

Enhancing Dense 3D-Reconstruction Using Covariance Information from Monocular SLAM

Dr. Iyad Hatem^{*}
Yasin Yousif^{**}

(Received 22 / 1 / 2019. Accepted 11 / 2 / 2019)

□ ABSTRACT □

Dense 3D-reconstruction from images is considered a multi-stage and slow process in computer vision. However, it is also important because of the data nature that it gives about the 3D shape of the different objects. If these objects are in places hard to reach by humans, then a remote-controlled robot may be used to do the job.

In this paper, a method to solve the previous problem is presented. The method depends on a camera mounted on a robot and a velocity sensor attached to that robot. The robot has to map the environment and localize itself within it simultaneously, using the Extended Kalman Filter approach. The output data is used to build a 3D model, for the object of interest. The goal here is to increase the accuracy using the covariance data of the estimated camera poses. This helps in choosing a subset of the images for the 3D-reconstruction. Lastly, a dense point cloud is formed after a bundle adjustment process.

Many experiments were done using simulation to compare the performance with and without considering covariance in the image selection phase then in the Sparse Bundle Adjustment (SBA) phase. At the end, the results were discussed.

This paper concludes that selecting the images with consideration of the covariance gives better accuracy than ignoring it. While considering the computed 2D-projections covariance in SBA cost equation did not show any increase in the overall accuracy.

Keywords: Extended Kalman filter, Covariance matrix, Simultaneous Localization and Mapping, Sparse Bundle Adjustment, 3D-Reconstruction, Structure from motion

^{*} Associate Professor, Department of Mechatronics, Faculty of mechanical & electrical engineering, University of Tishreen, Lattakia, Syria.

^{**}Postgraduate Student, Department of Mechatronics, Faculty of mechanical & electrical engineering, University of Tishreen, Lattakia, Syria.

تحسين إعادة البناء ثلاثية الأبعاد الكثيفة باستخدام معلومات التغيرات من طريقة SLAM وحيدة الكاميرا

*الدكتور: إياد حاتم

**ياسين يوسف

(تاريخ الإيداع 2019 / 1 / 22. قُبل للنشر في 2019 / 2 / 11)

□ ملخص □

تعد إعادة البناء ثلاثية الأبعاد الكثيفة من الصور إحدى المهام متعددة المراحل والبطيئة في رؤية الحاسب ولكنها مهمة نظراً لطبيعة البيانات التي تقدمها حول الشكل الفراغي للأجسام. وإذا كانت هذه الأجسام في مواضع يصعب الوصول لها، يمكن استعمال روبوت متحكم به عن بعد للقيام بالمهمة.

في هذا البحث تم وضع طريقة لحل المسألة السابقة، باستعمال كاميرا وحيدة على روبوت، للقيام أولاً برسم البيئة والمسار بالتزامن بتطبيق مقارنة مرشح كالمان الموسع. ثم استخدام البيانات الناتجة لبناء نموذج ثلاثي الأبعاد للجسم موضع الاهتمام. الهدف هنا هو الحصول على دقة أكبر بالاستعانة ببيانات التغيرات المرافقة لمواضع الكاميرات التي يقدمها المرشح لتحسين عملية اختيار الصور لبناء النموذج ثلاثي الأبعاد. بالنهاية يتم إنشاء غيمة نقطية كثيفة للنموذج بعد عملية تعديل حزم مشنت Sparse Bundle Adjustment.

تم إجراء عدة تجارب من خلال المحاكاة للمقارنة بين الأداء مع وبدون إدخال التغيرات لمرحلة اختيار الصور، وكذلك لمرحلة عملية تعديل الحزم المشنت SBA، ومن ثم مناقشة النتائج.

خُصّص البحث إلى أن اختيار الصور مع أخذ التغيرات بعين الاعتبار يعطي دقة أفضل مما هو عليه بدون أخذه. أما إضافة التغيرات لتقديرات المساقط في عملية SBA فلم تعطي دقة إضافية في التجارب.

الكلمات المفتاحية: مرشح كالمان الموسع، مصفوفة التغيرات، تحديد الموضع ورسم البيئة المتزامنين، تعديل الحزم المشنت، إعادة البناء ثلاثية الأبعاد، البنية من الحركة.

* أستاذ مساعد - قسم هندسة الميكاترونك - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.

** طالب دراسات عليا (ماجستير) - قسم هندسة الميكاترونك - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.

مقدمة:

إن تطبيقات عمليات إعادة البناء من الصور كثيرة ومتنوعة، ومنها إنشاء أجسام وبيئات افتراضية لألعاب الفيديو والتخطيط العمراني بمساعدة الحاسب وحفظ المناطق التراثية بشكل رقمي، ولكن استخدام صور ثنائية البعد للكاميرا للقيام بذلك يبدو صعباً، ومع ذلك هناك عاملان يجعلان من استخدام الكاميرات العادية حلاً مرغوباً، أولهما الكلفة القليلة للكاميرات وثانيهما الانتشار الكبير لها.

لذلك فقد تم التركيز هنا على الكاميرات المثبتة على الروبوتات وهذا مهم لأتمتة عملية التقاط الصور وتعزيز القدرة على إعادة البناء ثلاثية الأبعاد لمناطق واسعة أو مناطق يصعب الوصول إليها مع الحصول على نماذج افتراضية ذات دقة بالمستوى المطلوب، وأحد المهام الأساسية والمباشرة هي إنشاء بيئة للروبوت نفسه، ليتمكن لاحقاً من التعرف على الأماكن وتفاذي العوائق بتلك البيئة.

بالتالي يتم استخدام طرق Visual SLAM الترشيفية كمرحلة أولى للحصول على نماذج ثلاثية الأبعاد دقيقة لبيئة أو لجسم محدد، ولكن شرط العمل بالزمن الحقيقي يجعلها تقدم نتائج أقل دقة. و يمكن لاحقاً القيام بالمزيد من العمل على خرج Visual SLAM لزيادة دقتها إلى مستويات أفضل. حيث يعتمد اختيار مجموعة صور البناء على التوزيع الهندسي المنتظم وقيم التباين Covariance الأقل بين تلك الصور ونحصل على غيمة نقطية أولية. ليتم لاحقاً تصغير الخطأ بقيمتها باستخدام عملية SBA ، وأخيراً تتم إعادة بناء كثيفة للجسم المركزي. وهكذا يمكن مكملة الطريقتين Visual SLAM و 3D-Reconstruction لتحقيق أفضل دقة ممكنة.

نقدم في هذه الورقة بدايةً هدف البحث وبعده تُعرض الأعمال السابقة في كل من الطريقتين المستخدمتين وهما Visual SLAM وإعادة البناء من الفيديو للغيمة النقطية. أما القسم الرابع فيعرض نظرة عامة على النظام بشكل كامل، والقسمين الخامس والسادس فيتطرقان لمرحلتين MonoSLAM و 3D-reconstruction بالترتيب. القسم السابع يشرح كيفية اشتقاق تباينات المسافات للنقاط، وفي القسم الثامن يتم عرض التجارب ونتائجها ومناقشتها وأخيراً بالقسم الأخير تم طرح أفكار لتطوير أبعاد.

أهمية البحث وأهدافه:

يهدف هذا البحث إلى تقليل صعوبة عملية إعادة البناء من الصور عبر توفير بيانات إضافية تعزز دقتها. وهذه البيانات تمثل التباين والتي يتم حسابها في مرحلة تحريك الروبوت من خلال تطبيق مقارنة مرشح كالمان الموسع Extended Kalman Filter

1 الدراسات المرجعية:

بما أن الطريقة المقترحة هنا تتكون من عمليتين هما إعادة البناء من الفيديو 3D-Reconstruction from video وطريقة SLAM البصرية فقد تم مراجعة الأبحاث لكلا المجالين كل على حدة، وكذلك لأبحاث تقوم بربط العمليتين معاً.

1.1 طريقة SLAM البصرية (Visual SLAM)

تعد الورقة في [1] من أولى الأبحاث التي تم فيها استخدام مرشح كالمان الموسع EKF لحل مسألة Monocular SLAM وذلك لتعقب كل من موضع الكاميرا بالإضافة للغيمة النقطية للبيئة.

تم لاحقاً إضافة تطويرات لها، و أهمها استخدام تمثيل آخر للنقاط عند التهيئة [2] ، إضافة لتقنية مسرعة للبحث وتدعى البحث الفعال Active Search و تتميز طريقة المرشح هذه بسرعتها التي تحقق الأداء المطلوب، و لكنها

تبقى محدودة بأخطاء تقريب مرشح كالمان الموسع للعلاقات اللاخطية بالإضافة لفرضية ثبات البيئة التي ستؤدي لأخطاء كثيرة بحالة البيئة الديناميكية. و هذا ما تم العمل عليه بالتعديلات اللاحقة في [3] حيث عدلت الطريقة لتقادي الأخطاء الناتجة عن الأجسام التي تعترض المشهد Occluding Objects و العودة للتقديرات الصحيحة بعد زوالها بالإضافة لتحسين متعلق بإلغاء تشويش الصورة عند تحريك الكاميرا السريع، و هذه الصيغة الأخيرة للطريقة هي الأكثر شبيهاً بالمطبق هنا، ولكنها تعاني من أخطاء التقريب بالإضافة لأخطاء نموذج الحركة القائم على فرضية ثبات السرعة.

بالنسبة للمقاربات الأخرى التي تستخدم مرشح كالمان الموسع هناك التطبيق في [4] والذي يقوم على تحديث حالة الروبوت فقط أي موضعه ودورانه وفق علاقات المطابقة بين الصور بدون أخذ تمثيل النقاط بالبيئة المحيطة وهذا لا يحل المسألة المتعلقة برسم البيئة Mapping. و لذلك يسمى هذا النمط من المسائل بالمرجع المذكور [4] بقياس المسار البصري Visual Odometry وهو يهتم فقط بحساب مسار الكاميرا دون الاهتمام بالبيئة [5].

بالنظر بشكل عام لحلول visual SLAM نجد حلولاً منافسة لا تستخدم المرشحات مطلقاً بل تستخدم طرقاً تدعى بتعديل الحزمة المحلي LBA Local Bundle Adjustment، وأحدث صيغها هو ORB-SLAM وقد جرت مقارنة بين هذه الطرق و الطرق الترشيحية في [6] وفيها تم توضيح تفوق هذه الطرق للحلول Local Bundle Adjustment (LBA) على الطرق الترشيحية بحالات محددة من ناحية الدقة، إلا أن طريقة مرشح كالمان تتميز بسهولة تطبيقها بالإضافة للحصول على معلومات التغيرات للقيم المقدرة عن طريقها مباشرة.

إحدى أنواع الحلول الأخرى تسمى بالحلول المباشرة Direct Methods في حين تسمى بقية الحلول وضمنها الطرق السابقة التي تعتمد على اشتقاق السمات من الصور بالحلول غير المباشرة Indirect Methods . تتميز الحلول المباشرة، وأشهرها [7] [8] بأنها تقوم بتعقب و إعادة بناء كل بكسل بالصورة بدل استخدام السمات فقط، وهذا له نتيجتان مباشرتان، أولاً الغيمة النقطية الناتجة كثيفة، و ثانياً الحسابات اللازمة كثيرة.

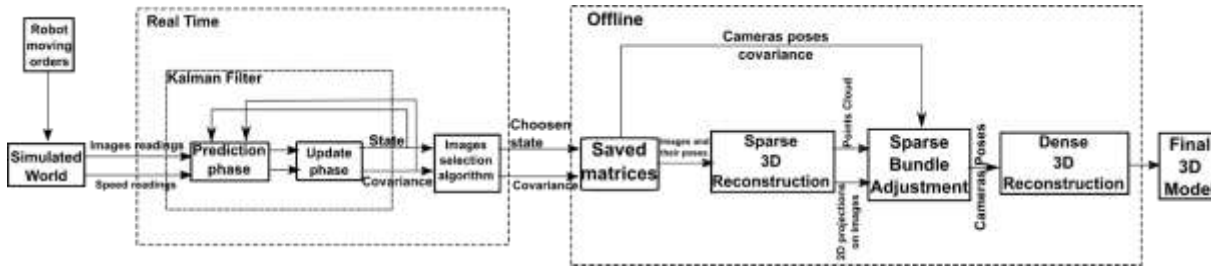
1.2 مجال إعادة البناء من الفيديو

بالنسبة لمرحلة حساب البنية ثلاثية الأبعاد فقد تم اتباع طريقة معروفة كما في الفصل 12 من [9] ، ليتم بعدها إنجاز تعديل حزم مشتت Sparse Bundle Adjustment (SBA) على النتائج كما هو في [10]. في بحث سابق [11] تم اعتماد مسألة مشابهة للمسألة قيد الدراسة هنا، أي استخدام عمليات Visual SLAM لإنشاء غيمة نقطية كثيفة ليس بالزمن الحقيقي Offline، حيث تم اختيار مجموعة صور حسب التوزيع الهندسي لأماكن الكاميرات. تم أيضاً تحسين تقديرات الموضع للصور المختارة لاحقاً ولكن أضاف هذا عبئاً حسابياً ولم يتجنب اختيار الصور المشوشة أو ذات التقديرات كبيرة الضجيج. كما لم يتم استخدام التغيرات فيه مطلقاً.

تم اتباع طريقة مشابهة عموماً في [12] حيث تم حساب مسار الكاميرا بغير الزمن الحقيقي ومن ثم اختيار مجموعة أولى من الصور حسب التوزيع الهندسي ولاحقاً إضافة مجموعة ثانية تعتمد على عدد السمات التي تم مطابقتها مع إطارات لاحقة، وهذا يجعل عملية اختيار الصور مكلفة حسابياً ويهمل مدى التغيرات بمواضع الكاميرات.

2 نظرة عامة على النظام:

يوضح الشكل (1) نموذجاً عاماً لمراحل عمل النظام. حيث تتكون الطريقة الكاملة من جزء أول بالزمن الحقيقي ويطبق على بيئة المحاكاة، حيث يتم عند تشغيل المحاكاة التحكم بحركة الروبوت والكاميرا فووه ضمن المشهد وهذا يعطي الدخل الخاص بالحساسات لخوارزمية المرشح ومعها خوارزمية اختيار الصور.



الشكل (1) نظرة عامة على النظام.

ويتمثل خرجها النهائي ببضعة صور مع مواضعها ومصفوفات تغييرها، ليتم حفظها كنواتج ويستمر العمل هكذا حتى انتهاء حركة الروبوت.

عندها يبدأ عمل الجزء الثاني بغير الزمن الحقيقي حيث يتم تمرير الأشعة الخاصة بمواضع الكاميرات بالإضافة للصور نفسها إلى خوارزمية إعادة البناء المشتتة Sparse لتعطي غيمة نقطية مشتتة. تُمرر هذه الغيمة بدورها إلى طريقة أمثلة للقيم كلها Sparse Bundle Adjustment SBA مع قيم تغاير المساقط المحسوبة بصيغة تقريبية، ومن هذه المرحلة يتم الحصول على القيم الجديدة لمواضع الكاميرات. وأخيراً يتم القيام بعملية إعادة بناء كثيفة لإعطاء غيمة كثيفة ودقيقة للنموذج المطلوب.

3 تشغيل برنامج MonoSLAM:

تم الاعتماد أساساً على البرنامج المطابق لما في المرجع [3] ولكن مع جملة تعديلات ستتم الإضاءة عليها فيما يلي. بعد تمرير الصور وقراءات بقية الحساسات يتم تطبيق خوارزمية المرشح بدءاً بالقياس ومن ثم التنبؤ فالتحديث، ولكن عملياً لا نعتبر دخل القياس هنا إلا قراءات الصور بعد مطابقة السمات ضمنها أما قراءات السرعة الانتقالية والدورانية فتؤخذ كمتغير ضجيج يؤثر على قيم شعاع السرعة الانتقالية والدورانية ضمن شعاع الحالة للمرشح. يتم تنفيذ دورة كاملة للمرشح (تنبؤ + تحديث) ضمن البرنامج فقط عند ورود صورة جديدة وانتهاء المرشح من معالجة الصورة السابقة.

بعد قراءة الصورة يتم قراءة آخر قيمة لحساس السرعة، ومن ثم حساب الأشعة المطلوب إضافتها للسرعة في المرشح أي $\Delta \Omega$ ، ΔV والتي تشكل شعاع الضجيج، حسب العلاقات التالية:

$$\begin{aligned} \Omega_{cam} &= \omega_z * (R_{state} * \vec{Z}) \\ \Delta \Omega &= \Omega_{cam} - \Omega_{state} \end{aligned} \quad (1)$$

أما العنصر الثالث من شعاع الدوران (حول Z) فيعطى بالعلاقة:

$$\theta_z = rotation2vector(R_{state})[2] \quad (2)$$

$$\Delta V = \begin{bmatrix} V_x * \cos(\theta_z) \\ V_x * \sin(\theta_z) \\ 0 \end{bmatrix} - V_{state}$$

حيث R_{state} مصفوفة الدوران الحالية، V_{state} هو شعاع السرعة الحالي، ω_z سرعة الدوران حول Z (من حساس السرعة)، و V_x السرعة على X (من حساس السرعة)، Ω_{state} السرعة الدورانية الحالية، و \vec{Z} الشعاع الواحد على المحور Z.

تلك العلاقات تعطي القيم اللازم إدخالها بمرحلة التنبؤ للمرشح. وفي تلك المرحلة، يتم أيضاً حساب أماكن السمات الجديدة بالصورة اللاحقة.

بعد ذلك تبدأ مرحلة التحديث بالمطابقة للسمات بين الصورتين وهذا يتم أيضاً بطريقة مبسطة تتمثل بعملية ضرب ترابطي Correlation بين المربعات الممثلة للرقع Patches من الصورة المحددة للسمات والأماكن المتوقعة لها ومقارنة الناتج مع عتبة وهي طريقة أقل دقة من غيرها ولكنها أسرع، يتم بعدها ترشيح المطابقات الناتجة وفق طريقة تدعى 1-point -RANSAC [13].

من ثم يتم تشكيل مصفوفة كالمات وإنجاز تحديث الحالة والتغيرات، وأخيراً إيجاد سمات جديدة بالصورة الجديدة إذا انخفض عدد السمات المتعقبة Tracked عن حد أدنى. بعد التحديث يتم تنفيذ برنامج اختيار الصور لعملية إعادة البناء المطلوبة.

من الملاحظ أن دورة المرشح لا تتأثر بتفاوت أزمنة ورود الصور لأن الفارق الزمني أو الخطوة الزمنية Δt يتم حسابها مجدداً مع ورود كل صورة و تُجرى وفق قيمتها الجديدة كل الحسابات.

3.1 اختيار الصور حسب التغيرات:

بعد التحديث للمرشح، نقوم بحساب قيمة تعبر عن المسافة المقطوعة وكمية الاستدارة الحاصلتين للكاميرا بين الموضع الأخير المحفوظ للروبوت والموضع الحالي كمجموع.

حسب قيمة هذا المقدار يتم تقرير هل تُحفظ الصورة أم لا، والخوارزمية (1) تحلل طريقة العمل هذه، حيث أن الفكرة خلف تصميمها هي مراعاة ثلاثة أمور:

- حصول انتقال ودوران للروبوت بمقدار كاف لاختيار الصورة لأن الصور المتقاربة تعطي ترتيبات كبيرة عند إعادة البناء ثلاثي الأبعاد.
- إعطاء وزن للدوران أكثر من الانتقال.
- الاستفادة من التغيرات المحسوب لأشعة الحالة للكاميرات وذلك باختيار الصورة الأقل تغيراً قدر الإمكان.

خوارزمية (1): آلية اختيار الصور حسب التغيرات والدوران والانتقال

الدخول: I ، $State_{current}$ ، $\Sigma_{current}$ ، $q_{current}$ ، $C_{current}$

الخروج: I_{chosen} ، Σ_{chosen} ، $State_{chosen}$

1- التهيئة:

$$State_{chosen} = [0, 0, 0, 0, 0, 0, 0] ; \text{old_pose} = 0$$

$$C_{prev} = State_{chosen}[0:3]$$

$$q_{prev} = State_{chosen}[3:7]$$

$$T_2 = 1000,000 ; \lambda_3 = 0.00008$$

2- لكل صورة واردة ا نحسب:

$$a = C_{current} - C_p$$

$$b = q_{current}[3:7] - q_{prev}[3:7]$$

$$dist = \lambda_1 a + \lambda_2 b \text{ (حيث } \lambda_1, \lambda_2 \text{ هي ثوابت تجريبية)}$$

$$passed = \text{old_pose} - dist$$

3- إذا كان $passed > \frac{MoveThresh}{2}$ و $passed < MoveThresh$ نحسب:

$$P = Sum(diag(\Sigma_{current}))$$

○ إذا كان $T_2 > P$:

$$T_2 = P$$

$$State_{temp} = State_{current}$$

$$I_{temp} = I$$

$$\Sigma_{temp} = \Sigma_{current}$$

$$dist_{temp} = dist$$

4- إذا كان $MoveThresh \leq passed$:

$$T_2 = 1000,000$$

○ إذا كان $Sum(diag(\Sigma_{current})) - T_2 > 3$:

$$State_{chosen} = State_{temp}$$

$$\Sigma_{chosen} = \Sigma_{temp}$$

$$I_{chosen} = I_{temp}$$

$$\text{old_pose} = dist_{temp}$$

○ وإلا:

$$State_{chosen} = State_{current}$$

$$\Sigma_{chosen} = \Sigma_{current}$$

$$I_{chosen} = I_{current}$$

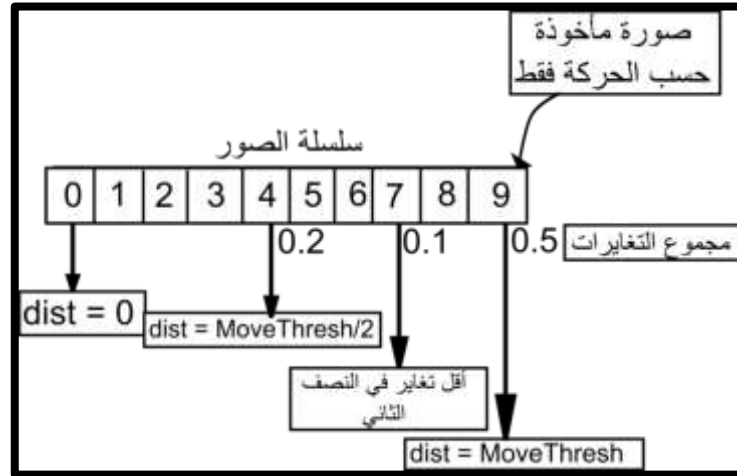
$$\text{old_pose} = dist$$

الجدول (1) يبين دلالات الرموز بالخوارزمية 1 السابقة

جدول (1) دلالات الرموز المستخدمة بالخوارزمية (1).

الصورة الواردة	/
شعاع الرباعي الممثل للدوران الحالي لكاميرا الروبوت	$q_{current}$
مركز كاميرا الروبوت الحالي	$C_{current}$
مصفوفة التغيرات لموضع الكاميرا الحالي	$\Sigma_{current}$
شعاع الحالة الحالي للكاميرا	$State_{current}$
الصورة المختارة كنتيجة	I_{chosen}
شعاع الحالة المرافق للصورة المختارة	$State_{chosen}$
مصفوفة التغيرات المرافقة للصورة	Σ_{chosen}
قيمة المجموع الأخير للدوران والانتقال	old_pose

الطريقة السابقة تقسم حركة الروبوت سواء انتقال أو دوران إلى مرحلتين، المرحلة الأولى تكون حركة الروبوت ضمن مجال محدد بنصف عتبة الحركة MoveThresh أي طالما كانت قيمة التابع المقترح أقل منه والمرحلة الثانية أكبر من نصف للعتبة ولكن أصغر من العتبة كاملة، كما يبين الرسم التوضيحي:



الشكل (2) كيفية اختيار الصور مع أخذ التغيرات والتحرك بعين الاعتبار.

إن تحديد قيم العتبات و الثوابت بتلك الطريقة يتم تجريبياً و بشكل يرتبط بالسرعات المفترضة و نمط الحركة وعدد الصور المطلوبة. فمثلاً فيما يتعلق بالثابتين λ_1 , λ_2 يتم أخذهما بحيث يكون λ_2 أكبر من λ_1 و ذلك لإعطاء الدوران وزناً أكبر من الانتقال بالمجمل، وبالنسبة للدوران الحاصل فقد تم أخذه على أنه الفرق المباشر لأشعة الرباعي. أما في أعمال سابقة [11] [12] تم أخذ الزاوية بين المحاور البصرية للكاميرات كتمثيل للدوران الحاصل يدخل بطريقة اختيار الصور ولكن لغرض التجربة تم الاكتفاء بالفرق بين أشعة الرباعيين مع التحديد الجيد لقيم λ_1 , λ_2 لتحقيق نتائج جيدة.

4 تشكيل الغيمة النقطية الأولية المخففة:

بعد الحصول على مواضع وقيم دوران الكاميرات من خرج المرشح واختيار مجموعة من الصور الملتقطة بشكل مناسب، ينتهي الجزء الذي يتطلب العمل فيه أداءً بالزمن الحقيقي.

تتمثل الخطوة التالية باستخدام هذه الصور مع التقديرات الأولية لقيم دورانها ومواضعها بالإضافة لبارامترات الكاميرا الداخلية لإعطاء غيمة نقطية مشتتة (غير كثيفة).

تم التطبيق لإعادة البناء باستخدام توابع لمكتبة OpenCV [14] حيث تم العمل على الصور والقيم المدخلة بعدة مراحل، وهو مطابق للطرق كما في [9] لإنشاء البنية من الحركة وحساب البنية، والغاية من هذا البرنامج فقط هي تقديم بقيم للغيمة النقطية لعمليات الأمثلة اللاحقة، المتمثلة بتعديل الحزم Bundle Adjustment كمهمة أساسية للبرنامج، والتالي هو تفاصيل العمل على مراحل البرنامج.

4.1 المرحلة الأولى: حساب المسارات Tracks للسّمات:

أولاً نقرأ كل صورة ونطابقها مع التي تليها وذلك أولاً باشتقاق سمات ORB [15] والمطابقة وفق مكتبة FLANN [16] ومن ثم تطبيق اختبارات إضافية لإيجاد التطابقات الجيدة، بالإضافة لذلك يتم حساب مواضع النقاط ثلاثية الأبعاد بين كل صورتين متتاليتين وإعادة إسقاطها عليهما، وذلك لإيجاد خطأ إعادة الإسقاط لكل منها، ومن ثم استبعاد التطابقات ذات الأخطاء الأكبر من عتبة ما.

هذا يحسن كثيراً من جودة التطابقات ويقلل كثيراً من القيم المتطرفة Outliers، وأخيراً تم حفظ السمات الجديدة المطابقة بشكل ناجح ضمن مصفوفات خاصة لتشكل سلاسل تطابقات للسمات Tracks عبر عدة صور.

4.2 المرحلة الثانية: بناء الغيمة النقطية:

لبناء النقاط من سلسلة التطابقات الخاصة بها، يتم بناء نقطة بين كل زوج صور تقع فيه تطابقات ضمن السلسلة المعنية وهكذا قد نحصل على عدة نتائج مختلفة لنفس النقطة، وهذا الأمر تم الاستفادة منه لإنجاز اختبار إضافي لدقة التطابقات، مما يساعد على استبعاد أي سلسلة تطابقات لا تعطي قيمةً مقاربةً لنقاط الفراغ ثلاثية الأبعاد التي تنتج عن كل من تلك الأزواج بفرض أن عدد التطابقات 3 على الأقل. ولأخذ قيمة نهائية للنقطة، نحسب متوسط تلك النقاط على كل من الإحداثيات.

بالنتيجة يتم الحصول على الغيمة النقطية التي ستمرر مع المساقط المرتبطة بكل نقطة منها والصور لتعديل الحزم المشتتة، ويمكن عرض النتيجة الأولية للغيمة النقطية ببرامج مثل MeshLab [17] (شكل 4).

5 تعديل الحزم المشتت مع التغيرات:

بعد حساب الغيمة النقطية المشتتة للمشهد، حيث إن الحركة أخذت شكل حلقة مغلقة وليست دائرية بالضرورة في التجارب وهذا متعمد لتسليط الصور على الجسم المركزي، نحتاج لعملية أمثلة Optimization تحديداً لمواضع الكاميرات والنقاط، وهذا تم باستعمال الطريقة في [18] وضمن إطار العمل ROS Robotic Operating System [19]، دخل هذا البرنامج سيكون بالإضافة للغيمة النقطية ومواضع الكاميرات والمساقط هو قيم تغير مواضع الكاميرات التي ستستعمل لحساب قيم تغير المساقط، وذلك بتطبيق طريقة تحسبها تقريباً من مصفوفات تغير مواضع الكاميرا، وتسمى انتشار التغيرات، Covariance Propagation [20].

ويمكن ضمن البرنامج أن يتم استبعاد النقاط ذات الأخطاء الكبيرة التي تُصنف كقياسات سيئة.

أخيراً يتم أخذ قيم مواضع الكاميرات والنقاط الجديدة الناتجة كخرج. وبعدها يتم إجراء عملية إعادة البناء من الصور بأعداد أكبر من السمات للحصول على نموذج نهائي كثيف.

5.1 حساب تغير المسافات من تغير الكاميرات:

من الممكن إدخال قيم تغير المسافات ضمن معادلة الكلفة، ونسعى من خلال ذلك لزيادة التحسين بعملية الأمثلة المرتبطة بدقة قيم التغيرات، ومن خرج المرشح لدينا تغير موضع الكاميرا فقط لذلك سنحاول حساب تغير المسافات من خلاله لإدخال تلك القيم كما ورد في [10].

وهناك تغير آخر هو تغير التثليث Triangulation ولكن لم يتم أخذه في التجارب والتركيز كان فقط على دراسة أثر التغيرات الأول الناتج عن الكاميرات فقط.

تتلخص طريقة حساب تغير المسافات من تغير الكاميرا بعملية تقريبية حيث يمكن إيجاد تغير متغير خرج تابع ما $Y = f(X)$ إذا عرفنا تغير الدخل X ومشتق هذا التابع نسبة للدخل J_f حسب العلاقة التالية [20]:

$$Y_{cov} = J_f * X_{cov} * J_f^T \quad (3)$$

حيث X_{cov} هو تغير الدخل و J_f^T هو منقول مصفوفة الجاكوبيان للتابع f ، و Y_{cov} تغير الخرج، و هكذا نحصل على تغير المسافات بفرض ثبات النقطة ثلاثية البعد و باعتبار تابع إسقاط النقطة على موضع الكاميرا الممثل كدخل X هو التابع المناظر f بالعلاقة السابقة، والمسقط هو الخرج Y .

النتائج والمناقشة:

لتقدير التحسين لمرحلة اختيار الصور تم إضافة مقطع لبرنامج المرشح يقوم باختيار الصور بناء على مدى الحركة والدوران فقط وبنفس العتبة MoveThresh المستخدمة لبرنامج اختيار الصور الآخر بالطريقة المقترحة الذي كان مع أخذ التغيرات كعامل إضافي.

بالنسبة للمرحلة الأولى MonoSLAM تم وضع القيم التالية للبارامترات بتجارب المحاكاة تجريبياً لزيادة الدقة قدر الإمكان:

الجدول (2) عناصر القطر الرئيسي لمصفوفة التغيرات لشعاع الضجيج مع قيم أخرى لاشتقاق ومطابقة السمات.

الانحراف المعياري لضجيج السرعة الخطية على x	sigma_vx	0.0000004
الانحراف المعياري لضجيج السرعة الخطية على y	sigma_vy	0.0000004
الانحراف المعياري لضجيج السرعة الخطية على z	sigma_vz	0.0000000004
الانحراف المعياري لضجيج السرعة الدورانية حول x	sigma_wx	0.000000002
الانحراف المعياري لضجيج السرعة الدورانية حول y	sigma_wy	0.0000002
الانحراف المعياري لضجيج السرعة الدورانية حول z	sigma_wz	0.0000000002
حجم نافذة السمة	window_size	30
عامل تصغير الصورة	Scale	10
العدد الأصغر للسمات بالصورة	min_features	20
العدد الأعظم للسمات بالصورة	max_features	35
عامل تكبير نافذة البحث عن مطابقات	sigma_size	4

أما ثوابت برنامج اختيار الصور المستخدم بالتجارب تم اختيارها تجريبياً أيضاً كما بالجدول (3):

جدول (3) قيم الثوابت المستخدمة ببرنامج اختيار الصور.

3.33	عامل الضرب بالانتقال λ_1
57.29	عامل الضرب بالدوران λ_2
26	عتبة الحركة MoveThresh
100000	قيمة التهيئة لأقل عتبة ترتيب مقبولة T_2
$10^{-5} \times 8.5$	عتبة فرق التغيرات المقبول لأخذ صورة 3

وستجري المقارنة بين صور تم اختيارها وفق الحركة وصور تم اختيارها مع أخذ التغيرات لمواقعها بعين الاعتبار إضافة للحركة، وذلك لنفس سلسلة الصور ونفس الحركة وهذا يقدم توضيحاً مباشراً لأهمية التغيرات. تتشكل نتيجة المقارنة هنا بعد القيام بمرحلة إعادة البناء المشتتة للنقاط بدلالة عدد النقاط الناتجة، لأنه سيبدل على قيم أدق لمواقع الكاميرا ونواحي أفضل للصور وتطابقات أكثر.

في التجارب كان الروبوت يدور والكاميرا موجهة نحو جسم مركزي ليكمل دورة كاملة بطول 26 متر، ضمن بيئة Gazebo [21] كما الشكل (3). و نوع الروبوت المثبت عليه الكاميرا هو TurtleBot والكاميرا مدورة بحيث تأخذ الصور جانبياً. تم أخذ الصور ثنائية البعد فقط من الكاميرا، بالإضافة لبيانات حساسات السرعة في الروبوت الذي يتم التحكم به من قبل المستخدم

تم إجراء عدة تجارب للمحاكاة، بنفس المشهد وتم حفظ مجموعات الصور ولكن مع تغيير معدل أخذ الصور أحياناً بتغيير MoveThresh، حتى تتقارب أعداد الصور من بعضها، مع ملاحظة أن التجربة الأخيرة بالجدول (4) هي تكرار للتجربة قبل الأخيرة على نفس الصور ولكن مع زيادة عتبات الأخطاء عند إيجاد النقاط، أما باقي التجارب فتمت وفق بارامترات موحدة لبرنامج إعادة البناء المشتتة، حيث نلاحظ النتائج المدرجة بالتالي:

جدول (4) نتائج إعادة البناء لمجموعتي الصور.

رقم التجربة	عدد الصور لأول مجموعة	عدد النقاط الناتجة لأول مجموعة بلا تغير	عدد الصور لثاني مجموعة	عدد النقاط الناتجة لثاني مجموعة مع تغير
1	31	1478	31	1613
2	43	3011	43	3589
3	45	2951	43	3431
4	51	4783	52	4754
5	51	5169	50	5237
6	51	6558	50	6913

الملاحظ هنا هو الزيادة بعدد النقاط للمجموعة الثانية التي تأخذ التغيرات بعين الاعتبار، حيث يتحقق هذا لكل التجارب ماعدا الرابعة وذلك بفرق ضئيل. السبب وراء هذا أن الفرق بين التغيرات للمواقع المختلفة بين المجموعتين كان صغيراً وبالتالي قل أثره.

ولتأكيد هذا التعليل، قمنا بإضافة شرط جديد وهو أن يكون فرق مجموع عناصر القطر الرئيسي لمصفوفة التغيرات للصورة المختارة بالمجموعة الأولى عن مجموع عناصر القطر الرئيسي للصورة المختارة بالمجموعة الثانية أكبر من حد معين λ_3 ليتم اختيار الصورة من المجموعة الثانية وإلا نختار الصورة من المجموعة الأولى.

تم وضع القيمة هنا λ_3 تجريبياً (كانت $10^{-5} \times 8.5$) وهذه القيمة مرتبطة بقيم مصفوفة تغير الضجيج . والجدول (5) يبين ثلاثة تجارب إضافية مع مراعاة نفس الظروف للتجربة الرابعة بالجدول السابق حيث كان المسار أقل شبهاً بالدائرة من غيره لأول تجربة فيه، والتجارب الباقية تبين أداء جيداً أيضاً، وهذا بعد تطبيق الشرط الأخير.

جدول (5) نتائج إعادة البناء لمجموعتي الصور بعد اشتراط فرق تغير أكبر من عتبة ما

رقم التجربة	عدد الصور لأول مجموعة	عدد النقاط الناتجة لأول مجموعة بلا تغير	عدد الصور لثاني مجموعة	عدد النقاط الناتجة لثاني مجموعة مع تغير
1	42	3120	42	3387
2	43	3399	43	5023
3	43	2865	43	3081

وفي أغلب التجارب السابقة من الملاحظ بوضوح زيادة أكبر بالنقاط للمجموعة الثانية من الصور، وذلك عائد لتجاوز صور تم تقدير موضعها بشكل خاطئ، مثلاً في حالة التوقف المفاجئ أو تغيير السرعة المفاجئ.

وفي معظم التجارب ازداد عدد النقاط الناتجة بالمئات تقريباً وذلك لحوالي 40 إلى 50 صورة مدخلة. وهذا يعني أن دقة قيم مواضع الصور أفضل، رغم أن المسار لم يحوي أي مطبات أو اهتزازات عشوائية.

لذلك تبين أن أخذ التغيرات بعين الاعتبار عند اختيار الصور أدى للحصول على دقة إضافية تزيد الكثافة للغيمة النقطية. في تجارب أخرى على مجموعات الصور المأخوذة من المحاكاة بما يتعلق بعملية SBA تم إضافة التغيرات المقدر لمساقط النقاط بالطريقة المذكورة سابقاً. وتم استنتاج عدم حصول زيادة بالدقة للنواتج.

هذا يدل على أن قيم التغيرات المقدر تلك غير دقيقة كفاية، لأن طريقة اشتقاقها تقريبيه من جهة، ولأن هناك عوامل أخرى مؤثرة عليها من جهة أخرى.

فهناك حد معين للخطأ بقيم التغيرات المقدر، نتلاشى بعده الفائدة من إضافتها كما تم التوضيح في [22] .

6 خاتمة وعمل مستقبلي:

تم الاستنتاج هنا أن إدخال التغيرات لمواضع الكاميرات كعامل عند اختيار الصور أدى لزيادة بكثافة الغيمة النقطية الناتجة من تلك الصور. ولكن يمكن كتحسين إضافي الموازنة بين هذا العامل الجديد وعامل التوزع الهندسي المنتظم للصور بعلاقة واحدة بدل وضع شرط لفرق التغيرات .

أما لمرحلة تعديل الحزم يمكن ادخال عامل تغير إعادة البناء للنقاط لحساب تغير أدق للمساقط و ثم اختبار مدى التحسين الحاصل بمرحلة تعديل الحزم المشتت SBA.

المراجع

- [1] A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "*MonoSLAM: Real-time single camera SLAM*," IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 6, 2007, 1052 - 1067.
- [2] J. Civera, A. J. Davison and J. M. Montiel, "*Inverse depth parametrization for monocular SLAM*," IEEE transactions on robotics, vol. 24, no. 5, 2008, 932 - 945.
- [3] L. Russo, S. Rosa, B. Bona and M. Matteucci, "*A ros implementation of the mono-slam algorithm*," International Journal of Computer Science & Information Technology, vol. 6, no. 1, 2014, 339 - 351.
- [4] L. Mingyang and A. I. Mourikis, "*High-precision, consistent EKF-based visual-inertial odometry*," The International Journal of Robotics Research, vol. 32, no. 6, 2013, 690 - 711,
- [5] B. Williams and I. Reid, "*On combining visual SLAM and visual odometry*," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, 3494 - 3500.
- [6] H. Strasdat, J. M. Montiel and A. J. Davison, "*Visual SLAM: why filter?*," Image and Vision Computing, vol. 30, no. 2, 2012, 65 - 77.
- [7] J. Engel, V. Koltun and D. Cremers, "*Direct sparse odometry*," IEEE transactions on pattern analysis and machine intelligence, 2017, vol. 4.
- [8] J. Engel, T. Schops and D. Cremers, "*LSD-SLAM: Large-scale direct monocular SLAM*," in European Conference on Computer Vision, Springer, 2014, 834 - 849.
- [9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.
- [10] M. I. Lourakis and A. A. Argyros, "*SBA: A software package for generic sparse bundle adjustment*," ACM Transactions on Mathematical Software (TOMS), vol. 36, 2009, 2,.
- [11] A. Rachmielowski, N. Birkbeck, M. Jagersand and D. Cobzas, "*Realtime visualization of monocular data for 3D reconstruction*," in 2008, Computer and Robot Vision, 2008. CRV'08. Canadian Conference on.
- [12] G. Pavoni, M. Dellepiane, M. Callieri and R. Scopigno, "*Automatic selection of video frames for path regularization and 3D reconstruction*," Proceedings of the 14th Eurographics Workshop on Graphics and Cultural Heritage, 2016, 1-10.

- [13] J. Civera, O. G. Grasa, A. J. Davison and J. M. Montiel, "*1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry*," vol. 27, no. 5, 2010, 609 - 631.
- [14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [15] E. RUBLEE, V. RABAUD, K. KONOLIGE and G. BRADAKI, "*ORB: an efficient alternative to SIFT or SURF*," in 2011, IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain.
- [16] M. Muja and D. Lowe, 2013, "*Fast library for approximate nearest neighbors (FLANN)*,"
- [17] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia, "*Meshlab: an open-source mesh processing tool.*," in Eurographics Italian chapter conference, vol. 2008, 129--136.
- [18] "SBA in ROS package," [Online]. Available: <https://code.ros.org/svn/ros-pkg/stacks/vslam/trunk>.
- [19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, "*ROS: an open-source Robot Operating System, 2009*," ICRA workshop on open source software, vol. 3, no. 3.2, 5,
- [20] B. Ochoa and S. Belongie, "*Covariance propagation for guided matching*," in Proceedings of the Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP), 2006.
- [21] N. Koenig and A. Howard, "*Design and use paradigms for Gazebo, an open-source multi-robot simulator*," in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004.
- [22] M. J. Brooks, W. Chojnacki, D. Gawley and A. Van Den Hengel, "*What value covariance information in estimating vision parameters?*," Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, 2001, 302-308,