

Robotos pakolási feladat megoldása környezetérzékelés segítségével

Robotic pick-and-place operation enhanced with environment sensing

HAJÓS Mátyás¹, BSc. szoftverfejlesztő, HORVÁTH Dániel¹, MSc. PhD. hallgató

¹ SZTAKI, Mérnöki és Üzleti Intelligencia Kutatólaboratórium
H-1111 Budapest, Kende utca 13-17.; Telefon: +36 1 279 6000; Fax: +36 1 466 7503
hajos.matyas|horvath.daniel@sztaki.hu; www.sztaki.hu

Kivonat

A gyártástudománnyal kapcsolatos kutatások egyik fő iránya a robotok környezetérzékelési és döntéshozatali képességének bővítése, hogy hatékonyan együttműködhessenek más robotokkal vagy emberekkel. Jelen dolgozat egy félig strukturált környezetben zajló pakolási, szerelési feladat megoldását mutatja be, fa építőközből történő absztrakt összeszerelésen keresztül. A köbök felismerése neurális hálót és hagyományos képfeldolgozási technikákat kombináló megoldással történik. A felismert darabokat egy olyan építési sorrendtervnek megfelelően helyezi a robot a másik palettára, amely terv 3D modell alapján automatikusan lett generálva.

Kulcsszavak: robotika, gépi látás

Abstract

One of the main interest in modern manufacturing science is to broaden the environment sensing and the decision-making abilities of robotic systems, in order to perform complex tasks in human-robot collaboration. The paper discusses an abstract pick-and-place operation with wooden blocks in a semi-structured environment. Combination of neural network and machine vision is used to detect the position of the blocks. The aim of the operation is defined by a 3D model, and the order of assembly is generated automatically.

Keywords: robotics, machine vision

1. BEVEZETÉS

A gyártástudomány és a számítástechnika együttes fejlődésével alakultak ki a kiberfizikai gyártórendszerek (CPPS, Cyber Physical Production System). Az elnevezés a fizikai világgal szoros kapcsolatban levő egységek összességére utal, melyek a termelés minden szintjén együttműködnek egymással [1]. Ezen rendszerek egyik legfontosabb eleme, azon gépek és robotok, melyek szenzorjaik segítségével képesek érzékelni környezetük változásait. Valamint az információikat egymás között megosztva intelligens módon együttműködnek, hogy maximalizálják hatékonyságukat.

Ennek megfelelően a modern robotrendszereknek képesnek kell lenniük alkalmazkodni a környezet változásaihoz. Hasonló problémát jelent az adaptív összeszerelés megvalósítása is [2][3]. Mindkettőhöz elengedhetetlen a fizikai munkatér változásainak követése. A környezet érzékelésének egy alapvető módja a vizuális, képi információk feldolgozása. A hagyományos képfeldolgozási eljárások (Machine Vision) megfelelő megvilágítás mellett specifikus megoldást nyújtanak adott feladatokhoz. Ezzel szemben az emberi látás modellezésével foglalkozó gépi látás (Computer Vision) területén alkalmazott mély neurális hálók az objektumok általános reprezentációját tanulják meg, így széleskörben használhatók. [4] [5]

A dolgozatban szereplő robot cella demonstrációs céllal készült, kialakításában a fenti szempontok figyelembevételével. Működésének bemutatására egy absztrakt pakolási feladat szolgál, melyben többszínű

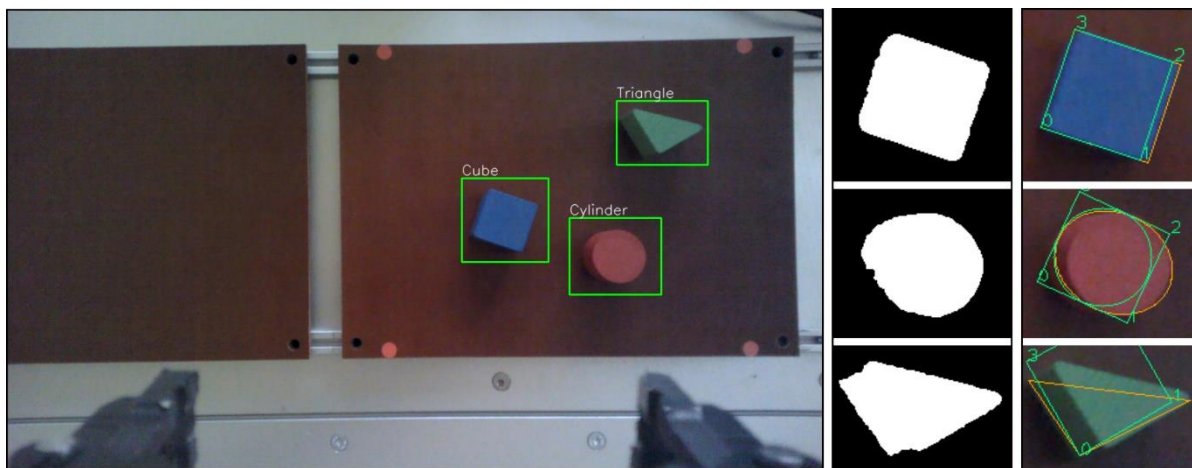
(piros, zöld, kék) fa építőelemeket (kocka, henger és háromszög alapú hasáb) használunk. Így összesen kilencféle építőelem áll rendelkezésre. A pakolást két bakelit paletta között hajtja végre a Universal Robots UR5 robotkar, amelyre erő/nyomaték szenzor, adaptív megfogó és kamera lett felszerelve.

2. OBJEKTUM DETEKTÁLÁS

Az említett építőelemek különböző számú egyensúlyi helyzettel rendelkeznek. A kockának egy, a hengernek kettő (alaplaján vagy palástján áll), a háromszögnek pedig három (alaplaj, valamelyik kisebb oldallaj, vagy a legnagyobb oldallaján áll) a képfeldolgozás szempontjából megkülönböztethető egyensúlyi helyzete van. A fejlesztés első lépésében minden elem csak egyféle helyzetben szerepelhetett, hogy az alakzatok felülnézeti képeik alapján egyszerűen megkülönböztethetők legyenek, vagyis a henger és a háromszög alapú hasáb csak alaplaján állhat.

A képkészítés a robotkara szerelt kamera segítségével történik. A képfeldolgozás folyamata 4 fő lépésre bontható. Az első lépésben az megkeresi a kép azon részeit melyek valamilyen építőkockát tartalmaznak, ehhez a YOLO [4] („You Only Look Once”) neurális háló harmadik verzióját [6] választottuk. A legtöbb objektum detektálást végrehajtó háló valamilyen osztályozó eljárást használ, amelyet a kép különböző részein alkalmazva kapják az objektumok pozícióit és valószínűségeit. A YOLO esetében ezzel szemben regressziós problémaként közelítik meg az objektum felismerést és egyetlen neurális háló ad becslést az objektumok pozícióira és valószínűségeire. Emiatt sokkal gyorsabb, és bár nincsen egyelőre erről mért adat, de remélhetőleg közel valós idejű detektálásra is alkalmazható. További előnye, hogy a háló mélyebb rétegei előtanítottak az ImageNet 1000 osztályból álló adathalmazán, így a tanítási folyamathoz nincs szükség milliós nagyságrendű tanuló adatbázisra [4][7]. A neurális háló kimenete az alakzatok középpontjai és a tengelyekkel párhuzamos befoglaló téglalapok (bounding box), orientációt nem tartalmaz.

A bounding box téglalapok kijelölik azt a ROI (region of interest) területet, ahol az orientáció meghatározását végrehajtó klasszikus algoritmusokat alkalmazzuk. Első lépésként egy meanshift szűrés a zajok csökkentésére és viszonylag nagyobb homogén területek létrehozására szolgál. Ezután a három színnek megfelelő maszkolásokkal a kép binarizálása történik, mely alapján döntést lehet hozni az alkatrész színéről és előáll az alakzat körvonala is a bináris képekre a neurális háló osztályai alapján illetve téglalapot, ellipszist vagy háromszöget. Végül az így kapott alakzatoknak az elforgatott határoló téglalapjai segítik a megfogási pozíció kiszámítását. A képfeldolgozás lépéseit az 1. ábra mutatja.



1. ábra A készített kép a neurális háló által felismert alakzatokkal (balra). Alakzatok bináris képe színszűrés után (középen) és az illesztett elforgatott határoló téglalapok (jobbra)

3. ALKATRÉSZEK MEGFOGÁSA

A fenti módszerrel kapott hely a pixelben megadott képi koordináta. Ahhoz, hogy a robot a megfelelő pozícióba mozogjon ezt át kell számítani a robot alap koordináta-rendszerébe. A transzformációt két lépésben végzi az algoritmus. Először a paletta lokális bázisához képest határozza meg a helyzetet. A pixel-valós transzformációt tartalmazó homogén transzformációs mátrixot a kamera kalibrációja során kapjuk meg [8]. Ezután a paletta és a robot bázisa közötti transzformáció segítségével,

ami a nullpontfelvételtől ismert, számítható ki a pozíció a robot koordináta-rendszerében. A szükséges kamera paraméterek meghatározását és a paletta nullpont felvételét egy részben automatizált program előre elvégzi és egy fájlban eltárolja az eredményeket, mindkét palettára vonatkozóan. A jelenlegi program már csak futásidőben olvassa be a transzformációs mátrixot. Így a pakolási folyamat nem függ a paletták helyzetétől, ha valamelyik új helyre kerül, csupán a hozzá tartozó frissített fájlból kell behívni az új transzformációs mátrixot.

Most, hogy ismert a darab valós helyzete és szögelfordulása, a következő lépés a megfogás megtervezése. Mivel – ebben az első változatban - nem volt szükség akadályok kikerülésére, vagy a darabok készülékbe illesztésére, ezért a legegyszerűbb megoldást, a függőleges irányú megfogást használjuk. Az építőelemek orientációját vizsgálva a hengerrel van a legkönnyebb dolgunk, mivel ez forgásszimmetrikus így tetszőleges Z körüli elfordulással felvehető. A kocka esetében szintén van lehetőség egyszerűsítésre, mivel 90 fokként az is szimmetrikus, itt mindig a legkisebb elfordulással járó orientációt kerül kiválasztásra. A háromszög alapú hasábot egy oldala és egy éle mentén fogja meg. Itt nincs lehetőség egyszerűsíteni, mindig a számított elfordulásnak megfelelően kell beállítanunk a megfogót.

4. TERV GENERÁLÁSA

Az összeszerelési (pakolási) sorrendterv generálható egy tetszőleges modell alapján, de tesztelési célokból természetesen lehetőség van közvetlenül (egy leíró fájlban) megadni. A sorrendiség leírására egy gráf szolgál melyben minden elemhez tartozik egy egyedi azonosító és egy tömb, ami azon építőelemek azonosítóit tárolja, melyeket le kell tenni, mielőtt az adott elem elhelyezhető. Itt elegendő a közvetlenül megelőző elemek azonosítóit tárolni, nem szükséges az összes korábbi elem. Ennek előnye, hogy megenged olyan helyzeteket amikor több lehetőség közül futásidőben választhatunk, hogy melyik kerüljön át hamarabb.

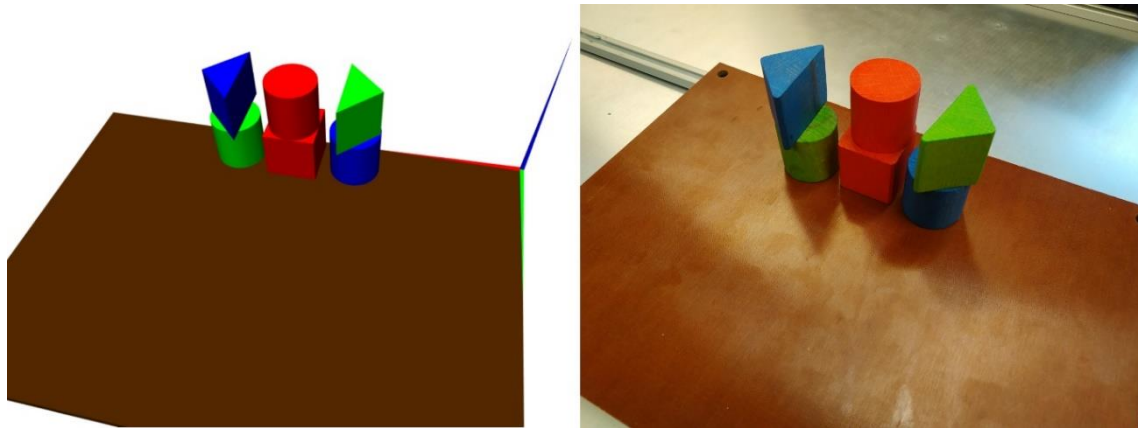
A másik lehetőség, sorrendterv generálás alapja egy X3D modell [9], amelyben a kívánt összeállítás formája írható le. A program beolvassa ezt a fájlt és összeveti a benne szereplő építőelemeket az általa ismertekkel. Az azonosított darabokhoz megkeresi a hozzájuk tartozó *Transform* elemeket és az ezekben adott transzformációkat egymás után alkalmazva kiszámítja a paletta bázisától mért koordinátákat. Ezzel előáll a pakoláshoz szükséges elemek listája és az elemek letételi pozíciói.

A pakolási sorrendtervhez az adott elem lerakásához szükséges elemeket kell kikeresni a listából minden elemhez. Ehhez az építőelemek gömbi környezetét ellenőrzi. Ha két elem egymás környezetében van, akkor első körben a magasság alapján dönt, a lejjebb kerülő elemnek kell korábban menni. Egyenlőség esetén a fájlban korábban definiált építőelem kerül a helyére előbb. Ilyen esetben azonban még azt is vizsgálni kell, hogy a két elem, az előírt orientációban lerakható-e egymás mellé. A második elem lerakásánál, nem ütközik-e bele a megfogó az elsőbe. Ha esetleg igen, megpróbálja a rendszer kihasználni a megfogásnál tárgyalt szimmetriákat és egy egyenértékű elforgatott pozícióban letenni az elemet. Ezt a korlátozást a megfogó miatt szükséges előírni, mivel annak is van szélessége, amelyet figyelembe kell venni.

A bemutatott módszer a sorrendterv automatikus előállításának egy meglehetősen egyszerű módja. Jelen helyzetben azonban a munkadarabok és összeállítások egyszerűsége miatt ez is megfelelő eredményeket ad. Érdemes megjegyezni, hogy vannak módszerek ennek a problémának a megoldására jóval összetettebb esetekben is [10].

5. GRAFIKUS FELHASZNÁLÓI FELÜLET

Mivel a dolgozatban bemutatott rendszer demonstrációs célra készül, mindenképpen szükséges valamilyen felhasználóbarát, interaktív felület biztosítása a működtetéshez. Ezért egy korábban kifejlesztett szerelés tervező modult [11] került a rendszerbe, ami egy webes interface, amely az X3DOM [12] segítségével jeleníti meg a háromdimenziós tartalmat. Ezen felületen megjelenik a teljes cella modellje a robottal és a palettákkal. Valamint a képkészítés után a felismert alkatrészek modellje is elhelyezésére kerül a megfelelő palettán. Ezután van lehetőség a felhasználónak az építőelemeket egérrel áthelyezni a másik palettára, így létrehozva a kívánt „épületet”. A kész modell ezután dolgozza fel a bemutatott program. Pakolás közben a virtuális robot együtt mozog a valós robottal. A felhasználói felület biztosítja, hogy amikor több építőelem is átkerülhet egyszerre, akkor a kiválasztható a következő elem. A virtuális és a valós építményeket a 2. ábra mutatja.



2. ábra A virtuálisan (balra) és a valójában megépített összeállítás (jobbra)

6. ÖSSZEGZÉS

A bemutatott robot rendszer teljesíti a kiberfizikai gyártórendszerekkel szemben támasztott elvárásokat. Képes érzékelni a pakolási környezetben fellépő változásokat és célja elérése érdekében ennek megfelelően beavatkozni. Az implementált rendszer felépítése lehetőséget biztosít a későbbiekben a modulok önálló továbbfejlesztésére, hogy összetettebb problémák megoldására is alkalmassá váljon. A tesztek alapján a rendszer gyenge pontja az alkatrészek helyzetének kellően pontos meghatározása, aminek oka a megvilágításra érzékeny színszűrők alkalmazása. Ezt érdemes lenne kiváltani egy rugalmasabb megoldással. Tervezett fejlesztés még emellett, egy tapadókorongos megfogó integrálása a rendszerbe, az ehhez szükséges megfogási és tervezési algoritmusokkal.

KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutatás az „Ipar 4.0 kutatási és innovációs kiválósági központ” projekt (GINOP-2.3.2-15-2016-00002) keretében valósult meg. Részben az Európai Unió H2020-as EPIC (www.centre-epic.eu/) projektje (No. 739592) támogatta.

IRODALOM

- [1] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda.: Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2):621-641, 2016
- [2] G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, G. Chryssolouris: Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach, *CIRP 2(2)*, 2010, pages 81-91
- [3] N. Kousia, C. Gkournelosa, S. Aivaliotisa, C. Giannoulisa, G. Michalosa, S. Makrisa: Digital twin for adaptation of robots' behavior in flexible robotic assembly lines, *Procedia Manufacturing*, Volume 28, 2019, pages 121-126
- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi., You only look once: Unified, real-time object detection., *IEEE Conference on Computer Vision and Pattern Recognition 2016*, pages 779-788
- [5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15*, pages 91-99, Cambridge, MA, USA, 2015. MIT Press.
- [6] Joseph Redmon and Ali Farhadi.: YoloV3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
- [7] D. Horváth: Advanced Robotic System Enhanced with Computer Vision, Diplomamunka, Budapesti Műszaki és Gazdaságtudományi Egyetem, Automatizálási és Alkalmazott Informatikai Tanszék
- [8] M. Hajós: Automatizált robot nullpontfelvétel kamerával és erőszennorral, Szakdolgozat, Budapesti Műszaki és Gazdaságtudományi Egyetem, Gyártástudomány és -technológia Tanszék, 2019
- [9] X3D, <https://www.web3d.org/x3d/what-x3d>
- [10] Cs. Kardos, A. Kovács, J. Váncza: Decomposition approach to optimal feature-based assembly planning. *CIRP Annals - Manufacturing Technology*, 66(1), 2017
- [11] B. Bálint: Interaktív szerelés tervező rendszer fejlesztése robot cellához, TDK dolgozat, Budapesti Műszaki és Gazdaságtudományi Egyetem, Gyártástudomány és -technológia Tanszék, 2019
- [12] X3DOM, <https://www.x3dom.org>