Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar

Szakdolgozat

Kamera alapú beltéri navigáció az AR drone eszközön

Petrovicz Benedek
Mérnök informatikus BSc
2017

Témavezető:
dr. Zsedrovits Tamás

Alulírott Petrovicz Benedek a Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Karának hallgatója kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem és a szakdolgozatban csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen a forrás megadásával megjelöltem. Ezt a Szakdolgozatot más szakon még nem nyújtottam be.

_____

Petrovicz Benedek

Tartalomjegyzék

Kivonat

A dolgozat arra törekszik, hogy az automatizált drónvezérlés témában érdekelt személyeknek, eligazítást és kiindulási alapot adjon. Amellett, hogy elméleti síkon, általános áttekintést ad a témáról, mellékletként tartalmazza ezen gondolatok tényleges megvalósítását is.

A kidolgozás kifejezetten olyan drón kutatókat céloz, akik pontos méréseket szeretnének végezni és ezáltal hatékony algoritmusokat fejleszteni. A kutatás során felhasznált összes eszköz, fontos részét képezi a rendszernek, mivel bármelyik elhagyása esetén használhatatlanná válik a kidolgozott kód. Bár az eszközök szimulálására van lehetőség, de az nagy mértékben ronthatja a mérések eredményét.

A kutatás során felhasznált drón a Parrot cég AR.Drone 2.0 nevű terméke. Viszonylag kis térben is biztonságos körülmények között használható. A biztonságot maga a drónra szerelt kemény szivacs alapú keret adja, mely amellett, hogy védi a drón propellereit, azt is megakadályozza, hogy a környezetében nagyobb kárt tudjon tenni. További biztonságot nyújt a drón fedélzeti számítógépén futtatott szkript, ami azonnal leállítja a drón rotorjait, amint azt érzékeli, hogy az eszköz elérte a 90 fokos dőlésszöget, ami által könnyedén megállítható az akkumulátorhoz való hozzáférés nélkül is.

A felhasznált eszközök közül a második legfontosabb, az Optitrack cég mozgáskövetésre alkalmas kamerarendszere. Ez képes akár centiméteres pontossággal lekövetni és háromdimenziós térben ábrázolni lényegében bármit, ami fel van szerelve a kamerákhoz kidolgozott fényvisszaverő markerekkel.

Az eddig felsorolt eszközök kommunikációját pedig, egy Linux rendszerrel ellátott gépen futó Robot Operating System (ROS) teszi lehetővé. Ezt a nyílt forráskódú rendszert széles körben alkalmazzák a robotikával foglalkozó cégek és kutatók, így rengeteg eszközhöz érhetők el különböző felhasználásra szánt implementációk.

A repülés közbeni utasítás közvetítést, QR kódok és a drón 2 kamerája segíti elő. A jelek felismerését a kamera képéből az ar_track_alvar nevű ROS csomag valósítja meg. Ez a csomag kifejezetten jól működik gyenge felbontású kamera remegő képével is, amiből nincs hiány egy ilyen projekt során. Az egyetlen hátrány abból adódik, hogy a drón egyszerre csak az egyik kamerája képét tudja közvetíteni a rendszernek.

A megvalósított forráskód egy ROS csomag formájában használható fel. Fel van készítve a rendszer minden összetevőjének vezérlésére és az adataik feldolgozására. A felhasználónak csupán az eszközök IP címeit kell pontosan megadnia a csomag számára, illetve, hogy drón melyik kameráját szeretné használni a QR kódok felismerésére.

A QR kódon szereplő adat alapján több különböző előre megírt algoritmust is képes végrehajtani. A projektben csak előre kiszámolt animációkat kapott meg parancsként a drón, amelyek segítségével képes átrepülni vagy egy irányban kikerülni az előtte lévő akadályt.

További fejlesztések során már csak ezzel a részével kell foglalkoznia az adott kutatónak. További szenzorok adatainak a felhasználása is egyszerűen megoldható. Mivel a kód python nyelven készült, gyorsan és problémamentesen lehet tovább alakítani.

Abstract

The thesis pursues at giving a guidance and a starting-point to people who are intrested in automated drone controlling. Besides that it gives a general theoretical overview of the topic, it contains the implementation of these thoughts as an attachment.

This work mostly aims at those drone researchers, who want to take accurate measurements and to develop effective algorithms with them. All of the items used in this research are important parts of the system, because leaving out any of them makes the code unusable. Although simulating them is another possibility, but it spoils the measurement results greatly.

The drone used in this research is the product of the Parrot company called AR.Drone 2.0. It can be used in relatively small places, under safe conditions also. The main reason of this safety is the hull of the drone made out of some hard foam. Besides that it protects the propellers of the drone, ensures that the drone is not causing any great damage in the environment around it. A script running on the on-board computer of the drone provides additional security, because it blocks the rotors of the drone as soon as it detects that the drone is tilted by 90 degrees so it can be stopped easily without reaching its battery.

The second most important item used in the research is the camera system of Optitrack that is capable of motion tracking. It is able to track with centimeter accuracy and represent in a three dimensional space essentially anything that is equipped with the reflective markers designed for these cameras.

These two items are communicating through Robot Operating System (ROS) on a computer with Linux. This open-source framework is used widely by companies and researchers intrested in robotics, so there is a wide variety of implementations for a lot of tools and devices.

Giving instructions mid-air, is solved by detecting and decoding QR codes from the video feed of the two camera on the drone. The detection from the camera feed is done by the ar_track_alvar ROS package. This package works really good with shaky image of low resolution cameras, that is frequent in projects like this. The only drawback is, that the drone can broadcast the video feed of only one camera at once.

The implemented source code can be used as a ROS package. It is suitable for controlling all of the components in the system and processing their data. The user is only needed to give the correct IP address of the items, and to choose which camera is used for detecting QR codes.

It is possible to call multiple different algorithms according to the data in the QR code. In the project there were only pre defined animations for bypassing over or next to an obstacle that is before the drone.

In further development processes this part is only intended to be worked on by the researcher. Data of additional sensors can be used easily. As the code has been written in python it can be formed rapidly fast and without any problem.

1. Bevezetés

A kutatás fő célja a Pázmány Péter Katolikus Egyetem Információs Technológiai és Bionikai Karán elindított UAV laboratórium alapjainak kidolgozása és lefektetése. Rengeteg lehetőség rejlik ebben a gyerekcipőben járó technológiában, így fontosnak tartom, hogy minél előbb megismerjék a hallgatók és részesei lehessenek a fejlesztésének.

A tervezés fő szempontja egy olyan rendszer megvalósítása, ahol az AR drón az általunk írt szkriptek segítségével, automatizált módon képes bizonyos feladatokat végrehajtani. Az OptiTrack kamerarendszer közreműködésével, egy olyan teszt környezetet hozhatunk létre, ami amellett, hogy megbízható, rengeteg plusz információval szolgál a hatékonyság növelésének érdekében. Könnyedén szimulálhatunk vele különböző háromdimenziós terekben való mozgást.

Fontos lépés lehet ez az egyetem életében, mivel a jó alapok megteremtése ebben a témában, nagyon jó lehetőségeket nyithat a jövő hallgatói felé. Aki nem zárkózik el teljes mértékben a közösségi médiáktól, az nagy eséllyel hallott vagy olvasott már drónokról és azok alkalmazásáról szóló cikket. Úgy gondolom ma pontosan abban a világban élünk, ahol ez az egyik leginkább felkapott téma fiatalok és idősebbek között is, mivel olyan piaci területeket nyitott meg a technológia, ami eddig még ismeretlen volt sokak számára.

1.1 Alkalmazási területek

ar_track_alvar: landing position with bottom camera

Coordinate z means, the altitude from the marker

Coordinate x, y should be as close as possible to 0

optitrack: presents reference coordinates until marker can not be seen

2. Feladatkiírás

A feladatok közé tartozik a drón és a kapcsolódó szakirodalom tanulmányozása és minél részletesebb megismerése. Ez magában foglalja a drón specifikációjának és a felhasználható eszközök pontos ismeretét. Ezek elengedhetetlen részei a drónnal végzett hatékony munkának.

A repülési tesztek, illetve a QR kód felismerés szigorú előfeltételeit képezik a kutatás bármelyik további munkafolyamata elvégzésének. Hiszen ezek a projekt fő építőelemei.

Az akadálypálya építés és az OptiTrack kamerarendszer adatainak felhasználása már több lehetőséget tárt elém, hiszen itt bármilyen beltéren elvégezhető folyamatot elképzelhettem, amit aztán igyekeztem megvalósítani.

2.1 Szakirodalom tanulmányozása

Lorem ipse

2.2 Repülő megismerése, irányításának megtanulása

Lorem ipse

2.3 Repülési tesztek végzése

Lorem ipse

2.4 QR kódok felismerése a repülő kamerájának segítségével

Lorem ipse

2.5 Mesterséges akadálypálya tervezése és építése

Lorem ipse

2.6 AR Drone irányítása a repülő szenzoraiból, valamint az Optitrack rendszerből származó információ és ROS segítségével

Lorem ipse

2.7 Elkészült munka és mérések dokumentálása

Lorem ipse

3. Előzmények

Hasonló alkotás pl: https://github.com/krishnan793/ar2landing_neural

A témában fellelhető szakirodalmi alkotások mennyisége sajnálatos módon igen csekély. Lévén, hogy egy olyan technológiáról van szó, ami csupán pár éve törte be magát a köztudatba, még nem alakult ki az a megbízható forrásokkal szolgáló környezet, mint ami egy évtizedek óta ismert technológia esetében nagy eséllyel megtalálható.

Az ehhez hasonló fejlesztések esetében leginkább a nyílt forráskódú projektekre és azok leírásaira lehet hagyatkozni, ám ebben az esetben nagyon körültekintően és kritikus szemmel kell szemlélődni, mivel köztudottan ezek a projektek sokszor rejthetnek hibákat. Ezek a hibák pedig egy nagyobb méretű drón esetében akár komoly következményekkel is járhatnak, mint ahogy az a 2017 május 9-én rendezett Golden State Race Series biciklis versenyen is megtörtént, ahol az egyik résztvevő biciklijét találta el egy drón miután belecsapódott egy közeli fába. [1]

irodalomkutatás, hasonló alkotások

4. A tervezés részletes leírása

Miután sikeresen teszteltem a drón, az OptiTrack kamerarendszer és a Robot Operating System működését, kezdődhetett a tervezés.

Volt itt 2 féle QR kód felismerős izé, egy landolós projekt, szimulátor (ajánlott később)

Saját package (python), saját launch file-al, 3 részre bontott forráskóddal.

a döntési lehetőségek értékelése és a választott megoldások indoklása

4.1 A munka során felhasznált eszközök

A Parrot cég által gyártott AR.Drone 2.0, ami egy távolról vezérelhető, kifejezetten beltéri használatra felkészített négyrotoros repülőgép. Wi-Fi kapcsolaton keresztül képes kommunikálni más eszközökkel. Tartalmaz egy 4GB-os tárhelyet ami lehetővé teszi, hogy egyedi algoritmusokat töltsünk fel és automatizáljuk a működését. A drón alsó részén található egy ultrahangos távolságmérő és egy légnyomásmérő szenzor, továbbá be van építve még egy 3 tengelyű giroszkóp, egy 3 tengelyű gyorsulásmérő szenzor és egy magnetométer, amik a drón stabilan tartását hivatottak biztosítani. [2]

8 darab OptiTrack Prime 13 típusú kamerából összeállított kamerarendszer. Ezek a kamerák UTP kábeleken keresztül kommunikálnak az őket vezérlő számítógéppel.

Egy NETGEAR ProSafe GS728TPP típusú switch, ami képes hálózati portokon keresztül meghajtani az OptiTrack kamerákat, illetve lebonyolítja a kommunikációt a laptopok és a kamera rendszer között.

Szükség esetére felkonfiguráltunk egy LINKSYS AC1900 típusú routert is, ami automatikusan csatlakozik a drónhoz, mikor látja. Így a kamerákon kívül minden más eszközünket tudtuk használni vezeték nélküli hálózattal a teszteléshez.

2 darab számítógép. Az egyik Windows 10 operációs rendszerrel, míg a másik Ubuntu Xenial operációs rendszerrel felszerelve.

4.2 A tesztkörnyezet felépítése

Ide jöhet Bartha mester tompaszögű hegyesszöge

4.3 A munka során felhasznált forráskódok

ROS, ardrone_autonomy, ardrone_tutorials, ar_track_alvar, vrpn

4.4 Robot Operating System

A Robot Operating System vagy rövidebben csak ROS egy nyílt forráskódú, általánosított „operációs rendszer” robotokhoz. Megvalósítja a hardver szintű absztrakciót, képes kezelni a hardverek vezérlését és az egyes folyamatok közti kommunikációt.

A ROS rengeteg olyan eszközt és könyvtárat biztosít, ami lehetővé teszi a különböző platformokon való fejlesztést. Támogatja a kód újra felhasználást, ami nagymértékben megkönnyíti a fejlesztők munkáját. A ROS rendelkezik beépített vizuális megjelenítővel is, ami kiváló segítséget nyújt a teszt környezetünk megjelenítéséhez a kódot futtató gépen.

A ROS rendszer építőelemei:

- **Node**: futtatható állomány, ami a ROS-t használja a többi node-dal való kommunikációhoz
- **Message**: primitív típusú adat vagy tömb, amit a node-ok küldhetnek és fogadhatnak
- **Topic**: a node-ok üzeneteket publikálhatnak bele, míg mások feliratkozhatnak rá, hogy megkapják az üzeneteket. Lényegében ezekből épül fel a binárisok közti kommunikációs csatorna
- **Service**: egy node-ban implementált szolgáltatás, ami a ROS-on keresztül meghívható

- **Master**: a ROS rendszer központi eleme, ami számon tartja a különböző node-okat, topic-okat, feliratkozásokat és publikálásokat
- **Parameter Server**: központi tároló, ami bármelyik node számára elérhető

[3]

4.5 Ardrone Autonomy

Az ardrone_autonomy egy ROS driver a Parrot AR.Drone 1.0 és 2.0-hoz. A ROS-on belül egy külön node-ként jelenik meg, ami a következő fontos topic-okat használja:

- **ardrone/navdata**: a drón navigációs adatait közvetíti
- **ardrone/image_raw**: a kamerák képeit közvetíti

Továbbá még számos *service* is meghívható benne, amivel a drón fedélzeti számítógépét vezérelhetjük. [4]

4.6 Virtual-Reality Peripheral Network

A VRPN egy platform független rendszer, ami interfészként működik szoftverek és fizika eszközök között. A ROS-sal való topic orientált kommunikációt is képes kezelni. Szerver – Kliens modell alapján működik és Internet Protokoll segítségével kommunikál más eszközökkel. [5]

4.7 Dokumentáció

A munka elején elindítottunk egy weboldalt, ahol folyamatos dokumentációt vezettünk arról, hogy mi hogyan oldottuk meg az egyes feladatokat. Ez számunkra is kedvező abban az esetben, ha egy hiba esetén vissza akarjuk követni a folyamatot. Egyelőre még nem publikus a weboldal. [8]

4.8 A ROS és a drón összekapcsolása

Mi a jelenlegi legújabb, Kinetic Kame nevű ROS verziót használjuk, amelyet Xenial verziójú Ubuntu-ra telepítettünk fel. Ahhoz, hogy vezérelni tudjuk a drónt, szükségünk volt egy driver-re, ami lehetővé teszi a ROS és az AR.Drone közti kommunikációt.

Az *ardrone_autonomy* ROS driver volt az, ami számunkra kedvező módon támogatta az általunk használt drónt. A segítségével le tudtuk kérdezni a drón navigációs adatait és kameraképeit, illetve vezérlő utasításokat tudtunk küldeni a fedélzeti számítógépnek.

4.9 Motive

Az OptiTrack rendszer hivatalos szoftvere a Motive. Képes a kamerák által modellezett háromdimenziós tér megjelenítésére és annak konfigurálására. [9]

A kamerákat UTP kábelekkel csatlakoztattuk egy switch-en keresztül a Motive szoftvert futtató géphez, ami előzetes beállítások nélkül azonnal felismerte őket. A kamerák kalibrálásához néhány ezer mintát kellett gyűjteniük a megfigyelt térből. A mintákat egy erre a célra készített kalibráló rúddal generáltuk manuálisan. [10]

A kalibrálást követően a szoftver kiszámítja a kamerák egymáshoz viszonyított pozícióit és elkészül a modellezett tér, amiben az eszközök helyzetét valós időben figyelhetjük meg.

A drónt felszereltük egy marker állvánnyal és 5 darab markerrel, ami alapján a kamerák egyértelműen meg tudják határozni a pozícióját és orientációját.

4.10 Kommunikáció a Motive és a ROS között

A drónon lévő markereket egyesíteni lehet szoftveresen és generálni lehet belőlük egy egybefüggő testet, ami az eredeti pontok által meghatározott alakzat súlypontjában tesz egy pivot point-ot. Ezzel a ponttal és a ráillesztett orientációs vektorral határozhatjuk meg egyszerűen a drón helyzetét. [11]

A Motive rendelkezik beépített VRPN szerverrel, amivel képes bármelyik marker adatait közvetíteni. A közvetített adat egy topicba kerül, aminek a neve megegyezik a markerekből generált test súlypontjának nevével. Ezt szükség esetén meg is változtathatjuk az átláthatóság érdekében.

Ezek után már nem volt más dolgunk, mint az általunk közvetített topic-ban szereplő adatokat feldolgozni és ezek alapján megfelelő vezérlő parancsokat küldeni a drón fedélzeti számítógépének.

5. Eredmények

A feladatunk végére érve kialakult a végső fejlesztői környezet, ami mindent biztosít ahhoz, hogy a fejlesztő által implementált kódot probléma nélkül tudja futtatni.

Az *ardrone_tutorials* nevű, nyílt forráskódú projekt tartalmazza a teszteléshez szükséges vezérlési parancsokat és a mintájára könnyedén készíthető saját alkalmazás. A projekt forráskódja python nyelven íródott, és az *ardrone_autonomy* által biztosított parancsokat használja.

A topic-okban közvetített adatokat könnyedén vizualizálhatjuk *rviz*-zel a ROS beépített grafikus megjelenítőjével. A kezelőfelületén megjelennek az aktuális topic-ok és a benne szereplő adatok, amelyeket megjeleníthetünk 2D-s vagy 3D-s környezetben. [12]

Amennyiben nem labor környezetben vagyunk és nincs lehetőségünk az AR.Drone-nal és OptiTrack kamerákkal tesztelni, használhatjuk a *tum_simulator* nevű programot, ami részben képes kiváltani a valós rendszert. Ez az otthoni fejlesztést is lehetővé teszi. [13]

6. Értékelés

Kritika – továbbfejlesztés (pl: swarming, mozgó objektumra landolás)

Az OptiTrack kamera rendszer igencsak érzékeny bármiféle változásra, viszont jól beállított konfiguráció és kalibrálás esetén nagyon pontos adatokat képes közvetíteni. Kiválóan használható a drón-ra fejlesztett alkalmazások teszteléséhez. A rendszer összeállítása és kalibrálása általában 15-20 percet vesz igénybe és nagyon egyszerűen elvégezhető. Nekünk csak a kamerákat és a számítógépet kell csatlakoztatnunk, illetve a kalibráló rudat mozgatnunk a megfigyelt területen. A számításokat elvégzi a Motive szoftver.

Amíg a drón repülés közben nem kap újabb parancsokat, elkezd lebegni. Ilyenkor próbál egy helyben maradni, amit az aljára szerelt ultrahang szenzor segítségével próbál megvalósítani. Azt tapasztaltuk, hogy tükröződő és egyszínű felületek fölött repülve nem

képes a helyben maradásra. További problémát jelent, hogy kis térben a falak közelsége megnöveli a drón-ra ható turbulenciát, ami miatt szintén képtelen megtartani a helyzetét. Ezekre a problémákra megoldást jelenthet, ha a kamerarendszerből kapott információk alapján próbáljuk egy helyben tartani a drónt, amíg nem adunk neki további parancsokat.

7. Összefoglalás

teljesítettségi mutató a kiírás függvényében

8. Köszönetnyilvánítás

Szeretnék köszönetet nyilvánítani a karnak a biztosított eszközökért és támogatásért. Nagyra tartom, hogy ilyen kifinomult dolgokkal van lehetőségünk dolgozni.

Továbbá köszönet illeti a már korábban említett nyílt forráskódú szoftver csomagok és könyvtárak fejlesztőit, akik remek munkát végeztek a témában.

9. Irodalomjegyzék

[1] „Golden State Race Series,” [Online].

[2] „Parrot AR.DRONE 2.0 Power Edition | Parrot Store Official,” Parrot, 2017. [Online]. Available: https://www.parrot.com/us/drones/parrot-ardrone-20-power-%C3%A9dition. [Hozzáférés dátuma: 04 05 2017].

[3] „Documentation - ROS Wiki,” 31 03 2017. [Online]. Available: http://wiki.ros.org/. [Hozzáférés dátuma: 05 05 2017].

[4] „ardrone_autonomy — ardrone_autonomy indigo-devel documentation,” 04 2014. [Online]. Available: http://ardrone-autonomy.readthedocs.io/en/latest/. [Hozzáférés dátuma: 05 05 2017].

[5] „vrpn Wiki,” 21 01 2017. [Online]. Available: https://github.com/vrpn/vrpn/wiki. [Hozzáférés dátuma: 06 05 2017].

[6] „Introduction · PX4 Developer Guide,” 2017. [Online]. Available: https://dev.px4.io/en/. [Hozzáférés dátuma: 07 05 2017].

[7] R. Roy és V. Bommakanti, „ODROID-XU4 Beginner's Guide,” 2015. [Online]. Available: https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf.

[8] A. Bartha és B. Petrovicz, „uavLab,” 2017. [Online]. Available: https://uavlab.itk.ppke.hu/. [Hozzáférés dátuma: 08 05 2017].

[9] „Motive Documentation - NaturalPoint Product Documentation,” 19 08 2016. [Online]. Available: http://wiki.optitrack.com/index.php?title=Motive_Documentation. [Hozzáférés dátuma: 06 05 2017].

[10] „OptiTrack - Calibration Tools,” 2017. [Online]. Available: https://optitrack.com/products/tools/. [Hozzáférés dátuma: 06 05 2017].

[11] „Rigid Body Tracking - NaturalPoint Product Documentation,” NaturalPoint Corporation, 25 01 2017. [Online]. Available: http://wiki.optitrack.com/index.php?title=Rigid_Body_Tracking. [Hozzáférés dátuma: 07 05 2017].

[12] „rviz - ROS Wiki,” 15 06 2016. [Online]. Available: http://wiki.ros.org/rviz. [Hozzáférés dátuma: 07 05 2017].

[13] „tum_simulator - ROS Wiki,” 12 05 2014. [Online]. Available: http://wiki.ros.org/tum_simulator. [Hozzáférés dátuma: 07 05 2017].

[14] „Press Releases | Airobotics,” Airobotics, 2017. [Online]. Available: http://www.airobotics.co.il/press-releases/. [Hozzáférés dátuma: 08 05 2017].

[15] „How Drone Will be Used In The Future - Business Insider,” 16 12 2013. [Online]. Available: http://www.businessinsider.com/how-drones-will-be-used-in-the-future-2013-12. [Hozzáférés dátuma: 08 05 2017].

[16] „From Driverless Cars to Automated Drones - The Future is Already Here | Airobotics,” 24 11 2016. [Online]. Available: http://www.airobotics.co.il/blog/driverless-cars-automated-drones-future-already/. [Hozzáférés dátuma: 08 05 2017].

10. Mellékletek

Ha van