Reconstrucció d'objectes 3D a partir d'imatges capturades

Oriol Rovira Vacas, Sílvia Sanvicente García

Abstract— La reconstrucció d'objectes 3D és un problema molt explorat en el camp de la visió per computador. Ser capaç de generar un objecte tridimensional és una funcionalitat bàsica per a l'anàlisi del món real amb multitud d'aplicacions possibles. En aquest treball s'han creat diversos datasets a partir de models 3D i s'han obtingut els paràmetres de la càmera amb Stereo Camera Calibrator. Juntament amb diversos datasets de The Middlebury Computer Vision Pages s'ha implementat l'algoritme Visual Hull per diferents nombres d'imatges i s'ha analitzat el seu rendiment mitjançant tan tècniques heurístiques com feature matching.

Keywords— 3D Object Reconstruction, Feature Matching, FPFH, KNN, PointCloud, Space Carving, STL, Visual Hull

1 Introducció

L'objectiu de la reconstrucció d'objectes 3D és inferir la geometria i l'estructura d'objectes existents en una escena a partir d'una o més imatges 2D. Aquesta funcionalitat és fonamental en moltes aplicacions com ara l'animació i el modelatge, la navegació de robots, manipuladors, impressió 3D, en el camp de la medicina per a poder diagnosticar millor als pacients, i més generalment en els camps de l'animació i modelatge orientat a continguts audiovisuals.

Al llarg dels anys, les tècniques utilitzades per a la reconstrucció d'objectes han evolucionat considerablement. Inicialment es va plantejar el problema des d'un punt de vista geomètric, intentant formalitzar matemàticament la projecció de 2D a 3D amb l'objectiu d'arribar a algorismes capaços de resoldre el problema. Actualment s'utilitza l'aprenentatge màquina per a resoldre aquest problema, entrenant models en les relacions entre 2D i 3D inclús amb només una sola imatge de l'objecte [1] [2].

Al llarg d'aquest article detallarem el procés emprat al llarg del desenvolupament d'un algoritme de reconstrucció d'objectes 3D a partir de múltiples imatges. Aquest algoritme està pensat per ser utilitzat pel robot que estem desenvolupant en paral·lel a l'assignatura de RLP, el qual necessita ser capaç de reconèixer objectes 3D del món real per tal de construir-los ell mateix utilitzant peces de LEGO. Degut al format virtual de RLP, serem capaços de generar datasets propis de manera ràpida i eficaç al propi entorn de simulació de RLP, tot i així, utilitzarem també datasets públics de models 3D per tal de facilitar el procés i tindre més mostres.

2 Estat de l'art

La reconstrucció 3D a partir d'una o diverses imatges RGB és un problema que es planteja des de fa anys a les branques de visió per computador, gràfics per computadors i l'aprenentatge computacional. Aquesta tècnica es pot aplicar a diversos camps, com els comentats prèviament o en altres com són la impressió 3D, la conducció de cotxes autònoms, la realitat virtual o la realitat augmentada.

Les tècniques per reconstruir un objecte en 3D es divideixen en dos mètodes, actius i passius. En el cas dels mètodes actius s'utilitza la informació que s'obté de projectar llum sobre l'objecte per realitzar la reconstrucció i en els mètodes passius s'analitzen dues o més imatges per obtenir aquesta informació [3][4]. Dins dels mètodes passius, podem analitzar les imatges a partir de la forma que tenen els objectes o podem aplicar stereo vision.

A l'hora d'analitzar la forma dels objectes en les imatges podem obtenir la informació necessària per fer la reconstrucció 3D de diverses maneres, ja sigui analitzant el moviment, la il·luminació, el desenfocament o les siluetes. D'aquests mètodes, un dels més utilitzats és l'anàlisi de les siluetes per obtenir el volum d'un objecte des de diferents punts de vista [5]. Aquesta tècnica consisteix a separar l'objecte del fons creant una imatge binària amb la silueta de l'objecte. Juntament amb els paràmetres de visualització de la càmera, la silueta es projecta formant un con. A partir de la intersecció de diversos cons obtinguts de diferents punts de vista podem reconstruir el volum de l'objecte [6].

També podem obtenir el 3D d'un objecte estudiant la disparitat de les imatges del dataset. Inicialment aquestes reconstruccions es realitzaven comparant només dues imatges amb la tècnica stero vision, però posteriorment s'ha evolucionat a la tècnica multiview stereo, utilitzant múltiples imatges [7]. Aquesta tècnica compara les imatges obtenint informació sobre la profunditat i crea un mapa de disparitat. Els valors d'aquest mapa de disparitat són inversament proporcionals a la profunditat de l'escena a la ubicació del píxel corresponent.

Donat que aquestes tècniques no donàvem resultats òptims, a partir de 2015, es van desenvolupar noves tècniques agafant el coneixement que es tenia fins al moment i aplicant tècniques d'aprenentatge màquina utilitzant grans quantitats de dades i xarxes neuronals convolucionals. Aquesta nova implementació ha permès utilitzar múltiples imatges RGB evitant la complexitat de disposar d'un calibratge perfecte de la càmera o fins i tot realitzar reconstruccions 3D amb una única imatge d'entrada amb xarxes com OccNet o Pixel2Mesh [8] [9].

3 Proposta

En ser un algoritme orientat a ser utilitzat per un robot, hem de tindre en compte les seves característiques a l'hora de construir una solució.

El robot utilitza una Raspberry Pi v4 per a funcionar, amb una càmera NoiR v2 la qual pot moure lliurement per l'entorn de treball.

La Raspberry Pi és un ordinador en miniatura construït en una sola placa, es compacte i barat, cosa que juntament amb la seva relativa potència el fa el candidat perfecte per a ser utilitzat en infinitat d'aplicacions de sistemes encastats.

En el nostre cas, tot i la seva potència, el fet d'estar treballant amb imatges i models 3D, creiem oportú evitar utilitzar implementacions basades en l'aprenentatge màquina per evitar sobrecarregar el sistema, ja que aquest no serà l'única funcionalitat que estarà executant en el seu cicle d'execució [10].

La càmera NoiR v2 és capaç de gravar vídeos en diferents

resolucions i fps, amb una resolució màxima de 1080x1920 i 30fps. La càmera també és capaç de capturar imatges estàtiques en una resolució de fins a 3280x2464. Això ens permet obtenir fotografies molt detallades sobre els objectes, potencialment permetent-nos obtenir models molt detallats. Per altra banda, en cas que la resolució suposi un problema de rendiment, sempre podem jugar amb la resolució (sigui directament a la càmera o en un procés de preprocessat) per tal de reduir l'esforç de l'algoritme a canvi de resultats menys precisos.

3.1 Datasets

Per avaluar el sistema utilitzarem quatre datasets, dos de tercers i dos propis. Els datasets de tercers estan extrets de la pàgina The Middlebury Computer Vision Pages [11], la qual facilita diversos datasets per avaluar l'exactitud i la completesa de reconstruccions 3D aplicant l'algoritme de multiview stereo. S'utilitzaran els datasets Temple i Dino els quals estan organitzats en tres subconjunts amb diferents vistes per poder avaluar els resultats amb diferents nombres d'imatges. Aquests subconjunts estan formats per 16, 48 i 312 imatges respectivament.

Per tal de poder establir el rendiment de l'algoritme Visual Hull, executarem els 2 datasets de Middlebury en les seves tres configuracions diferents d'imatges, amb l'objectiu de poder quantificar la relació entre nombre d'imatges d'input i qualitat del model resultant.

Al no existir groundtruth per a aquests datasets de Middlebury (hem sol·licitat a l'autor el fitxer ply que utilitza i no l'ha volgut compartir) utilitzarem els resultats del subset de major imatges com a groundtruth dels altres subsets de cada model. Tot i que no és un mètode perfectament exacte, les condicions sobre les quals estem estudiant l'aplicació d'aquest algorisme ja requereix una implementació on s'utilitzin poques imatges, tant per el propi temps que es trigaria a capturar tal quantitat d'imatges com el conseqüent cost computacional de càlculs tan grans.

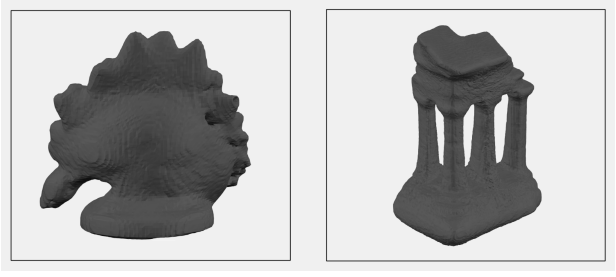

Fig. 1. Groundtruth del model 3D dels datasets Dino (esquerra) i Temple (dreta).

Per crear els datasets propits s'han escollit dos models 3D, RubberDuck i Mazinger Z, de la pàgina web GrabCAD en format .dm [12]. També s'ha utilitzat un model d'un tauler d'escacs per a realitzar la calibració i poder obtenir els paràmetres de la càmera. Mitjançant el programari Autodesk Fusion 360, s'han fixat vuit vistes i s'han renderitzat els models en format .png. Amb l'aplicació Stereo Camera Calibrator de MATLAB hem obtingut els paràmetres de la càmera gràcies al tauler d'escacs [13].

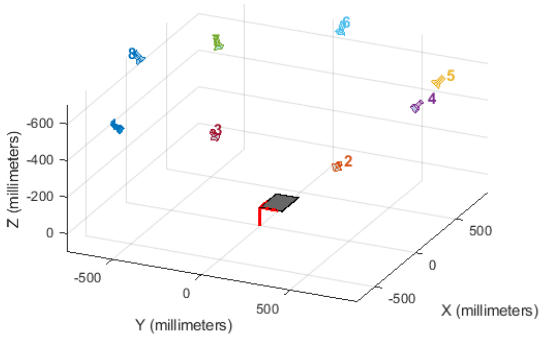

Fig. 2. Imatge de les posicions de les càmeres obtingudes amb el tauler d'escacs a l'aplicació Stereo Camera Calibrator de MATLAB.

3.3 Tècniques emprades

Per realitzar la reconstrucció 3D dels objectes a partir de les imatges de les diferents vistes s'utilitzarà l'algoritme Visual Hull, el qual es basa en la forma de la silueta de l'objecte [14]. En primer lloc, cal separar l'objecte del fons i construir una imatge binària on el fons estigui constituït per píxels de color negre (valor 0) i la silueta de l'objecte i el seu interior siguin píxels de color blanc (valor 1).

Per reconstruir a partir de les siluetes, aquestes s'han de projectar, mitjançant els paràmetres de la càmera, com si sortissin del centre òptic i abastessin l'espai 3D generant un con. Aquests cons no tenen com a base un cercle sinó la silueta de la vista projectada. En lloc de projectar cap enrere cada silueta a l'espai 3D, tots els vòxels (píxels 3D) de la quadrícula de vòxels es projecten en cada silueta. Un cop realitzada aquesta projecció es converteixen els vòxels (x, y, z) a valors 2D (x, y) i es comprova si el píxel (x, y) pertany a la silueta. Finalment s'obté el volum de l'objecte com la intersecció de tots els cons generats pels diferents punts de vista.

En un entorn purament físic, la validació d'aquest algoritme seria relativament feixuga, ja que s'hauria de disposar de tant el model com de l'objecte físicament idèntic. Per sort el robot existeix en un entorn simulat al CoppeliaSim, per tant només hem de disposar de l'objecte virtualment per tal de fer el test i la validació.

4 Experiments, resultats i anàlisi

Malauradament, tot i els esforços en crear múltiples datasets casolans, i tot i haver aconseguit crear paràmetres de càmera artificials que siguin correctes, la implementació de la utilització d'aquests datasets ha resultat impossible.

Tot i que podem arribar a generar algunes projeccions, el model resultant no correspon a l'output esperat. Hem adaptat el codi a les necessitats dels nostres datasets, però tot i així no ha estat suficient. Respecte al VisualHull creiem que es tracta d'un error per part de l'autor, que en comptes de crear una meshgrid dinàmica en funció de l'input, ha hardcodejat els valors de les bounding boxes de tal manera que si les escalem a unes mides prou grans com per poder projectar al nostre model, a causa del pèssim control de la memòria i la multitud de còpies de matrius

de múltiples dimensions que genera, el programa ens requeriria de 200 GB de memòria RAM per a poder finalitzar l'execució. Tot i així, hem provat d'assignar prou memòria virtual mitjançant un SSD per a intentar-lo córrer igualment, i tot i que si bé és cert que l'output millora i és capaç de situar una part dels vòxels a sobre de les siluetes, aquests són massa poc densos i no resulten en cap model coherent (tenint en compte que l'execució requereix hores de funcionament fins a la finalització).

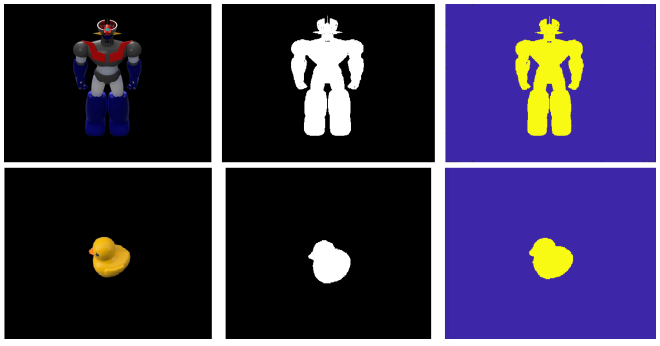

Fig. 3. Datasets propis a partir dels models 3D RubberDuck i Mazinger Z.

4.1 Mesures de rendiment

Un vòxel és la unitat més petita d'espai 3D que es pot crear, equivalent a un píxel en 3D. Cada vòxel està associat a un conjunt de facets, que són estructures geomètriques normalment triangulars que utilitzem per representar l'objecte en format STL. Per quantificar els diferents models que hem obtingut dels dos datasets, hem analitzat el nombre de fasets obtinguts en funció del nombre d'imatges de cada subset.

	16 imatges	48 imatges	312 imatges
facets Dino	189240	178537	144552
facets Temple	70224	58166	54264

Taula 1. Nombre de facets depenent del nombre d'imatges.

Com bé es pot observar en la Taula 1, el nombre de facets disminueix quan utilitzem més imatges. Per aquests models en concrets, aquesta disminució de facets és conseqüència del fet que en tenir més imatges, obtenim models més complets, amb menys protuberàncies i forats i conseqüentment els resultats són més llisos. A la Figura 4, podem veure les dades de la Taula 1 representades gràficament.

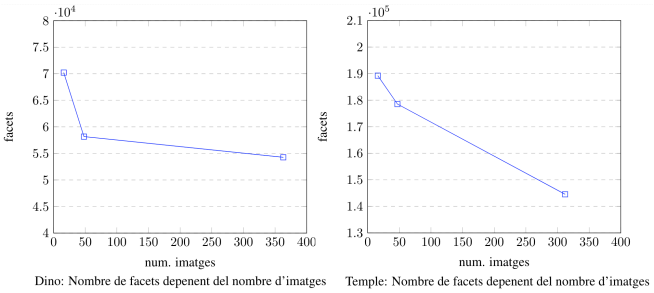

Fig. 4. Gràfica dels facets depenent del nombre d'imatges.

Per poder visualitzar les diferències entre models de manera més directe, hem superposat els point clouds de les figures. En lila es pot veure el groundtruth i en verd el model sent comparat.

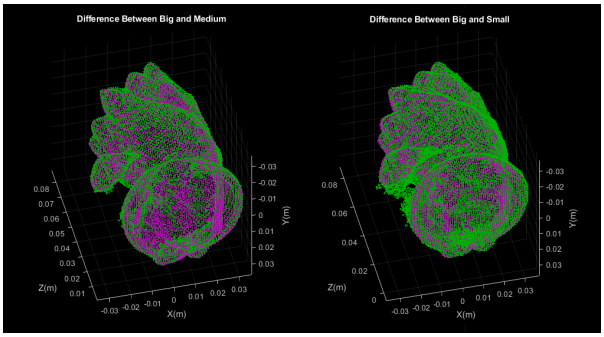

Fig. 5. Superposició models (verd) vs. groundtruth (lila)

Tot i la utilitat per a nosaltres és alta, i una persona seria fàcilment capaç de determinar quin model és millor, a nivell matemàtic encara no proporciona cap informació quantificable.

Utilitzant aquesta mateixa representació en point cloud, podem calcular les diferències entre groundtruth i resultat, simplement comparant les distancies entre punts del model.

Inicialment vam veure que, a causa de les diferències en nombre de facetes (i per conseqüència, de punts en els point clouds) realitzar una simple comparació no era suficient, ja que el nombre de punts variable fa que les figures més denses tinguin punts molt més propers que les figures amb menys, fent que els resultats variïn en funció del punt de referència seleccionat.

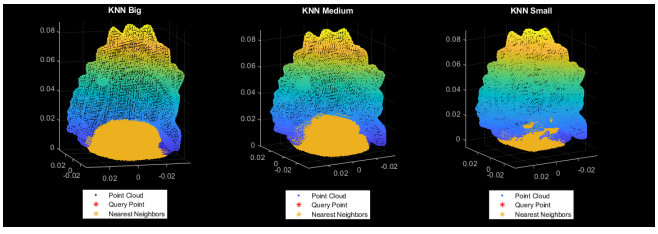

Fig. 6 KNN desde origen (0,0,0)

Per solucionar aquest problema i obtenir els resultats més precisos possibles utilitzem l'algorisme de KNN com a algorisme de feature matching. Al qual li passem les features detectades al groundtruth i al model analitzat mitjançant FPFH [15].

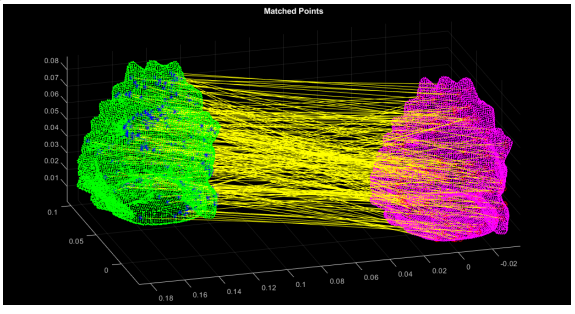

Fig. 7 Parelles resultants KNN aplicat a FPFH features

Així obtenim el nombre de features que s'han connectat i les

seves scores. Utilitzant aquests resultats ja podem quantificar de manera exacta el rendiment de cada escenari.

Amb aquests resultats, podem determinar que, a major nombre de fotos, major nombre de facetes i pitjor model resultant.

El nombre de facetes és particularment important per a l'aplicació objectiu de l'algorisme, ja que està lligada directament al cost computacional de treballar amb aquest model, tant per a la seva representació com en el seu hipotètic tractament posterior, un cop agafat el resultat per part del següent element de la pipeline interna del robot.

Cal mencionar també però el cost computacional inicial de la creació del model de més qualitat, tot i que la configuració estàndard amb la qual es presenten els resultats és relativament eficaç (9 segons), intentar augmentar la resolució del model (reduint mida de vòxels i augmentant en número) incrementa exponencialment tant en temps com en memòria requerida per a l'execució del programa.

La millora dels resultats amb l'increment de nombre de fotografies sembla indicar l'existència d'un cert "sweet spot" on la relació nombre de facets i exactitud del model, juntament amb cost computacional, arriben a valors màxims. Amb el triple de fotografies, entre els datasets de 16 i 48, la millora quant a facets és del 17,7%, i la millora quant a exactitud del model és del 93%. Mentre que augmentar el nombre de fotografies de 48 a 363 només aporta una millora del 6,7%, juntament amb un augment del temps d'execució de 44 vegades l'anterior, i un consum de memòria difícilment justificable per a ser executant en un ordinador de baixes prestacions.

5 Conclusions

Creiem que un estudi amb datasets més variables en quant a nombre d'imatges en el rang d'entre 48 i 360 seria ideal per a ser capaços de localitzar el "sweet spot" de qualitat vs rendiment. Creiem que per a poder realitzar un estudi més profund en aquest camp és necessari disposar de la capacitat de crear datasets personalitzats. Els requeriments de cada dataset varien molt entre implementacions, cosa que porta a que els datasets existents estiguin també enfocats a alguna aplicació concreta i siguin difícilment adaptables al cas concret en el qual es treballa.

Després de la realització d'aquest informe hem vist que la implementació d'algorismes de reconstrucció 3D és més difícil d'implementar del que crèiem. La quantitat de recursos online és relativament petita, i les implementacions estan fetes molt específicament per resoldre certs problemes de reconstrucció concrets.

Un problema recurrent al llarg de les hores treballades ha estat la incompatibilitat constant entre diferents implementacions i datasets, igual que la poca flexibilitat del codi utilitzat per part dels pocs recursos que existeixen.

Hem pogut comprovar de primera mà les mancances de l'algorisme VisualHull, i tot i que ara creiem que la reconstrucció d'objectes 3D mitjançant siluetes deixa molt a desitjar respecte a competidors on s'utilitzen tècniques capaces de detectar profunditat i sobrepassar aquestes mancances, seguim creient que VisualHull (o alguna de les variants) seria el candidat perfecte per a ser utilitzat pel robot, si bé seria necessària una major optimització del codi (possiblement implementar l'algorisme en un llenguatge més optimitzable) per a què la Raspberry Pi del robot no resultes sobrecarregada.

Bibliografia

[1] Han, X., Laga, H., & Bennamoun, M. (2019). Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. IEEE transactions on pattern analysis and machine intelligence.

[2] Salvi, A., Gavenski, N., Pooch, E., Tasoniero, F., & Barros, R. (2020, July). Attention-based 3D Object Reconstruction from a Single Image. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[3] Herbort, S., & Wöhler, C. (2011). An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods. 3D Research, 2(3), 1-17.

[4] Saaidi, A., & Satori, K. (2014, May). Multi-view passive 3D reconstruction: comparison and evaluation of three techniques and a new method for 3D object reconstruction. In 2014 International Conference on Next Generation Networks and Services (NGNS) (pp. 194-201). IEEE.

[5] Yemez, Y., & Schmitt, F. (2004). 3D reconstruction of real objects with high resolution shape and texture. Image and Vision computing, 22(13), 1137-1153.

[6] Anke, B., Olaf, H., Volker, R., & Ulas, Y. (2008). A Benchmark dataset for performance evaluation of shape-from-X algorithms. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 16, 26.

[7] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006, June). A comparison and evaluation of multi-view stereo reconstruction algorithms. In 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06) (Vol. 1, pp. 519-528). IEEE.

[8] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., & Jiang, Y. G. (2018). Pixel2mesh: Generating 3d mesh models from single rgb images. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 52-67).

[9] Xu, Q., Wang, W., Ceylan, D., Mech, R., & Neumann, U. (2019). Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. arXiv preprint arXiv:1905.10711.

[10] Zhang, Y., Gibson, G. M., Hay, R., Bowman, R. W., Padgett, M. J., & Edgar, M. P. (2015). A fast 3D reconstruction system with a low-cost camera accessory. Scientific reports, 5(1), 1-7.

[11] Seitz, S. (2021). Multi-view stereo evaluation web page. http://vision.middlebury.edu/mview/.

[12] GrabCAD. (2021). Design Community, CAD Library, 3D Printing Software Website. https://grabcad.com/.

[13] Calibrator, C., & Calibrator, S. (2021). Stereo Camera Calibrator App - MATLAB & Simulink - MathWorks. https://es.mathworks.com/help/vision/ug/stereo-camera-calibrator-app.html.

[14] Schneider, D. C. (2014). Visual Hull.

[15] Rusu, R. B., Blodow, N., & Beetz, M. (2009, May). Fast point feature histograms (FPFH) for 3D registration. In 2009 IEEE international conference on robotics and automation (pp. 3212-3217). IEEE.

Enllaç per descarregar el codi utilitzat