مدل‌سازی صفحه‌ای محیط‌های داخلی با استفاده از تصاویر RGB-D

مقداد پاک‌نژاد و مهدی رضائیان*
گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران

چکیده

در رباتیک و به‌طور خاص برای ساخت نقشه‌های سه‌بعدی از محیط‌های داخلی، تفسیر تصاویر RGB-D به مسئله مهمی تبدیل شده است. در این مقاله جهت کاهش حجم داده‌ها و تسریع ساخت نقشه سه‌بعدی، تصاویر عمق به ابرهای نقطه‌ای تبدیل و سپس آن‌ها بر مبنای صفحات تصویر قطعه‌بندی می‌شوند. پس از برازش مدل صفحه‌ای متناظر با هر قطعه، تعداد مشخصی از نقاط روی صفحات تولید و سپس با اجرای الگوریتم تکراری نزدیک‌ترین نقطه (ICP) روی این نقاط، ماتریس‌های دوران و انتقال بین هر دو فریم تخمین زده شده و تصویر تثبیت می‌شود. نتایج نشان می‌دهد که روش ارائه‌شده، به‌طور متوسط سرعت را درصورت استفاده از فریم‌های متوالی ۵۵ درصد و درصورت استفاده از فریم‌های غیرمتوالی ۹۱ درصد افزایش می‌دهد. روش پیشنهادی می‌تواند منجر به کاهش حجم محاسبات در مسئله مکان‌یابی و تهیه نقشه هم‌زمان (SLAM) شود.

واژگان کلیدی: مسئله تهیه نقشه، تصاویر RGB-D، حس‌گر کینکت.

Indoor Planar Modeling Using RGB-D Images

Meghdad Paknezhad & Mehdi Rezaeian*
Dept. of Computer Engineering, Yazd University,Yazd, Iran

Abstract

In robotic applications and especially 3D map generation of indoor environments, analyzing RGB-D images have become a key problem. The mapping problem is one of the most important problems in creating autonomous mobile robots. Autonomous mobile robots are used in mine excavation, rescue missions in collapsed buildings and even planets' exploration. Furthermore, indoor mapping is beneficial in finding and rescuing missions. With recent advances, mobile robots are used in hazardous missions such as radioactive areas or collapsing buildings. Having the environment's map beforehand can boost efficiency and effectiveness of the mission. In order to digitize the environment, several 3D scans are needed. However, these scans should be merged according to a global coordination system to create a correct, consistent model. This process is called image registration. If the robot with 3D scanner is able to accurately localize itself, the registration can be done directly by robots pose. However, due to imprecise robot sensors, self-localization is error prone. Therefore, the geometric structure of overlapping 3D scans is considered. In order to registering various points sets, Iterative Closest Point (ICP) algorithm is used. ICP is the most common approach to align point clouds in two consecutive image frames. This algorithm uses a point to point approach. RGB and depth images which are captured by Kinect are used in this study. In order to reducing data points and performing faster 3D map creation, depth images are converted to point clouds and then segmentation is done according to image planes. For this purpose RGB images are segmented by region growing segmentation algorithm. In this algorithm, the image was initially over segmented. This algorithm uses stack data structure and Euclidean distance in Lab color space to segment the image. Euclidean distance in Lab color space describes the resemblance of two colors to each other. In this algorithm, the aim is to label each pixel to a segment. To this end, each unlabeled pixels Euclidean distance to its neighboring mean color is checked to be within a threshold. For over-segmentation, if the distance satisfies the smaller threshold, the more pixels will be merged to the segment. Afterwards a plane was fit to each segment. After segmentation, each segment should be represented by a plane. Eventually, the segments were merged based on the product of normal vectors and plane fitting error criteria. After segmentation, planes were fit to the new segments again. A given number of points were generated on the plane. ICP algorithm was executed on these points and transfer and rotation

* نویسندهٔ عهده‌دار مکاتبات
* Corresponding author

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

matrices were obtained. Generating points on the plane results in fewer points. Therefore, the points were reduced and algorithms performance was increased. The results show that the proposed method increases the speed up to 55 and 91 percent in consecutive and non-consecutive frames on average, respectively.

Keywords: **Mapping Problem, RGB-D Images, Kinect sensor.**

۱- مقدمه

تهیه نقشه رباتیک به مسئله به‌دست‌آوردن مدل مکانی از محیط فیزیکی از طریق ربات‌های متحرک گفته می‌شود. به‌طورمعمول نقشه‌ها برای ناوبری ربات (به‌عنوان‌مثال، مکان‌یابی) استفاده می‌شوند. برای به‌دست‌آوردن نقشه، ربات باید دارای حس‌گری که آن را قادر به درک جهان خارج می‌کند، باشد. به‌طورمعمول حس‌گرهای دوربین، رادار، لمسی، قطب‌نما، سامانه موقعیت‌یابی سراسری[1] (GPS) و محدوده یاب سونار، لیزر و مادون‌قرمز برای تهیه نقشه استفاده می‌شوند؛ بااین‌حال، تمام این حس‌گرها در معرض خطا (نوفهٔ اندازه‌گیری)، هستند [1].

ربات متحرک برای حرکت در محیط ناشناخته، نیاز به ساخت نقشه از محیط و مکان‌یابی خود در نقشه به‌صورت هم‌زمان دارد که به آن مسئله مکان‌یابی و تهیه نقشه هم‌زمان[2] (SLAM) گفته می‌شود. در محیط‌های فضای باز، این مسئله توسط GPS با دقت خوبی قابل‌حل است؛ اما در محیط‌های داخلی و یا در مکان‌هایی که داده‌های GPS در دسترس یا به‌اندازه کافی قابل اعتماد نیستند، تخمین موقعیت ربات سخت می‌شود و می‌توان راه‌حل‌های دیگری جایگزین شوند [2].

به‌طورکلی مسئله تهیه نقشه به‌عنوان یکی از مهم‌ترین مسائل در پیگیری ساخت ربات‌های متحرک خودمختار در نظر گرفته می‌شود. ساخت ربات‌های خودمختار در کاوش‌های داخل معادن و ساختمان‌های دچار حادثه‌شده و حتی کاوش بر روی سطح سیارات، مورد استفاده قرار می‌گیرند. وظیفه تهیه نقشه سه‌بعدی از محیط‌های داخلی در مأموریت‌های جستجو و نجات می‌تواند مفید باشد. در چند سال گذشته، ربات‌های جستجو و نجات توسعه یافته‌اند و در مأموریت‌های در مناطق خطرناک برای انسان مانند داخل ساختمان نزدیک به فروپاشی و یا محیط‌های دارای آلودگی رادیواکتیو استفاده می‌شوند. دردسترس‌بودن یک نقشه از این مناطق به‌طور چشم‌گیری بهره‌وری و اثربخشی این عملیات را می‌تواند بهبود دهد [3].

استخراج عمق از داده‌های دوربین به‌تنهایی به‌ویژه در محیط‌های داخلی با مناطق تاریک یا کم‌بافت دشوار است. دوربین‌های RGB-D نوعی سامانه حسی هستند که تصاویر RGB را همراه با اطلاعات عمق هر پیکسل دریافت می‌کنند. میکروسافت‌کینکت نوعی حس‌گر RGB-D است که برای ایجاد نقشه‌های سه‌بعدی از محیط‌های داخلی مناسب است؛ زیرا می‌تواند تصاویر RGB را با دقت ۶۴۰ ×۴۸۰ و عمق را با ۳۰ فریم در ثانیه به‌دست آورد [4].

به‌تازگی چند مجموعه داده[3] کینکت معرفی شده‌اند. بیش‌تر مجموعه داده‌های مورد استفاده در قطعه‌بندی معنایی تصویر، اشیا را در مرکز تصاویر و تحت شرایط نور مناسب قرار می‌دهند؛ درحالی‌که نسخه دوم مجموعه داده عمق NYU، شامل صحنه‌های ادارات، مغازه‌ها، اتاق‌های خانه‌هایی است که بسیاری از اشیا روشنایی یکنواخت دارند [5]. مجموعه داده NYU-v2، بزرگ‌تر و متنوع‌تر از مجموعه داده‌های موجود معرفی شده است. این مجموعه داده ۴۰۷،۰۲۴ تصویر RGB-D با وضوح ۶۴۰×۴۸۰، شامل طیف گسترده‌ای از ساختمان‌های تجاری و مسکونی در سه شهر مختلف ایالات‌متحده، است [6]. شکل‌های (۱و۲) به‌ترتیب تصویر RGB و تصویر عمق خروجی کینکت از یکی از مجموعه داده‌های NYU-v2 را نشان می‌دهند.

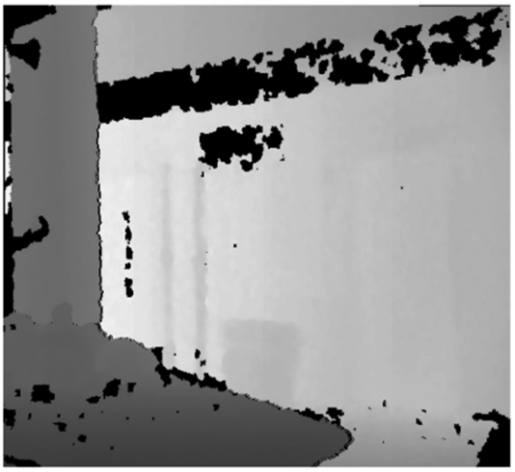


(شکل-۱): تصویر رنگی خروجی کینکت از مجموعه داده NYU
(Figure-1): Kinects color image. From NYU dataset.

[1] Global Positioning System
[2] Simultaneous Localization and Mapping
[3] Dataset

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]



(شکل-۲): تصویر عمق خروجی کینکت از مجموعه داده NYU

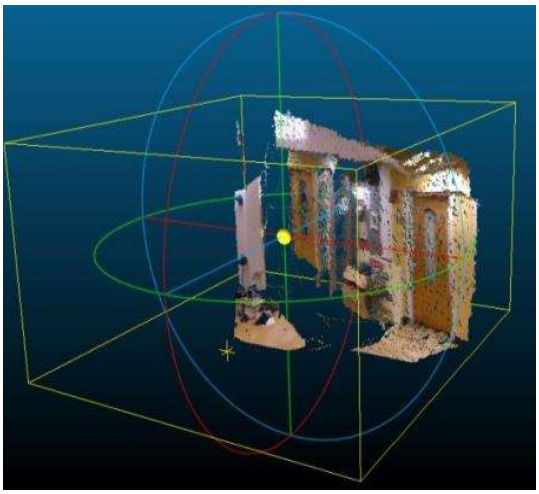
(Figure-2): Kinects depth image. From NYU dataset.

تصویر عمق گرفته‌شده توسط روابط (۱) و (۲) به ابرنقطه‌ای سه‌بعدی تبدیل می‌شود.

$$X = \frac{(u - u_0)Z}{f} \qquad (۱)$$

$$Y = \frac{(v - v_0)Z}{f} \qquad (۲)$$

در این روابط u و v سطر و ستون پیکسل و u_0، v_0 و f پارامترهایی هستند که توسط دوربین تنظیم می‌شوند و Z این نقاط همان مقدار عمق تصویر گرفته‌شده توسط حس‌گر کینکت است. شـکل (۳) ابر نقطه‌ای به‌دست‌آمده از تصویر عمق به‌همراه رنگ پیکسل‌ها در تصویر RGB را نشان می‌دهد.



(شکل-۳): نمایش همزمان ابر نقطه‌ای و رنگ پیکسل‌ها

(Figure-3): Simultaneous display of point cloud and RGB pixels.

برای دیجیتال‌کردن محیط‌ها به پویش‌های متعدد سه‌بعدی نیاز است. برای ایجاد مدلی درست و سازگار، باید پویش در یک سامانه مختصات هماهنگ، ادغام شوند. این فرآیند را تثبیت تصویر[1] می‌گویند. اگر مکان‌یابی ربات توسط پویش‌گر سه‌بعدی دقیق باشد، تثبیت به‌طور مستقیم توسط موقعیت ربات می‌تواند انجام شود. بااین‌حال، با توجه به این‌که حس‌گرهای ربات اغلب نادقیق هستند، مکان‌یابی ربات نادرست می‌شود. بنابراین ساختار هندسی پویش سه‌بعدی باهم تداخل دارند. از الگوریتم تکراری نزدیک‌ترین نقطه[2] (ICP) برای تثبیت مجموعه‌های نقاط استفاده می‌شود.

فرض کنید دو مجموعه نقطه سه‌بعدی مستقل $\hat{M}$ با ابعاد N_m و $\hat{D}$ با ابعاد N_d موجود باشد. در این الگوریتم تبدیل $(\boldsymbol{R}, \boldsymbol{t})$ طوری پیدا می‌شود که ماتریس دوران $\boldsymbol{R}$ و بردار انتقال $\boldsymbol{t}$ تابع هزینه رابطه (۳) را به کمینه رساند.

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} \omega_{i,j} \left\| \hat{\boldsymbol{m}}_i - (\boldsymbol{R}\hat{\boldsymbol{d}}_j + \boldsymbol{t}) \right\|^2 \qquad (۳)$$

در فضا اگر i امین نقطه $\hat{M}$ توصیفی مشابه با j امین نقطه $\hat{D}$ داشته باشد، $\omega_{i,j}$ یک و در غیر این صورت صفر در نظر گرفته می‌شود. در رابطه (۳)، $\hat{\boldsymbol{m}}_i$ نقطه i ام از مجموعه نقطه $\hat{M}$ و $\hat{\boldsymbol{d}}_j$ نقطه j ام از مجموعه نقاط $\hat{D}$ است. در این الگوریتم دو چیز باید محاسبه شود: نخست، نقاط متناظر و دوم، تبدیل $(\boldsymbol{R}, \boldsymbol{t})$ به‌طوری‌که بر اساس نقاط متناظر، $E(\boldsymbol{R}, \boldsymbol{t})$ کمینه شود.

الگوریتم ICP به‌صورت تکراری نقاط متناظر را محاسبه می‌کند. در هر گام تکرار، الگوریتم نزدیک‌ترین نقاط مانند نقاط متناظر را انتخاب و تبدیل $(\boldsymbol{R}, \boldsymbol{t})$ را برای به کمینه رساندن رابطه (۳) محاسبه می‌کند. بسل[3] و همکارانش در [7] ثابت کردند که این روش در یک کمینه خاتمه می‌یابد. تمام گام‌های الگوریتم ICP خطا را کاهش می‌دهند. به‌عبارت‌دیگر رابطه (۴) در تمام مراحل این الگوریتم برقرار است.

$$E_{i+1}(\boldsymbol{R}, \boldsymbol{t}) < E_i(\boldsymbol{R}, \boldsymbol{t}) \qquad (۴)$$

درنهایت، این الگوریتم رابطه (۴) را کمینه و تعداد نقاط متناظر را بیشینه می‌کند [8].

الگوریتم ICP از رویکرد نقطه به نقطه استفاده می‌کند و متداول‌ترین رویکرد برای ترازکردن ابرهای نقطه‌ای سه‌بعدی از دو تصویر عمق متوالی است؛ اما این الگوریتم

[1] Image Registration
[2] Iterative Closest Point
[3] Besl

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

به‌طورمعمول با مقداردهی اولیه بد پارامترهای تبدیل، منجر به بهینه محلی[1] می‌شود. همچنین صفحات بزرگ مانند سقف یا دیوارها در تصاویر عمق، به تناسب نقاط سه‌بعدی تبدیل را به‌دست می‌آورند [9]. در این مقاله جهت افزایش سرعت ICP، صفحاتی از ابرهای نقطه‌ای استخراج می‌شوند.

در این مقاله دو هدف دنبال می‌شود. نخست، با برازش[2] صفحات از ابرنقطه‌ای به‌دست‌آمده از هر قطعه، حجم داده‌ها کاهش و ساخت نقشه سه‌بعدی تسریع شده است. علاوه‌براین، در مرحله قطعه‌بندی، روشی بر اساس برازش صفحه پیشنهاد شده است که ابتدا تصویر به قطعات زیادی تقسیم و سپس ادغام می‌شوند. هم‌چنین، جهت استفادهٔ هم‌زمان از ویژگی‌های رنگ و فاصله، تصاویر RGB و عمق به الگوریتم‌های خوشه‌بندی داده می‌شوند تا تصویر، به قطعات مناسب‌تری تقسیم شود.

این مقاله در ۴ بخش به بررسی موضوع می‌پردازد. بخش دوم، کارهای انجام‌شده در مسئله تهیه نقشه را معرفی می‌کند. در بخش سوم روش پیشنهادی جهت افزایش سرعت تهیه مدل سه‌بعدی بیان می‌شود. در بخش چهارم به تجزیه‌وتحلیل خروجی‌های مدل پرداخته و در پایان در بخش پنجم خلاصه و نتیجه‌گیری ارائه می‌شود.

۲- کارهای انجام‌شده

پژوهش‌های تهیه نقشه، سابقه طولانی دارد. در دهه ۱۹۸۰ و اوایل دهه ۱۹۹۰، زمینه ساخت نقشه به‌طور عمده به رویکردهای متریک و توپولوژیک، تقسیم شد. نقشه متریک خواص هندسی محیط را می‌گیرد، درحالی‌که نقشه توپولوژیک اتصال مکان‌های مختلف را توصیف می‌کند. ارائه اولیه‌ای از رویکردهای متریک، الگوریتم ساخت نقشه مشبک اشغال‌شده[3] الفس[4] و مروک[5] است. الگوریتم ساخت نقشه متریک دیگری توسط چتلا[6] و لمند[7] ارائه شد که با استفاده از مجموعه‌ای از چندوجهی‌ها[8] به توصیف هندسی محیط می‌پردازد. نقشه توپولوژیک، محیط را به‌عنوان فهرستی از مکان‌های مهمی که از طریق کمان به هم متصل می‌شوند، نشان می‌دهد. به‌طورمعمول روی کمان، اطلاعات نحوه حرکت از یک مکان به مکان دیگر نوشته می‌شود. به‌هرحال، همیشه تفاوت رویکردهای متریک و توپولوژیک مبهم است. به‌طورتقریبی تمام بخش‌ها در رویکردهای توپولوژیک، به اطلاعات هندسی تکیه می‌کنند. درعمل، نقشه‌های ایجادشده از رویکرد متریک، نسبت به نقشه‌های ایجادشده از رویکردهای توپولوژیک بهتر است؛ اما قدرت تفکیک و هزینه محاسباتی بالاتری نیز دارند [1].

به‌طور خاص، هدف، ساخت نقشه با استفاده از هر دو داده عمق و رنگ است. برخی از پژوهش‌گران، مانند [4] به‌منظور ساخت نقشه بر اساس حس‌گر RGB-D در مسئله SLAM، روش‌های مبتنی بر گراف را به‌دلیل ثبات آن در درازمدت اتخاذ کرده‌اند. برخی از این روش‌ها با استفاده از ویژگی‌های بصری مانند تبدیل ویژگی مقیاس ثابت[9] (SIFT) و ویژگی‌های قدرتمند سرعت بالا[10] (SURF) داده‌های تصویر را برای به‌دست‌آوردن تبدیل اولیه[11] بین فریم‌های دوربین استخراج می‌کنند و سپس توسط الگوریتم ICP ابرهای نقطه‌ای اولیه را تطابق می‌دهند [4]. این روش‌ها تهیه نقشه سه‌بعدی متراکم را توسط میلیون‌ها نقطه از محیط ارائه می‌کنند. با توجه به هزینه محاسباتی بالا برای تطابق داده‌های زیاد حس‌گر، پیاده‌سازی آن‌ها در زمان بلادرنگ بدون کمک‌گرفتن از منابع محاسبات موازی مانند واحد پردازش‌گر ترسیمی[12] (GPU) سخت است. به همین دلیل، روش‌های مختلفی برای کاهش ابعاد داده‌ها توسط استخراج ویژگی‌های هندسی از داده‌های حس‌گر خام پیشنهاد شده است [10].

اگرچه ابرنقطه‌ای سه‌بعدی برای تطبیق[13] فریم به فریم و بازسازی سه‌بعدی مناسب است، آن‌ها اطلاعات ارزشمند موجود در تصاویر را نادیده می‌گیرند. از سوی دیگر، دوربین‌های رنگی اطلاعات بصری را ثبت می‌کنند. بنابراین بهتر است از هر دو داده تصویر و عمق کمک گرفت [4].

استفاده از صفحه به‌جای ابرنقطه‌ای خام، مزایای مختلفی ازجمله کاهش داده، تطبیق و اجرای سریع[14] دارد. بااین‌حال، برای مسئله مکان‌یابی و تهیه نقشه هم‌زمان

[1] Local Optimization
[2] Fit
[3] Occupancy Grid Mapping Algorithm
[4] Elfes
[5] Moravec
[6] Chatila
[7] Laumond
[8] Polyhedra

[9] Scale Invariant Feature Transform
[10] Speeded-Up Robust Features
[11] Initial Transformation
[12] Graphics Processing Unit
[13] Alignment
[14] Fast Rendering

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

به‌صورت برخط[1]، به استخراج سریع صفحات نیاز است. برخی از روش‌های پیشین استخراج صفحه مانند [11] از بردارهای نرمال برای قطعه‌بندی نقاط استفاده می‌کنند؛ اما به‌دلیل جستجوی نزدیک‌ترین همسایه و تجزیه ماتریس کوواریانس محلی، بار محاسباتی زیادی دارد. پاپینگا[2] و همکارانش در [12] روش ناحیه در حال رشد سریع، پیشنهاد دادند. در روش آن‌ها با هر بار اضافه‌شدن یک نقطه جدید به صفحه، مدل صفحه به‌روزرسانی می‌شود.

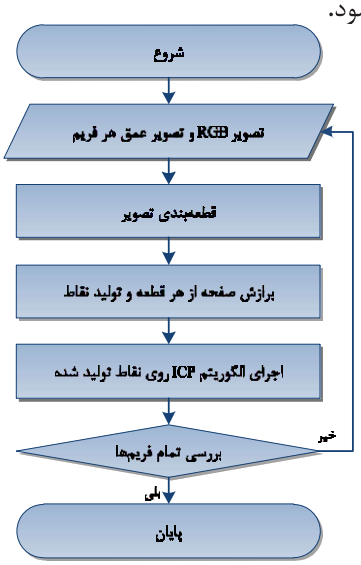
مسائل بازسازی سطح[3] و قطعه‌بندی بیش از دو دهه به‌خوبی مورد مطالعه قرار گرفته‌اند. مسئله بازسازی سطح برای تصویر واحد، تصاویر متعدد، داده‌های استریو، داده‌های لیزر، داده‌های رنگ و عمق تثبیت‌شده[4] و در همین اواخر داده‌های RGB-D مورد بررسی قرار گرفته است. از آن‌جایی‌که در بسیاری از کاربردهای دنیای واقعی از بازسازی سطح استفاده می‌شود، توجه مداوم پژوهش‌گران به این موضوع مهم جلب شده است. به‌تازگی، با معرفی حس‌گر کینکت کاربردهای متعددی در یک دوره زمانی به‌نسبه کوتاه پدید آمده است [13].

در برخی از کارها مانند [14] و [15] فرض می‌کنند دوربین و لیزر تراز شده‌اند و از نشانه‌های هندسی و ظاهری استفاده می‌کنند. استرم[5] و همکارانش در [14] الگوریتم قطعه‌بندی بر اساس گراف[6] را پیشنهاد دادند که رنگ و عمق گرفته‌شده توسط لیزر را ترکیب می‌کنند. اگرچه الگوریتم آن‌ها، نشانه‌های رنگ و عمق را با شیوه‌ای ساده ترکیب کردند، اما از روش‌های قبلی که فقط از داده لیزر یا رنگ استفاده می‌کنند، پایدارتر بوده و با توجه به پیچیدگی محاسباتی کم آن در کاربردهای بلادرنگ نیز قابل استفاده است. تیلر[7] و همکارانش در [15] الگوریتمی پیشنهاد کردند که ابرنقطه‌ای سه‌بعدی و تصویر رنگی مرتبط با آن، گرفته شده توسط آرایه‌ای از دوربین‌ها و پویش‌گر فاصله‌یاب هوکیو[8] نصب‌شده روی ربات انسان‌نمای PR2، را به‌عنوان ورودی می‌گیرد؛ سپس با استفاده از الگوریتم اجماع نمونه تصادفی[9] (RANSAC) سطوح مسطح[10] را پیدا می‌کنند. الگوریتم، زمانی که هیچ داده بیشتری برای برازش وجود نداشته باشد، متوقف می‌شود. در [13] از روش زنجیره مارکف مونت کارلو[11] برای مسائل قطعه‌بندی و بازسازی سطح استفاده می‌شود.

[1] Online
[2] Poppinga
[3] Surface Reconstruction
[4] Registered
[5] Strom
[6] Graph Based Segmentation
[7] Taylor
[8] Hokuyou Range Scanner
[9] RANdom SAmple Consensus

۳- روش پیشنهادی

شـــکل (۴) روندنمای[12] مراحل روش پیشنهادی را نشان می‌دهد که در ادامه هر مرحله با جزئیات بیشتری توضیح داده می‌شود.



(شکل-۴):روندنمای طرح کلی روش پیشنهادی
(Figure-4): Flowchart of proposed method

۳-۱- قطعه‌بندی تصویر

در این پژوهش برای قطعه‌بندی تصاویر RGB-D سه الگوریتم خوشه‌بندی K-Means [16]، شبکه خود سازماندهی[13] (SOM) و خوشه‌بندی C میانگین فازی[14] (FCM) مورد بررسی قرار گرفتند و نتیجه عملکرد آن‌ها با الگوریتم قطعه‌بندی صفحه‌ای پیشنهادی مقایسه شدند.

۳-۱-۱- قطعه‌بندی تصاویر RGB-D با استفاده از الگوریتم‌های خوشه‌بندی

برای قطعه‌بندی با الگوریتم‌های خوشه‌بندی، ابتدا می‌بایست داده‌های نمونه را برای خوشه‌بندی آماده کرد. هر نمونه داده شش بعد دارد. به‌عبارت‌دیگر داده‌ها به‌صورت ماتریس M × N با N=6 و M=Cols×Rows است. هر ستون این ماتریس شامل مختصات x، y و z پیکسل، به‌دست‌آمده از تصویر عمق خروجی کینکت، و رنگ پیکسل در فضای RGB است و تعداد

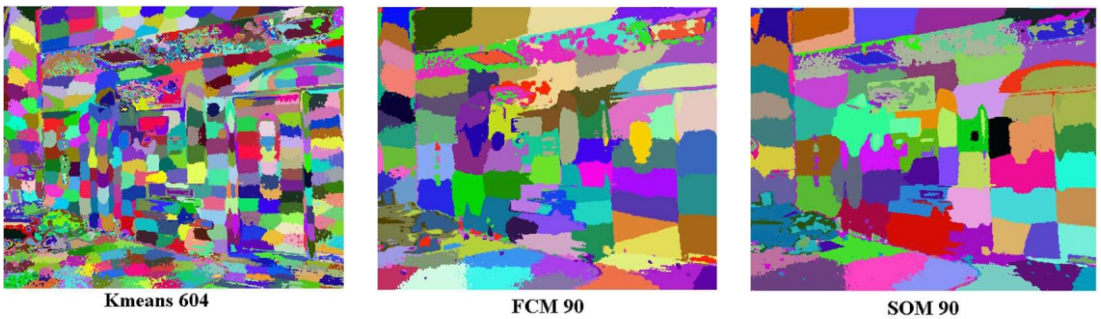
[10] Planar
[11] Markov Chain Monte Carlo
[12] Flowchart
[13] Self Organization Map
[14] Fuzzy C-Mean

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

سطرهای این ماتریس برابر تعداد پیکسل‌های تصویر اولیه است؛ سپس ماتریس نمونه‌ها به‌عنوان ورودی به الگوریتم خوشه‌بندی SOM با تعداد نرون مشخص و الگوریتم‌های خوشه‌بندی K-Means و FCM با تعداد خوشه (K) مشخص، داده می‌شود. خروجی الگوریتم‌ها، برداری است که دارای M برچسب با مقادیر $\{1,2,..,k\}$ است که تعیین می‌کند هر پیکسل از ماتریس ورودی متعلق به کدام خوشه است. بعد از به‌دست‌آوردن بردار خروجی، تصویری با رنگ‌های تصادفی که نشان دهنده قطعه‌ها است، می‌توان تولید کرد. برای این منظور ماتریسی به ابعاد K×3 و با اعداد تصادفی در بازه [۰-۲۵۵] ایجاد و از آن برای مشخص‌کردن رنگ هر قطعه استفاده می‌شود. برای دست‌یابی به نتایج بهتر، ماتریس ورودی را در بازه صفر تا یک هنجارسازی کرده و ستون‌های مربوط به مختصات هندسی را در وزن ثابت بزرگ‌تر از یک ضرب کرده تا در خوشه‌بندی، اهمیت موقعیت هندسی پیکسل نسبت به ویژگی رنگ آن، وزن بیش‌تری داشته باشد. شکل (۶) خروجی قطعه‌بندی با استفاده از الگوریتم‌های خوشه‌بندی K-Means، SOM و FCM را نشان می‌دهد.



(شکل-۵): تصویر در فضای رنگی Lab
(Figure-5): Image in Lab Color Spacepieces



(شکل-۶):خروجی قطعه‌بندی با استفاده از الگوریتم‌های خوشه‌بندی K-Means، SOM و FCM (اعداد، تعداد قطعات را تعیین می‌کنند)
(Figure-6): segmentation output using the K-Means, SOM, and FCM clustering algorithms (Numbers determine the count of pieces)

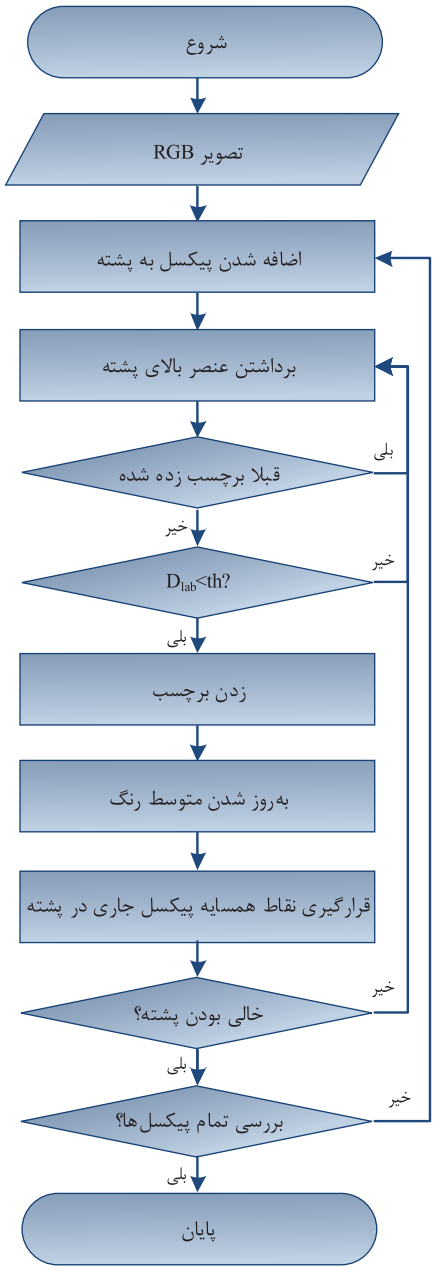
۲-۱-۳- قطعه‌بندی صفحه‌ای پیشنهادی

ابتدا توسط الگوریتم قطعه‌بندی پیشنهادی، تصویر به بخش‌های مختلف تقسیم می‌شود. برای کاهش زمان محاسبه، در ابتدا قطعه‌بندی بیش‌ازاندازه تصویر انجام و سپس به هر بخش، مدل مسطح برازش داده می‌شود [13]. این الگوریتم از سه مرحله اصلی قطعه‌بندی، برازش صفحه و ادغام قطعات تشکیل شده است که در ادامه این مراحل توضیح داده می‌شوند.

قطعه‌بندی پیشنهادی

ابتدا توسط الگوریتم قطعه‌بندی پیشنهادی، تصویر به بخش‌های مختلف تقسیم می‌شود. روندنمای الگوریتم قطعه‌بندی در شکل (۷) نشان داده شده است. در الگوریتم قطعه‌بندی پیشنهادی از ساختمان داده پشته و فاصله اقلیدسی در فضای رنگی Lab تصویر، استفاده شده است. فاصله اقلیدسی در فضای رنگی Lab، میزان شباهت رنگ را بیان می‌کند که در شکل (۷) با حرف D_{Lab} نشان داده شده است. شکل (۵) تصویر در فضای رنگی Lab را نشان می‌دهد.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

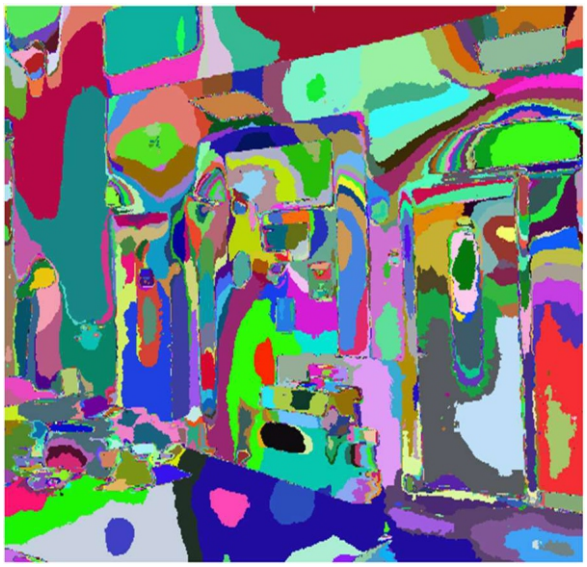


(شکل-۷): روندنمای الگوریتم قطعه‌بندی پیشنهادی
(Figure-7): Flowchart of proposed segmentation algorithm

در این الگوریتم پیکسل ابتدایی تصویر به پشته اضافه شده و تا زمانی که پشته خالی نشده باشد، عنصر بالای پشته برداشته می‌شود. درصورتی‌که پیکسل برداشته شده، درقبل به‌عنوان جزئی از قطعه برچسب نخورده باشد و فاصله اقلیدسی آن با متوسط رنگ‌های قطعه جاری در فضای رنگی Lab از مقدار آستانه کم‌تر باشد، جزء قطعه جاری محسوب می‌شود؛ سپس متوسط رنگ قطعه جاری توسط رابطه (۵) به‌روز و هشت نقطه همسایگی پیکسل مورد نظر در پشته قرار می‌گیرد. پس از خالی‌شدن پشته، الگوریتم سراغ پیکسل قطعه‌بندی‌نشدهٔ بعدی در تصویر می‌رود و این عملیات روی کل تصویر تکرار می‌شوند.

$$c(t+1) = c(t) + \frac{1}{t+1}\,(x - c(t)) \qquad (۵)$$

در رابطه (۵)، $c(t)$، متوسط t نقطه اولیه، x، نقطه جدید و $c(t+1)$، متوسط نقطه جدید است. شکل (۸) تصویر خروجی الگوریتم قطعه‌بندی پیشنهادی در فریم اول با ۳۵۲۰ قطعه را نشان می‌دهد.



(شکل-۸): تصویر خروجی الگوریتم قطعه‌بندی پیشنهادی
(Figure-8): Output image of proposed segmentation's algorithm

برازش صفحه

پس از قطعه‌بندی تصویر، صفحه برازش‌شده از نقاط سه‌بعدی حاصل از تصویر عمق هر قطعه، توسط رابطه (۶) تخمین زده می‌شود. در این رابطه A، B و C مقادیر بردار نرمال، n تعداد نقاط هر قطعه و x_i، y_i و z_i مختصات نقطه i ام می‌باشند.

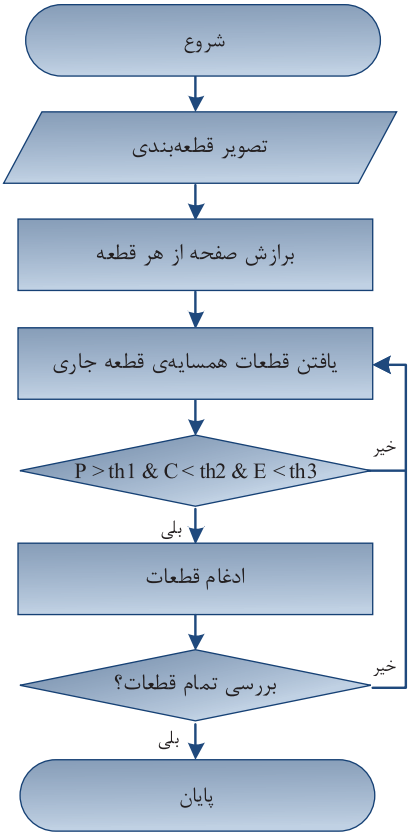
$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} y_i^2 & \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{i=1}^{n} x_i z_i \\ \sum_{i=1}^{n} y_i z_i \\ \sum_{i=1}^{n} z_i \end{bmatrix} \qquad (۶)$$

ادغام قطعات

روندنمای الگوریتم ادغام قطعات در شکل(۹) نشان داده شده است. در این شکل P، C و E به‌ترتیب نشان‌دهندهٔ حاصل‌ضرب زاویه بردار نرمال بین دو قطعه، تعداد پیکسل هر قطعه و خطای صفحه برازش شده از هر قطعه است.

قطعه‌های همسایه درصورتی‌که حاصل‌ضرب زاویه بردار نرمال آن‌ها از مقدار آستانه بیش‌تر و تعداد پیکسل هر قطعه و خطای صفحه برازش‌شده از هر قطعه کم‌تر از مقدار آستانه باشند، با هم ادغام می‌شوند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]



(شکل-۹): روندنمای الگوریتم ادغام قطعات
(Figure-9) Merging algorithms flowchart

برای مقایسه بردار قطعه‌های همسایه و ادغام آن‌ها از پشته استفاده شده است. برای ادغام، ابتدا قطعه نخست در پشته قرار داده می‌شود و مادامی‌که پشته خالی نشده باشد، قطعه‌ها از پشته خارج و حاصل‌ضرب بردار نرمال آن با بردار نرمال هر قطعه مقایسه‌شده و درصورتی‌که از مقدار آستانه بیش‌تر و خطای صفحه برازش شده از هر قطعه و تعداد پیکسل آن، از مقدار آستانه کم‌تر شوند، ضمن اضافه‌شدن قطعه‌های همسایه به داخل پشته، عمل ادغام انجام می‌شود.

برای پیداکردن قطعه‌های همسایه، اشتراک مستطیل احاطه‌کنندهٔ هر قطعه با تمام مستطیل‌های احاطه‌کننده قطعه‌های دیگر محاسبه شده و قطعه‌های دارای اشتراک، به‌عنوان قطعه‌های همسایه در نظر گرفته می‌شوند. شـــکل (۱۰) تصویر خروجی الگوریتم ادغام پیشنهادی با ۲۸۰ صفحه را نشان می‌دهد.

۳-۱-۳- مقایسه روش‌های قطعه‌بندی ارائه شده

در الگوریتم قطعه‌بندی پیشنهادی، بر اساس تغییر مقدار آستانه فاصله اقلیدسی در فضای رنگی Lab، تعداد قطعات



(شکل-۱۰): تصویر حاصل از الگوریتم ادغام پیشنهادی
(Figure-10): Output image of proposed merging algorithm

الگوریتم قطعه‌بندی پیشنهادی تغییر می‌کنند. جـــدول (۱) تعداد قطعات تصویر و زمان اجرای حاصل از این تغییر را نشان می‌دهد.

(جدول-۱): تعداد قطعات با تغییر پارامتر الگوریتم قطعه‌بندی
(Table-1): Number of components by changing the parameter of segmentation algorithm

حد آستانه فاصله اقلیدسی	تعداد قطعات تصویر	زمان اجرا (ثانیه)
10	8115	347
15	5130	283
20	3520	212
25	2719	188
30	2071	170

همان‌طورکه جـــدول (۱) نشان می‌دهد، با افزایش فاصله اقلیدسی، قطعاتی که در فضای رنگی میزان شباهت بیش‌تری با پیکسل‌های همسایه دارند، باهم ادغام می‌شوند. درنتیجه تعداد قطعات کم‌تر با اندازه بزرگ‌تری تشکیل می‌شوند و زمان اجرای الگوریتم کاهش می‌یابد.

در الگوریتم ادغام، بر اساس تغییر پارامتر زاویه بردار نرمال صفحات همسایه، تعداد صفحات تغییر می‌کنند. جـــدول (۲) نتایج حاصل از تغییر پارامتر حد آستانه زاویه بردار نرمال بین صفحات در تصویر حاصل از الگوریتم قطعه‌بندی را نشان می‌دهد.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

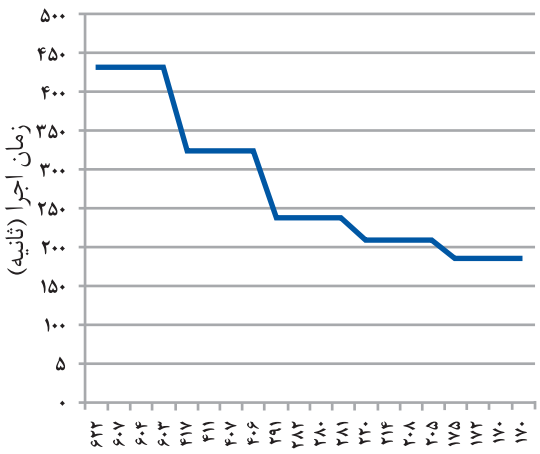
(جدول-۲): تعداد صفحات با تغییر پارامتر الگوریتم ادغام

(Table-2): Number of pages by changing the parameter of the merging algorithm

فاصله اقلیدسی	زاویه بردار نرمال (درجه)	تعداد صفحات
10	15	622
	30	607
	45	604
	60	603
15	15	417
	30	411
	45	407
	60	406
20	15	291
	30	282
	45	280
	60	281
25	15	220
	30	214
	45	208
	60	205
30	15	175
	30	172
	45	170
	60	170

همان‌طورکه جدول (۲) نشان می‌دهد تأثیرگذارترین پارامتر الگوریتم قطعه‌بندی، مقدار آستانه فاصله اقلیدسی است و تغییر مقدار آستانه زاویه بردار نرمال، تأثیر چندانی در تعداد صفحات ندارد. همچنین با افزایش زاویه نرمال بین صفحات، صفحاتی که بردار نرمال آن‌ها با بردار نرمال صفحات همسایه خود، توازی کم‌تری دارند، باهم ادغام می‌شوند. درنتیجه تعداد صفحات کم‌تر با اندازه بزرگ‌تری تشکیل می‌شوند. (۱۱) زمان اجرای مجموع هر دو الگوریتم قطعه‌بندی و ادغام را نسبت به تعداد صفحات (۲) نشان می‌دهد.

در الگوریتم قطعه‌بندی پیشنهادی، تعداد قطعات مناسب به کاربرد و شرایط مورد نظر بستگی دارد. هر قدر آستانه فاصله اقلیدسی افزایش یابد، تعداد صفحات حاصل از الگوریتم ادغام و زمان اجرا کاهش می‌یابند. در بخش نتایج عددی میزان تأثیر تعداد صفحات مختلف در میزان خطای نقشه ساخته‌شده، نشان داده خواهد شد.



(شکل-۱۱): زمان اجرا نسبت به تعداد صفحات

(Figure-11): Runtime relative to the number of pages

فاصله نقطه $\boldsymbol{x}_0 = (x_0, y_0, z_0)$ از صفحه با معادله $ax + by + cz + d = 0$ توسط رابطه (۷) محاسبه می‌شود.

$$D = \frac{ax_0 + by_0 + cz_0 + d}{\sqrt{a^2 + b^2 + c^2}} \qquad (۷)$$

در رابطه (۷)، a، b و c مقادیر بردار نرمال به‌دست‌آمده توسط رابطه (۶) و d با قراردادن مرکز ثقل صفحه در معادله آن، به‌دست آمده است.

به‌علت محدودیت‌های موجود مانند صفرشدن دترمینان در برازش صفحه به‌علت یکسان‌بودن مقادیر، عدم توانایی برازش صفحات از قطعه‌های دارای کم‌تر از سه نقطه و وجود داده‌هایی با مقدار عمق صفر به علت عدم بازگشت موج مادون‌قرمز به حس‌گر کینکت، به‌طورتقریبی چهار درصد نقاط اولیه از بین می‌روند. متوسط خطای فاصله نقاط فریم از صفحات برازش شده $۱۰^{-۱۷}$ است. به‌عبارت‌دیگر صفحات به‌خوبی روی قطعه‌ها برازش شده‌اند.

جدول (۳) زمان اجرای قطعه‌بندی با استفاده از الگوریتم‌های خوشه‌بندی SOM، FCM و K-Means برحسب ثانیه را نشان می‌دهد.

(جدول-۳): زمان اجرای قطعه‌بندی با استفاده از الگوریتم‌های خوشه‌بندی

(Table-3): Runtime segmentation using clustering algorithms

روش‌های خوشه‌بندی	30 خوشه	60 خوشه	90 خوشه
زمان اجرای SOM	118	232	433
زمان اجرای FCM	332	1522	2052
زمان اجرای K-Means	38	123	195

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

همان‌طور که جـــدول (۳) نـشان مـی‌دهد، اجرای الگوریتم K-Means برای نود خوشه، نسبت به الگوریتم‌های SOM و FCM به‌ترتیب ۲٫۲ و ۱۰٫۵ برابر سریع‌تر است. جـدول (۴) زمان اجرای قطعه‌بندی با استفاده از الگوریتم خوشه‌بندی K-Means برحسب ثانیه را نشان می‌دهند. تعداد خوشه‌های جـــدول (۴) برابر با تعداد صفحات نهایی حاصل از الگوریتم پیشنهادی، در نظر گرفته شده است تا بتوان زمان اجرای الگوریتم K-Means را با الگوریتم پیشنهادی مقایسه کرد.

از آن‌جایی که نتیجه اجرای روش‌های خوشه‌بندی استفاده‌شده به مراکز خوشه‌های اولیه بستگی دارند و مراکز خوشه‌ها به‌صورت تصادفی انتخاب می‌شوند، اعداد جـدول‌های (۳و۴)، حاصل میانگین ۱۰ بار اجرای مستقل الگوریتم‌های مذکور هستند.

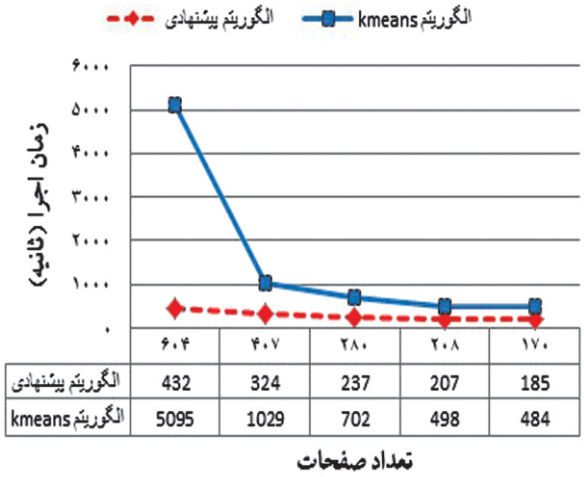
(جدول-۴): زمان اجرای قطعه‌بندی با استفاده از الگوریتم K-Means
(Table-4): runtime segmentation using K-Means algorithm

	170 خوشه	208 خوشه	280 خوشه	407 خوشه	604 خوشه
زمان اجرا	484	498	702	1029	5095

از معایب قطعه‌بندی با استفاده از الگوریتم‌های خوشه‌بندی SOM و FCM می‌توان به زمان اجرای بالا و تعیین تعداد خوشه‌ها از قبل، اشاره کرد. این الگوریتم‌ها برای تعداد خوشه‌های بزرگ‌تر، زمان اجرای چندین ساعت دارند. درحالی‌که الگوریتم قطعه‌بندی پیشنهادی بر اساس میزان فاصله اقلیدسی و زاویه بردار نرمال به‌صورت هوشمند، تعداد قطعات مناسب را استخراج می‌کند و با توجه به تعداد خوشه‌ها از سرعت به‌نسبه خوبی نیز برخوردار است.

شـکل (۱۲) تفاوت زمان اجرای الگوریتم پیشنهادی با الگوریتم K-Means را نشان می‌دهد.

همان‌طور که شـکل (۱۲) نشان می‌دهد، زمان اجرای الگوریتم K-Means برای خوشه‌های زیاد، کند است. برای مثال برای ۶۰۴ صفحه، زمان اجرای الگوریتم K-Means ۵۰۹۵ ثانیه و ۱۱٫۸ برابر الگوریتم پیشنهادی است؛ درحالی‌که زمان اجرای الگوریتم پیشنهادی، برای صفحات متفاوت، اختلاف چندانی ندارد. برای مثال زمان اجرای الگوریتم پیشنهادی برای ۴۰۷ و ۶۰۴ صفحه، به‌ترتیب ۳۲۴ و ۴۳۲ ثانیه است؛ درحالی‌که زمان اجرای الگوریتم K-Means به‌ترتیب ۱۰۲۹ و ۵۰۹۵ ثانیه است. همچنین در روش K-Means تعداد خوشه‌ها باید از قبل تعیین شوند.



(شکل-۱۲): تفاوت زمان الگوریتم K-Means و روش پیشنهادی
(Figure-12): Time difference between K-Means algorithm and proposed method

۲-۳- برازش صفحه از هر قطعه و تولید نقاط

پس از مرحله قطعه‌بندی، از قطعه‌های به‌دست‌آمده از الگوریتم‌های قطعه‌بندی مذکور، توسط رابطه (۶) صفحه برازش داده می‌شود؛ سپس روی صفحه برازش‌شده از هر قطعه، N=200 نقطه تولید می‌شود.

اگر نقطه $P(x,y,z)$ یکی از نقاط داخل قطعه باشد، مختصات P در روی صفحه برازش‌شده با معادله $ax+by+cz=d$ توسط رابطه (۸) به‌دست می‌آید.

$$P'(x',y',z')=\begin{bmatrix}x\\y\\z\end{bmatrix}+t\begin{bmatrix}a\\b\\c\end{bmatrix} \qquad (۸)$$

در رابطه (۸)، a، b و c مقادیر بردار نرمال به‌دست‌آمده توسط رابطه (۶) و t از رابطه (۹) به‌دست می‌آید.

$$t=\frac{-ax-by-cz+d}{a^2+b^2+c^2} \qquad (۹)$$

در رابطه (۹)، d با قراردادن مرکز ثقل صفحه در معادله آن و توسط رابطه (۱۰) محاسبه می‌شود.

$$d=aM_x+bM_y+cM_z \qquad (۱۰)$$

در رابطه (۱۰)، M_x، M_y و M_z به‌ترتیب نشان‌دهندهٔ میانگین مختصات x، y و z نقاط هر قطعه است. پس از محاسبه مختصات P' برای تمام نقاط داخل قطعه، N نقطه تصادفی از بین این نقاط انتخاب می‌شود. اگر تعداد پیکسل‌های قطعه از N کم‌تر باشد، تمام نقاط داخل قطعه انتخاب می‌شوند؛ سپس این عمل برای تمام قطعات تصویر،

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

تکرار می‌شود.

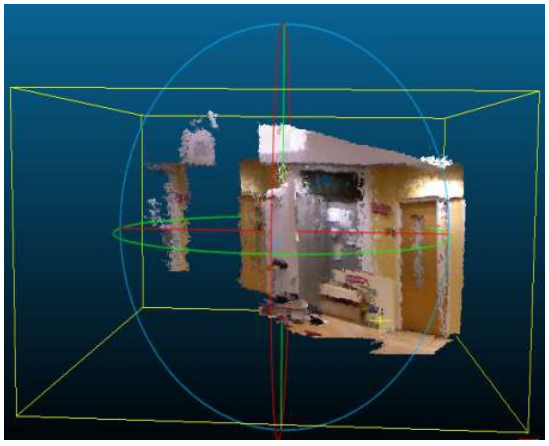
۳-۳- اجرای الگوریتم ICP روی نقاط تولیدشده

به‌جای اجرای الگوریتم ICP بر روی تمام نقاط یک فریم با ۲۳۹،۵۴۷ نقطه، الگوریتم ICP روی نقاط تولیدشده از صفحات، اجرا می‌شود. پس از اجرای الگوریتم ICP، ماتریس‌های انتقال و دوران تبدیل بین هر فریم به‌دست می‌آید؛ سپس توسط رابطه (۱۱)، تصویرها تثبیت می‌شود. A_i در رابطه (۱۱) توسط رابطه (۱۲) به‌دست می‌آید.

$$\prod_{i=1}^{n} A_i \qquad (۱۱)$$

$$A_i = \begin{bmatrix} R_i & T_i \\ \vec{0}^T & 1 \end{bmatrix}_{4\times4} \qquad (۱۲)$$

در رابطه (۱۲)، R_i و T_i نشان‌دهنده ماتریس دوران و انتقال فریم i ام و n تعداد فریم‌های موردنظر برای ساخت نقشه است. برای نمایش تصویر، رنگ‌های پیکسل‌ها در تصویر RGB، در کنار نقاط تبدیل یافته قرار می‌گیرند و ماتریس شش‌بعدی که شامل مختصات و رنگ نقاط است، با تعداد $n \times size(image)$ سطر ایجاد می‌شود. شکل (۱۳) خروجی تصویر تثبیت‌شده را برای پنج فریم متوالی به ابعاد ۵۶۱×۴۲۷ و با استفاده از الگوریتم قطعه‌بندی پیشنهادی نشان می‌دهد. ابعاد این ماتریس ۶×۱،۱۹۷،۷۳۵ است. تصاویر RGB این پنج فریم در شکل (۱۴) نشان داده شده است. برای نمایش بهتر و سریع‌تر از نسخه ۲،۶،۲ نرم‌افزار CloudCompare استفاده شده است. این نرم‌افزار، نرم‌افزاری متن‌باز برای پردازش ابرنقطه‌ای است [17].



(شکل-۱۳):خروجی تصویر تثبیت‌شده برای پنج فریم
(Figure-13): The output of registered image for 5 frames.



(شکل-۱۴): تصاویر RGB پنج فریم متوالی
(Figure-14): RGB images of 5 consecutive frames

۴- نتایج عددی

برای مقایسهٔ مدل ایجادشده توسط روش پیشنهادی با مدل ایجادشده توسط الگوریتم ICP روی تمام نقاط فریم‌ها، دویست نقطه روی صفحات به‌دست‌آمده از الگوریتم‌های قطعه‌بندی ذکرشده تولید و سپس الگوریتم ICP بین این دو مدل اجرا و خطای آن به‌عنوان خطای مدل در نظر گرفته می‌شود.

تمام الگوریتم‌ها در رایانه Core-i7-2600 با چهار گیگابایت حافظه داخلی و با نرم‌افزار متلب ۲۰۱۳، اجرا شده‌اند. همچنین مدل ایجادشده با الگوریتم‌های خوشه‌بندی، پنج بار به‌طور مستقل انجام و سپس میانگین خطای مدل و زمان آن‌ها محاسبه شده است.

به‌علت سرعت پایین الگوریتم‌های SOM و FCM و توانایی مقایسه الگوریتم‌های مختلف، این الگوریتم‌ها تنها بر روی ۲۵ فریم متوالی مورد بررسی قرار گرفته است؛ سپس

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]

الگوریتم دقیق‌تر برای فریم‌های بیش‌تر اجرا می‌شود.

جدول (۵) و شکل (۱۵) به‌ترتیب زمان اجرای الگوریتم ICP روی تمام نقاط فریم‌ها (مدل‌های مرجع) و خروجی نقشه ایجادشده در ۲۵ فریم متوالی را نشان می‌دهند.

جـدول (۶) زمان اجرا و خطای مدل تهیه نقشه با الگوریتم‌های قطعه‌بندی SOM، FCM و K-Means با ۹۰ صفحه برای ۲۵ فریم متوالی را مقایسه می‌کند.

"زمان اجرا بدون قطعه‌بندی" در تمامی جداول نشان دهنده زمان محاسبه‌شده برای هر دو مرحله "برازش صفحه از هر قطعه و تولید نقاط" و "اجرای الگوریتم ICP روی نقاط تولیدشده" (تمام مراحل به‌جز قطعه‌بندی) است.

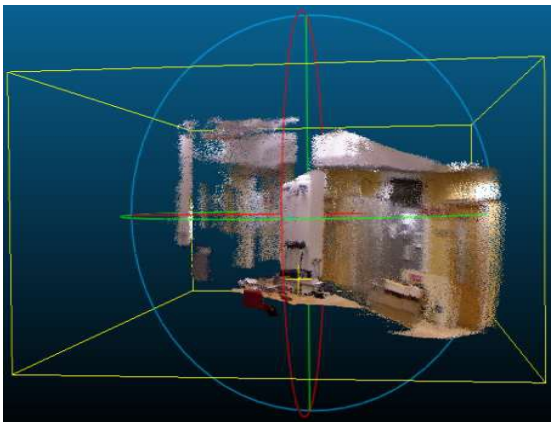
شـکل‌های (۱۶ و ۱۷) به‌ترتیب نقشه ساخته‌شده توسط الگوریتم‌های قطعه‌بندی SOM و FCM با ۹۰ صفحه برای ۲۵ فریم متوالی را نشان می‌دهند.

(جدول-۵): الگوریتم ICP روی تمام نقاط فریم‌های متوالی
(Table-5): ICP algorithm on all points of consecutive frames

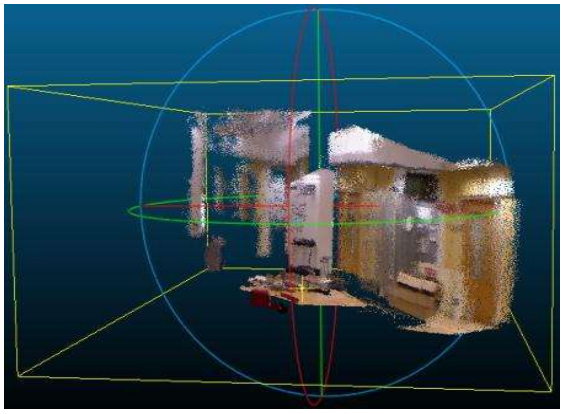
تعداد فریم‌ها	زمان (ثانیه)
25	16258
61	82894

(جدول-۶): مقایسه نقشه‌های ایجادشده توسط الگوریتم‌های قطعه‌بندی
(Table-6): Compares the maps created by the segmentation algorithms

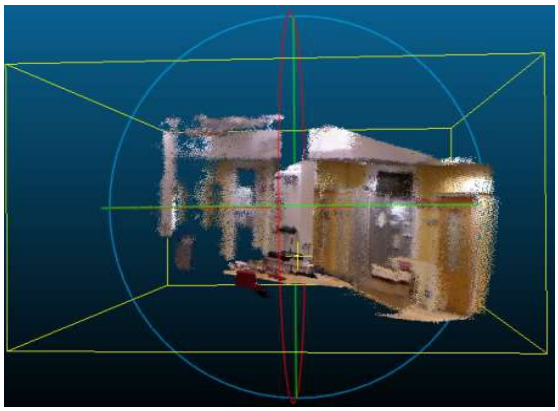
الگوریتم	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان اجرا (ثانیه)
SOM	4.7	141	17550
FCM	3.4	141	38248
K-Means	6.3	161	4453



(شکل-۱۵): خروجی الگوریتم ICP روی تمام نقاط
(Figure-15): Output of ICP algorithm on all points



(شکل-۱۶): خروجی الگوریتم قطعه‌بندی SOM با ۹۰ نرون
(Figure-16): Output of SOM segmentation algorithm with 90 neurons



(شکل-۱۷): خروجی الگوریتم قطعه‌بندی FCM با ۹۰ خوشه
(Figure-17): Output of FCM segmentation algorithm with 90 clusters

همان‌طور که جـدول (۶) نشان می‌دهد برای نود قطعه، الگوریتم K-Means از الگوریتم‌های SOM و FCM به‌ترتیب ۳٫۹ و ۸٫۶ برابر سریع‌تر است. بنابراین در ادامه، نقشه ساخته‌شده توسط الگوریتم قطعه‌بندی صفحه‌ای پیشنهادی با الگوریتم K-Means مقایسه می‌شود. جهت مقایسه، میانگین تعداد صفحات مستخرج از ۲۵ فریم متوالی در الگوریتم پیشنهادی، به‌عنوان تعداد خوشه‌های الگوریتم K-Means در نظر گرفته می‌شود. این تعداد صفحات برای الگوریتم پیشنهادی با فاصله اقلیدسی ۱۵، ۲۰ و ۲۵ در فضای رنگی Lab به‌ترتیب برابر ۴۱۹، ۳۲۳ و ۲۶۵ است.

جدول‌های (۷، ۸ و ۹) به‌ترتیب زمان اجرا و خطای مدل تهیه نقشه توسط الگوریتم‌های قطعه‌بندی صفحه‌ای پیشنهادی با فاصله اقلیدسی ۱۵، ۲۰ و ۲۵ در فضای رنگی Lab را با تعداد خوشه‌های معادل در الگوریتم K-Means برای ۲۵ فریم متوالی مقایسه می‌کند. پس از قطعه‌بندی (پیشنهادی یا K-Means)، روش پیشنهادی ("برازش صفحه

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

از هر قطعه و تولید نقاط" و "اجرای الگوریتم ICP روی نقاط تولید شده") اجرا می‌شوند.

(جدول-۷): مقایسه الگوریتم پیشنهادی با فاصله پانزده و K-Means

(Table-7): Comparison of proposed algorithms with a distance of 15 and K-Means

الگوریتم قطعه‌بندی	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان اجرا (ثانیه)
صفحه‌ای	2	4201	11789
K-Means	3.9	823	26857

(جدول-۸): مقایسه الگوریتم پیشنهادی با فاصله بیست و K-Means

(Table-8): Comparison of proposed algorithms with a distance of 20 and K-Means

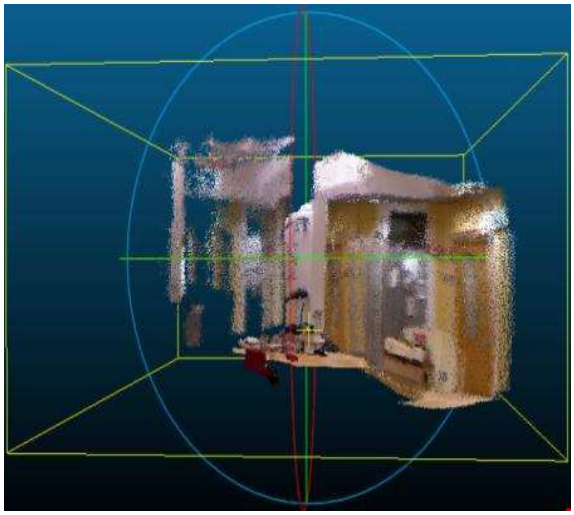
الگوریتم قطعه‌بندی	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان اجرا (ثانیه)
صفحه‌ای	2.2	2306	8052
K-Means	4	373	19930

(جدول-۹): مقایسه الگوریتم پیشنهادی با فاصله ۲۵ و K-Means

(Table-9): Comparison of proposed algorithms with a distance of 25 and K-Means

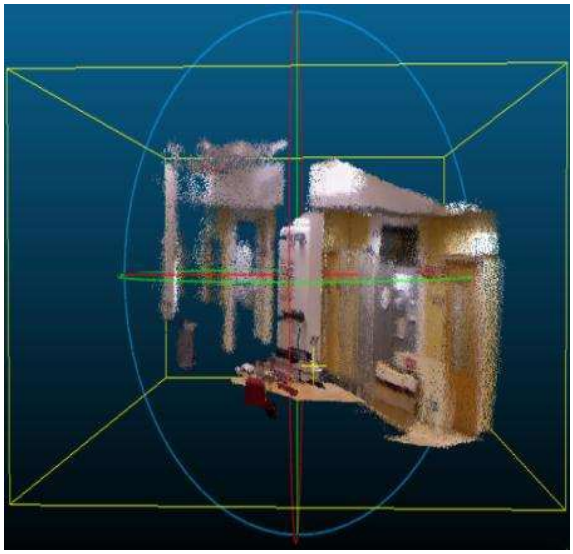
الگوریتم قطعه‌بندی	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان اجرا (ثانیه)
صفحه‌ای	2.4	1480	6438
K-Means	4.5	250	16856

مقایسه جدول‌های (۷، ۸ و ۹) نشان می‌دهد که در تهیه نقشه با ۲۵ فریم متوالی، خطای بهبودیافته با الگوریتم پیشنهادی با فاصله اقلیدسی ۱۵، ۲۰ و ۲۵ در فضای رنگی Lab نسبت به الگوریتم خوشه‌بندی K-Means با تعداد خوشه‌های معادل، به‌ترتیب ۴۹، ۴۵ و ۴۷ درصد است؛ بنابراین بالاترین دقت، متعلق به الگوریتم پیشنهادی با فاصله اقلیدسی پانزده است که سرعت اجرا را ۵۶ درصد نیز افزایش می‌دهد. از این رو در ادامه از این الگوریتم استفاده خواهد شد. شکل‌های (۱۸ و ۱۹) به ترتیب نقشه ساخته‌شده با الگوریتم‌های قطعه‌بندی پیشنهادی با فاصله اقلیدسی پانزده در فضای رنگی Lab و K-Means با ۴۱۹ خوشه برای ۲۵ فریم متوالی را نشان می‌دهند.



(شکل-۱۸): خروجی الگوریتم قطعه‌بندی پیشنهادی با فاصله ۱۵

(Figure-18) Output of the proposed segmentation algorithm with a distance of 15



(شکل-۱۹): خروجی الگوریتم قطعه‌بندی K-Means با ۴۱۹ خوشه

(Figure-19): Output of the K-Means segmentation algorithm with 419 clusters

در ادامه الگوریتم قطعه‌بندی پیشنهادی به‌جای ۲۵ فریم متوالی، بر روی ۶۱ فریم با فاصله‌های مختلف ۲، ۳، ۴ و پنج فریم و برای سه صحنه مختلف اجرا می‌شود.

جدول‌های (۱۰ و ۱۱) به‌ترتیب زمان اجرا و خطای مدل ساخت نقشه توسط الگوریتم ICP را روی تمام نقاط فریم‌ها و توسط الگوریتم پیشنهادی با فاصله‌های مختلف با مدل مرجع برای ۶۱ فریم در صحنه نخست مقایسه می‌کنند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

(جدول-۱۰): مقایسه الگوریتم ICP با فاصله‌های مختلف

(Table-10): Comparison of ICP algorithm with different distances

فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان (ثانیه)
2	1.3	41744
3	1.7	26940
4	2	20873
5	4.7	16534

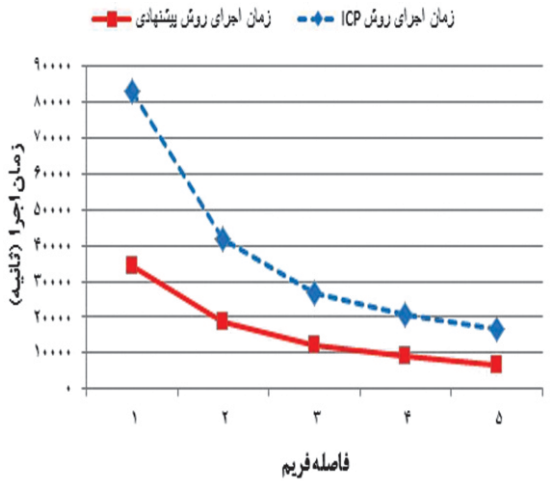
(جدول-۱۱): مقایسه الگوریتم پیشنهادی با فاصله‌های مختلف

(Table-11): Comparison of the proposed algorithm with different distances

فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان (ثانیه)
1	2.3	14029	34288
2	2.5	8178	18462
3	3.1	5215	11934
4	4	3766	9021
5	7.5	2867	6796

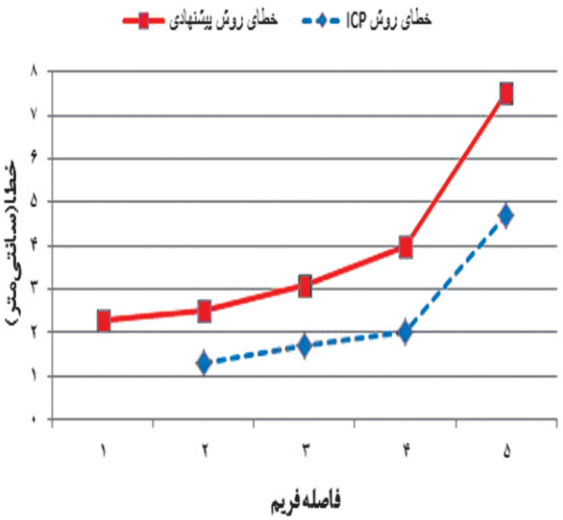
شکل‌های (۲۰ و ۲۱) به‌ترتیب زمان اجرا و خطای مدل روش پیشنهادی را با الگوریتم ICP روی تمام نقاط برای ۶۱ فریم با فاصله‌های مختلف در صحنه نخست نشان می‌دهند.

شکل‌های (۲۲ و ۲۳) به‌ترتیب خروجی الگوریتم ICP را روی تمام نقاط برای ۶۱ فریم متوالی و با فاصله پنج فریم در صحنه نخست نشان می‌دهند.



(شکل-۲۰): مقایسه زمان اجرای روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه نخست

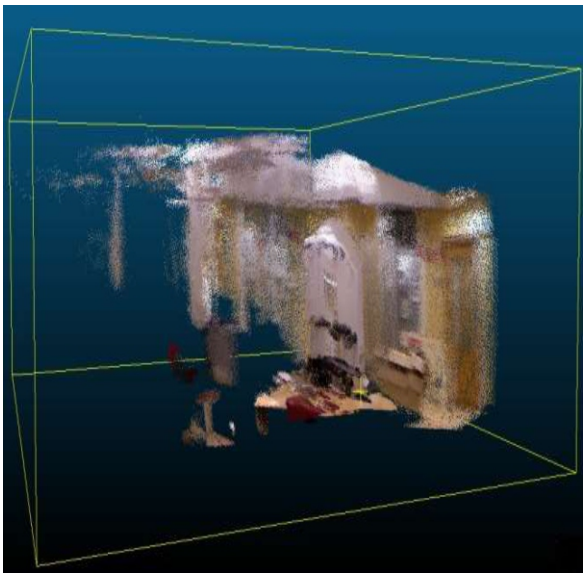
(Figure-20) Comparison of the running time of the proposed method with the ICP algorithm for 61 frames with different distances in the first scene



(شکل-۲۱): مقایسه خطای مدل روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه نخست

(Figure-21) Comparison of error model of the proposed method with the ICP algorithm for 61 frames with different distances in the first scene

شکل‌های (۲۴ و ۲۵) به‌ترتیب خروجی الگوریتم روش پیشنهادی را با فاصله اقلیدسی ۱۵ برای ۶۱ فریم متوالی و با فاصله پنج فریم در صحنه نخست نشان می‌دهند.

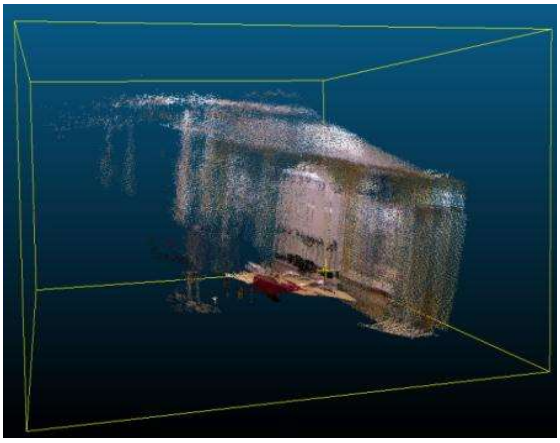


(شکل-۲۲): خروجی الگوریتم ICP برای ۶۱ فریم متوالی در صحنه نخست

(Figure-22); Output of the ICP algorithm for 61 consecutive frames in the first scene

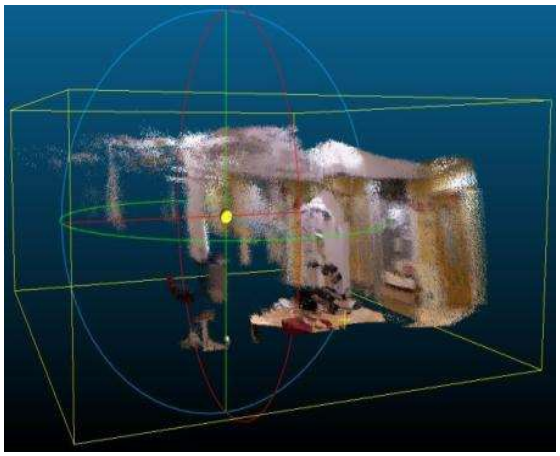
[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]



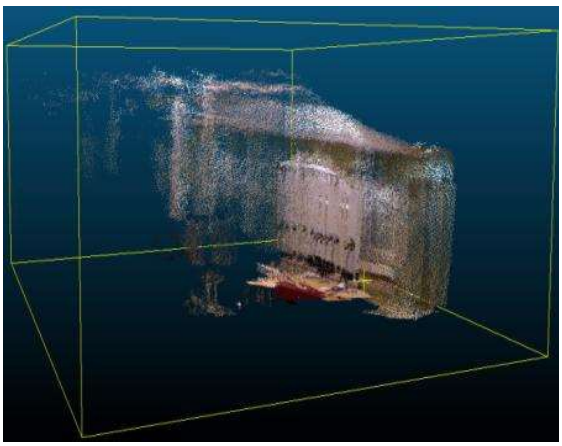
(شکل-۲۳): خروجی الگوریتم ICP برای ۶۱ فریم با فاصله پنج فریم در صحنه نخست

(Figure-23): Output of the ICP algorithm for 61 frames with 5 frames apart in the first scene



(شکل-۲۴): خروجی الگوریتم پیشنهادی برای ۶۱ فریم متوالی در صحنه نخست

(Figure-24): Output of the proposed algorithm for 61 consecutive frames in the first scene



(شکل-۲۵): خروجی الگوریتم پیشنهادی برای ۶۱ فریم با فاصله ۵ فریم در صحنه نخست

(Figure-25): Output of the proposed algorithm for 61 frames with 5 frames apart in the first scene

جدول‌های (۱۲ و ۱۳) به‌ترتیب زمان اجرا و خطای مدل ساخت نقشه توسط الگوریتم ICP را روی تمام نقاط فریم‌ها و توسط الگوریتم پیشنهادی با فاصله‌های مختلف با مدل مرجع برای ۶۱ فریم در صحنه دوم مقایسه می‌کنند.

(جدول-۱۲): مقایسه الگوریتم ICP با فاصله‌های مختلف

(Table-12): Comparison of ICP algorithm with different distances

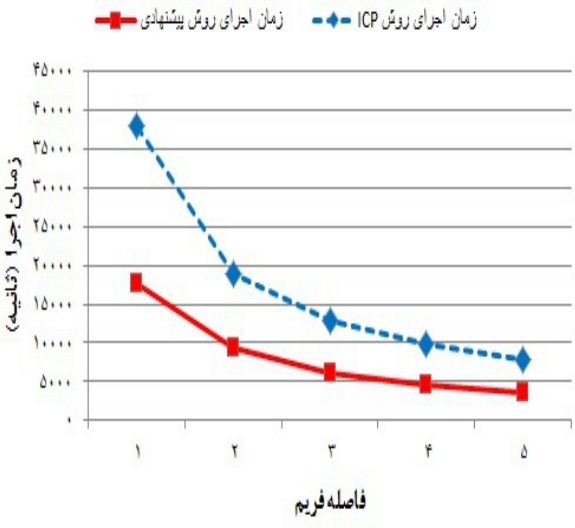
فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان (ثانیه)
1	مدل مرجع	38113
2	2.5	19002
3	4.2	12946
4	5	9803
5	5.8	7840

(جدول-۱۳): مقایسه الگوریتم پیشنهادی با فاصله‌های مختلف

(Table-13): Comparison of the proposed algorithm with different distances

فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان (ثانیه)
1	4.5	4396	17727
2	4.8	2600	9258
3	5.5	1727	6156
4	5.8	1135	4474
5	6.3	934	3613

شکل‌های (۲۶ و ۲۷) به‌ترتیب زمان اجرا و خطای مدل روش پیشنهادی را با الگوریتم ICP روی تمام نقاط برای ۶۱ فریم با فاصله‌های مختلف در صحنه دوم نشان می‌دهند.

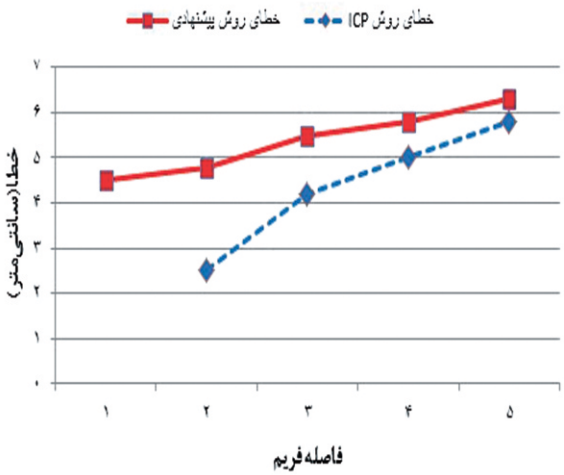


(شکل-۲۶): مقایسه زمان اجرای روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه دوم

(Figure-26): Comparison of the running time of the proposed method with the ICP algorithm for 61 frames with different distances in the second scene

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

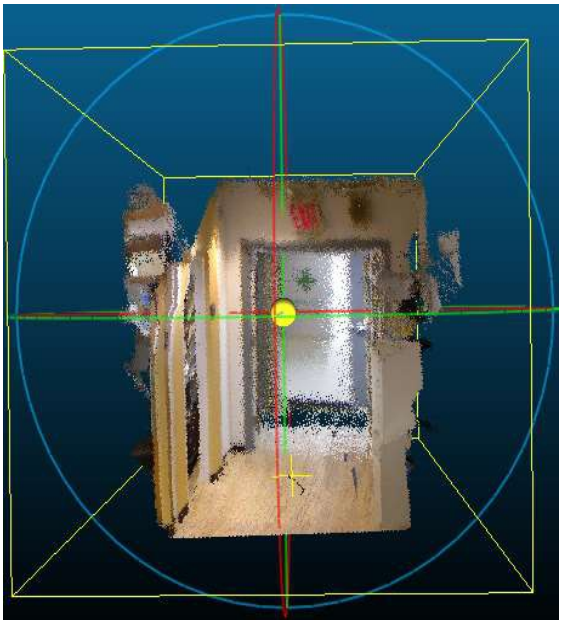


(شکل-۲۷): مقایسه خطای مدل روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه دوم

(Figure-27): Comparison of error model of the proposed method with the ICP algorithm for 61 frames with different distances in the second scene

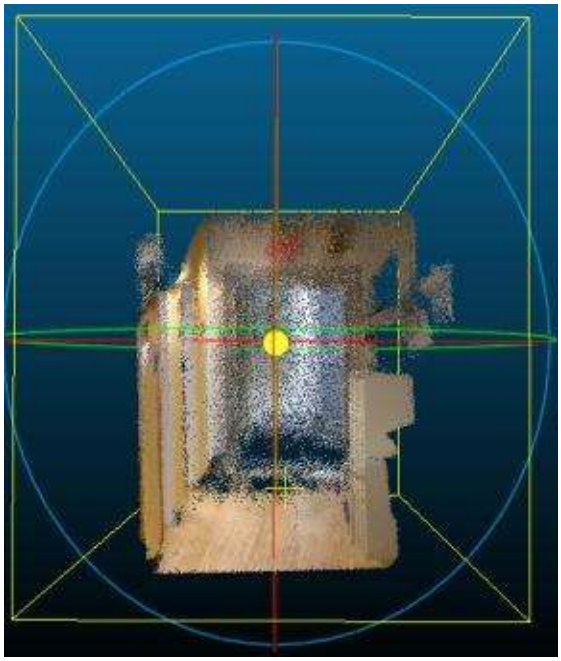
شکل‌های (۲۸ و ۲۹) به‌ترتیب خروجی الگوریتم ICP را روی تمام نقاط برای ۶۱ فریم متوالی و با فاصله پنج فریم در صحنه دوم نشان می‌دهند.

شکل‌های (۳۰ و ۳۱) به‌ترتیب خروجی الگوریتم روش پیشنهادی را با فاصله اقلیدسی پانزده برای ۶۱ فریم متوالی و با فاصله پنج فریم در صحنه دوم نشان می‌دهند.



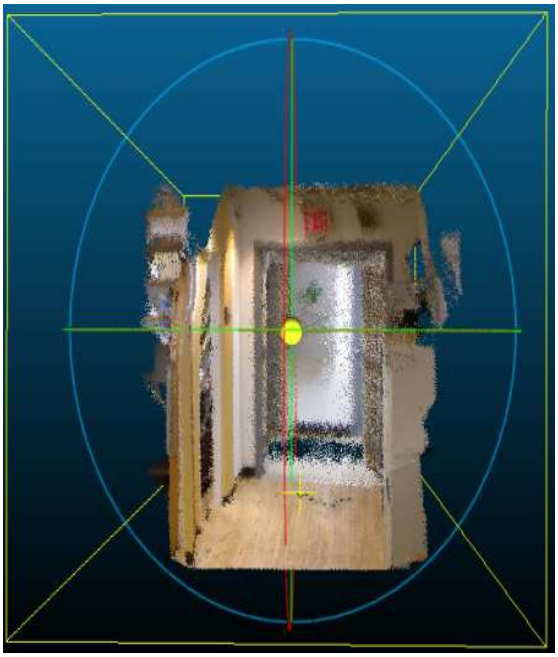
(شکل-۲۸): خروجی الگوریتم ICP برای ۶۱ فریم متوالی در صحنه دوم

(Figure-28): Output of the ICP algorithm for 61 consecutive frames in the second scene



(شکل-۲۹): خروجی الگوریتم ICP برای ۶۱ فریم با فاصله پنج فریم در صحنه دوم

(Figure-29): Output of the ICP algorithm for 61 frames with 5 frames apart in the second scene

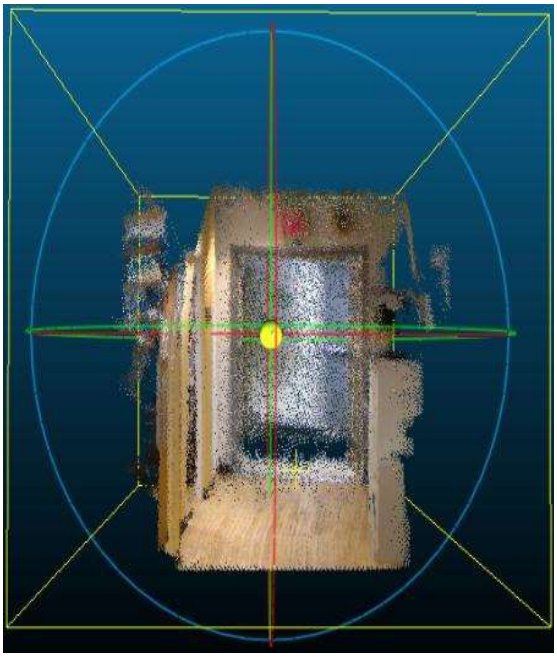


(شکل-۳۰): خروجی الگوریتم پیشنهادی برای ۶۱ فریم متوالی در صحنه دوم

(Figure-30): Output of the proposed algorithm for 61 consecutive frames in the second scene

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]



(شکل-۳۱): خروجی الگوریتم پیشنهادی برای ۶۱ فریم با فاصله پنج فریم در صحنه دوم

(Figure-31): Output of the proposed algorithm for 61 frames with 5 frames apart in the second scene

جدول‌های (۱۴ و ۱۵) به‌ترتیب زمان اجرا و خطای مدل ساخت نقشه توسط الگوریتم ICP را روی تمام نقاط فریم‌ها و توسط الگوریتم پیشنهادی با فاصله‌های مختلف با مدل مرجع برای ۶۱ فریم در صحنه سوم مقایسه می‌کنند.

(جدول-۱۴): مقایسه الگوریتم ICP با فاصله‌های مختلف

(Table-14): Comparison of ICP algorithm with different distances

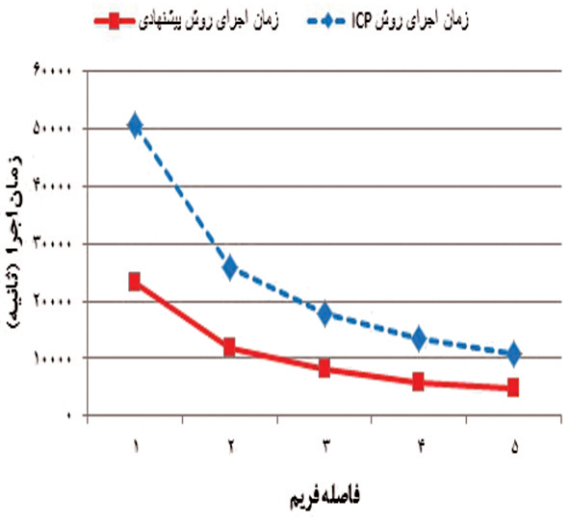
فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان (ثانیه)
1	مدل مرجع	50736
2	7.2	25883
3	11	17840
4	14.1	13470
5	15.4	10877

(جدول-۱۵): مقایسه الگوریتم پیشنهادی با فاصله‌های مختلف

(Table-15): Comparison of the proposed algorithm with different distances

فاصله فریم‌ها	خطای مدل (سانتی‌متر)	زمان اجرا بدون قطعه‌بندی (ثانیه)	زمان (ثانیه)
1	4.7	6452	23315
2	5.1	3280	11664
3	5.2	2598	8242
4	5.8	1545	5703
5	5.9	1248	4597

شکل‌های (۳۲ و ۳۳) به‌ترتیب زمان اجرا و خطای مدل روش پیشنهادی را با الگوریتم ICP روی تمام نقاط برای ۶۱ فریم با فاصله‌های مختلف در صحنه سوم نشان می‌دهند.



(شکل-۳۲): مقایسه زمان اجرای روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه سوم

(Figure-32): Comparison of the running time of the proposed method with the ICP algorithm for 61 frames with different distances in the third scene



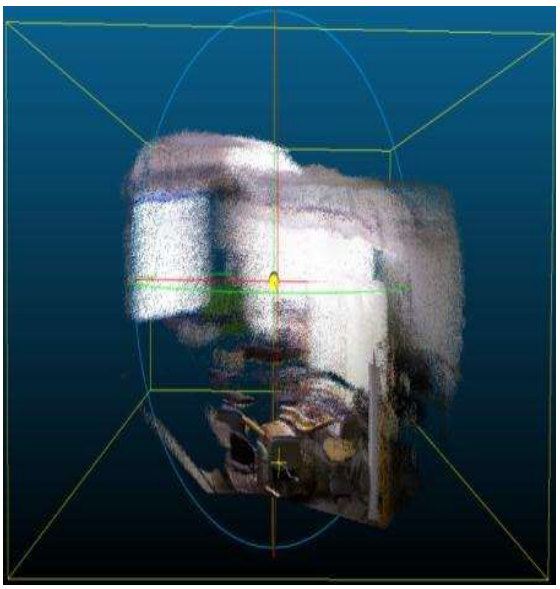
(شکل-۳۳): مقایسه خطای مدل روش پیشنهادی با الگوریتم ICP برای ۶۱ فریم با فاصله‌های مختلف در صحنه سوم

(Figure-33): Comparison of error model of the proposed method with the ICP algorithm for 61 frames with different distances in the third scene

شکل‌های (۳۴ و ۳۵) به‌ترتیب خروجی الگوریتم ICP روی تمام نقاط را برای ۶۱ فریم متوالی و با فاصله پنج فریم در صحنه سوم نشان می‌دهند.

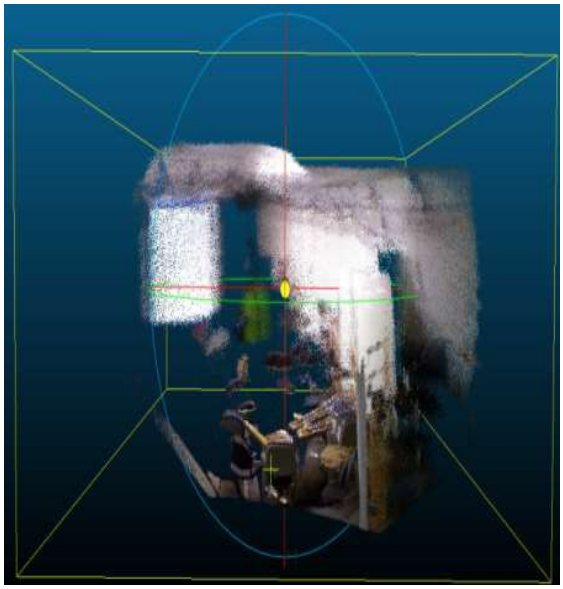
شکل‌های (۳۶ و ۳۷) به‌ترتیب خروجی الگوریتم روش پیشنهادی را با فاصله اقلیدسی ۱۵ برای ۶۱ فریم متوالی و با فاصله ۳ و ۵ فریم در صحنه سوم نشان می‌دهند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]
[DOI: 10.29252/jsdp.14.3.143]



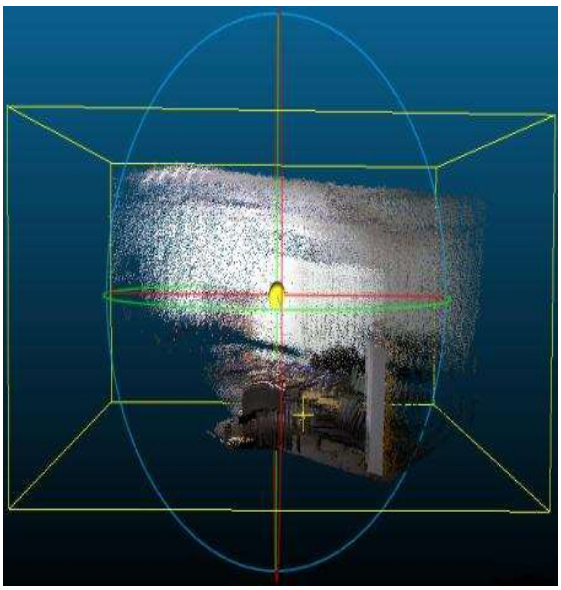
(شکل-۳۴): خروجی الگوریتم ICP برای ۶۱ فریم متوالی در صحنه سوم

(Figure-34): Output of the ICP algorithm for 61 consecutive frames in the third scene



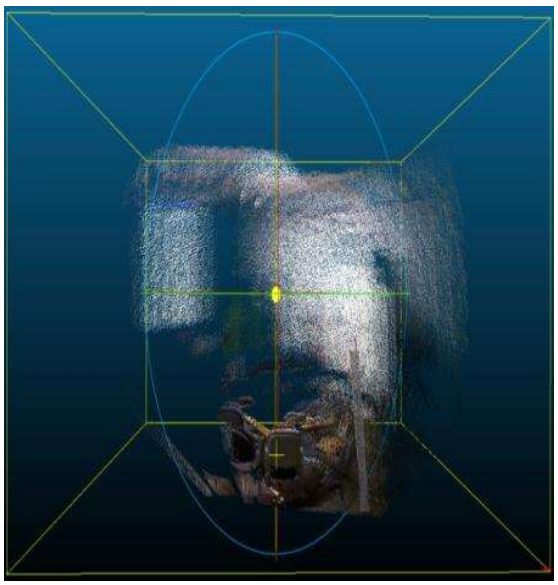
(شکل-۳۶): خروجی الگوریتم پیشنهادی برای ۶۱ فریم متوالی در صحنه سوم

(Figure-36): Output of the proposed algorithm for 61 consecutive frames in the third scene



(شکل-۳۵): خروجی الگوریتم ICP برای ۶۱ فریم با فاصله ۵ فریم در صحنه سوم

(Figure-35): Output of the ICP algorithm for 61 frames with 5 frames apart in the third scene



(شکل-۳۷): خروجی الگوریتم پیشنهادی برای ۶۱ فریم با فاصله ۵ فریم در صحنه سوم

(Figure-37): Output of the proposed algorithm for 61 frames with 5 frames apart in the third scene

۵- بحث و نتیجه‌گیری

همان‌طور که روندنمای شکل (۴) نشان می‌دهد، تصاویر RGB و عمق گرفته‌شده از کینکت ورودی الگوریتم پیشنهادی هستند. تصویر عمق توسط روابط (۱) و (۲) به ابرنقطه‌ای تبدیل می‌شود که به‌ترتیب این تصاویر در شکل‌های (۱، ۲ و ۳) نشان داده شده‌اند. نخست تصویر به

چهار روش الگوریتم‌های صفحه‌ای پشنهادی، خوشه‌بندی K-means، خوشه‌بندی SOM و خوشه‌بندی FCM قطعه‌بندی می‌شود. در الگوریتم صفحه‌ای پیشنهادی، قطعه‌بندی بیش‌ازاندازه تصویر انجام و سپس به هر قطعه، با رابطه (۶)، مدل صفحه‌ای برازش داده می‌شود. درنهایت این قطعات بر اساس حاصل‌ضرب زاویه بردار نرمال بین دو قطعه، تعداد پیکسل هر قطعه و خطای صفحه برازش‌شده از هر قطعه

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

ادغام می‌شوند. ماتریس ورودی، در روش‌های خوشه‌بندی K-means، SOM و FCM، ترکیب تصویر RGB و ابر نقطه‌ای است. این ماتریس ورودی، در بازه صفر تا یک هنجارسازی‌شده و جهت افزایش تأثیر موقعیت هندسی پیکسل نسبت به رنگ آن، ستون‌های ابر نقطه‌ای در وزن ثابت بزرگ‌تر از یک ضرب می‌شوند.

پس از مرحله قطعه‌بندی، با رابطه (۶) از قطعه‌های به‌دست آمده از الگوریتم‌های قطعه‌بندی مذکور، صفحه برازش داده می‌شود؛ سپس تعداد ثابتی نقطه روی این صفحات تولید می‌شود؛ درنهایت ماتریس‌های دوران و انتقال بین هر دو فریم، توسط اجرای الگوریتم ICP روی این نقاط تولید شده، به‌جای اجرای آن روی تمام نقاط فریم، تخمین و توسط رابطه (۱۱) مدل سه‌بعدی ساخته می‌شود. بنابراین حجم داده‌ها کاهش و ساخت نقشه سه‌بعدی تسریع می‌یابد.

شکل‌ها و جداول بخش نتایج عددی نشان می‌دهند که دقت مدل ایجادشده در روش پیشنهادی به‌نحوه قطعه‌بندی وابستگی چندانی ندارد؛ ولی الگوریتم قطعه‌بندی پیشنهادی، سرعت و دقت بیش‌تری نسبت به الگوریتم‌های خوشه‌بندی K-Means، SOM و FCM دارد.

در مسئله تهیه نقشه، کم‌ترین خطای مدل برای ۲۵ فریم متوالی، دو سانتی‌متر و مربوط به اجرای قطعه‌بندی پیشنهادی با فاصله اقلیدسی پانزده است. در این حالت، سرعت ۲۷ درصد افزایش می‌یابد.

برای صحنه نخست بیش‌ترین افزایش سرعت برای ۶۱ فریم متوالی، ۵۹ درصد با خطای مدل ۲٫۳ سانتی‌متر و برای فریم‌های غیرمتوالی، ۹۲ درصد با خطای مدل ۷٫۵ سانتی‌متر است. برای صحنه دوم بیش‌ترین افزایش سرعت برای ۶۱ فریم متوالی، ۵۳ درصد با خطای مدل ۴٫۵ سانتی‌متر و برای فریم‌های غیرمتوالی، ۹۱ درصد با خطای مدل ۶٫۳ سانتی‌متر است. برای صحنه سوم بیش‌ترین افزایش سرعت برای ۶۱ فریم متوالی، ۵۴ درصد با خطای مدل ۴٫۷ سانتی‌متر و برای فریم‌های غیرمتوالی، ۹۱ درصد با خطای مدل ۵٫۹ سانتی‌متر است که مربوط به اجرای قطعه‌بندی پیشنهادی با فاصله ۵ فریم است.

با مقایسه شکل‌های (۲۲ و ۲۴) یا شکل‌های (۳۵ و ۳۷) می‌توان دریافت که از لحاظ نمایش، روش پیشنهادی بهتر از روش ICP (مدل مرجع) است. همچنین شکل (۳۳) نشان می‌دهد که در صحنه سوم، با افزایش فاصله فریم‌ها، خطای روش ICP افزایش می‌یابد؛ درحالی‌که در روش پیشنهادی تغییر زیادی نمی‌کند. مقایسه جدول‌های (۱۴ و ۱۵) نشان می‌دهد که روش پیشنهادی در مقایسه با روش ICP، خطای مدل را در صحنه سوم با فاصله پنج فریم ۶۲ درصد کاهش می‌دهد.

جداول بخش نتایج عددی نشان می‌دهند که بیش‌ترین زمان اجرای ساخت نقشه مربوط به الگوریتم‌های قطعه‌بندی است. برای مثال همان‌طور که جدول (۶) نشان می‌دهد، مدل ساخته شده برای ۲۵ فریم متوالی توسط اجرای قطعه‌بندی FCM با ۹۰ خوشه، خطای مدل ۳٫۴ سانتی‌متر با زمان اجرای ۳۸،۲۴۸ ثانیه دارد. درحالی‌که فقط ۱۴۱ ثانیه آن مرتبط به تمام مراحل روش پیشنهادی به غیر از قطعه‌بندی است. به‌عبارت‌دیگر ۹۹٫۶ درصد از زمان اجرا، مربوط به الگوریتم قطعه‌بندی است. درصورتی‌که این الگوریتم هر فریم را فقط به ۹۰ قطعه تقسیم می‌کند؛ بنابراین برای تسریع بیش‌تر الگوریتم‌های قطعه‌بندی را به‌صورت موازی اجرا یا از GPU می‌توان استفاده کرد.

به‌کارگیری نقشه ایجادشده با توجه به برازش صفحه به ابرنقطه‌ای حاصل از تصویر عمق حس‌گر کینکت، در مسئله مکان‌یابی و تهیه نقشه هم‌زمان قابل استفاده است.

۶- مراجع 6- References

[1] S. Thrun, "Robotic mapping: a survey," in Exploring artificial intelligence in the new millennium, L. Gerhard and N. Bernhard, Eds., ed: Morgan Kaufmann Publishers Inc., 2003, pp. 1-35.

[2] V. HÖGMAN, "Building a 3D map from RGB-D sensors," Master dissertation, Dept. Computer Science, Computer Vision and Active Perception Laboratory Royal Institute of Technology (KTH), Stockholm, Sweden, 2011.

[3] P. Vieira and R. Ventura, "Interactive mapping in 3D using RGB-D data," IEEE International Symposium on Safety Security and Rescue Robotics (SSRR), 2012, pp. 1-6.

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments," in Experimental Robotics, ed: Springer Berlin Heidelberg, 2010, pp. 477-491.

[5] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in Proc. 12th European conference on Computer Vision - Volume Part V, Florence, Italy, 2012.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]

[6] C. Couprie, C. e. Farabet, L. Najman, and Y. LeCun, "Indoor Semantic Segmentation using depth information,", International Conference on Learning Representations (ICLR2013), 2013.

[7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 14, pp. 239-256, 1992.

[8] A. Nüchter, 3D Robotic Mapping: Springer-Verlag Berlin Heidelberg, 2009.

[9] N. V. D. Hau, N. D. Thang, T. T. L. Anh, and T. C. Hung, "Combined Plane and Point Registration of Sequential Depth Images for Indoor Localization " in Third International Conference on Advances in Computing, Electronics and Electrical Technology - CEET 2015, 2015, pp. 136-140.

[10] L. Tae-kyeong, L. Seungwook, L. Seongsoo, A. Shounan, and O. Se-young, "Indoor mapping using planes extracted from noisy RGB-D sensors," in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012, pp. 1727-1733.

[11] W. Caihua, H. Tanahashi, H. Hirayu, Y. Niwa, and K. Yamamoto, "Comparison of local plane fitting methods for range data," in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, pp. I-663-I-669 vol.1.

[12] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3D range images," in Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008, pp. 3378-3383.

[13] C. Erdogan, M. Paluri, and F. Dellaert, "Planar Segmentation of RGBD Images Using Fast Linear Fitting and Markov Chain Monte Carlo," in Proc. Ninth Conference on Computer and Robot Vision, 2012.

[14] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, 2010, pp. 2131-2136.

[15] C. J. Taylor and A. Cowley, "Fast scene analysis using image and range data," in Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 3562-3567.

[۱۶] ق. حسین پور، "یک الگوریتم ردیابی خودرو مبتنی بر ویژگی با استفاده از گروه بندی سلسله مراتبی ادغام و تقسیم،" مجله پردازش علائم و داده ها، ۱۳۹۴.

[16] Gh. Hossein poor, "A feature-based vehicle tracking algorithm using hierarchical cluster integration and division," in signal and data Processing (JSDP), 1394.

[17] D. Girardeau-Montaut, "Cloudcompare, a 3D point cloud and mesh processing free software," http://www.danielgm.net/cc (accessed 08.02.2016) 2011.



مقداد پاک‌نژاد مدرک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر به‌ترتیب در گرایش‌های نرم‌افزار و هوش مصنوعی از دانشگاه شیخ بهایی اصفهان در سال ۱۳۹۰ و دانشگاه یزد در سال ۱۳۹۴ اخذ کرد. زمینه‌های مورد علاقه ایشان بینایی کامپیوتر و پردازش تصویر است.
نشانی رایانامهٔ ایشان عبارت است از:
packnezhad@stu.yazd.ac.ir



مهدی رضائیان دارای دکترای فتوگرامتری از دانشگاه ETH سوییس بوده و مدرک کارشناسی ارشد خود را در رشتهٔ مهندسی پزشکی (بیوالکتریک) از دانشگاه تهران و کارشناسی خود را در رشتهٔ مهندسی برق (الکترونیک) از دانشگاه صنعتی خواجه نصیرالدین طوسی اخذ کرده است. وی هم‌اکنون عضو هیئت علمی گروه مهندسی کامپیوتر دانشگاه یزد است. زمینهٔ پژوهشی مورد علاقه ایشان شامل بینایی ماشین، پردازش تصویر و تشخیص الگو است.
نشانی رایانامهٔ ایشان عبارت است از:
mrezaeian@yazd.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-06]

[DOI: 10.29252/jsdp.14.3.143]