jakub\\_bratska\\_maria\\_fif\\_uk.pdf](http://psych.fss.muni.cz/files/remiar_jakub_bratska_maria_fif_uk.pdf).
- [56]. My Oculus FPV setup in an R/C car with head tracking for camera pan/tilt. *reddit*. [Online] [Dátum: 4. 11 2014.] [http://www.reddit.com/r/oculus/comments/1say05/my\\_oculus\\_fpv\\_setup\\_in\\_an\\_rc\\_car\\_with\\_head/](http://www.reddit.com/r/oculus/comments/1say05/my_oculus_fpv_setup_in_an_rc_car_with_head/).
- [57]. Handling Joysticks. *SDL*. [Online] [Dátum: 5. 10 2014.] [http://sdl.beuc.net/sdl.wiki/Handling\\_Joysticks](http://sdl.beuc.net/sdl.wiki/Handling_Joysticks).
- [58]. Sockets Tutorial. *LinuxHowtos.org*. [Online] [Dátum: 5. 11 2014.] [http://www.linuxhowtos.org/C\\_C++/socket.htm](http://www.linuxhowtos.org/C_C++/socket.htm).
- [59]. Sockets Tutorial. *Computer Science at RPI*. [Online] [Dátum: 5. 11 2014.] <http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>.