

移動型見守りロボットのための視点プランニング

Igi Ardiyanto (豊橋技術科学大学) 三浦 純 (豊橋技術科学大学)

Viewpoint Planning for a Mobile Attendant Robot

Igi Ardiyanto (Toyohashi Univ. Tech.) and Jun Miura (Toyohashi Univ. Tech.)

Abstract: This paper describes a viewpoint planning algorithm for a mobile attendant robot, which always watches a specific person, in indoor environments. Rather than continuously following the person, the robot moves carefully selects a viewpoint considering both the topology feature of the environment and the person movement. The topology feature is extracted using a skeletonization technique to get a set of viewpoints. The robot searches for *escaping gaps* from which the person may get out of the robot's sight, and plans its action based on the *geodesic motion model*. Simulation results using 3D simulator are provided to show the effectiveness and feasibility of the algorithm.

1 はじめに

家庭やオフィスで日常生活をサポートしてくれるサービスロボットへの期待が高まっている。そのようなロボットのの一つとして、ここでは移動型の見守りロボットを取り上げる。このロボットは、自由に歩き回る人物を常に視野に入れるように自らの位置を移動させる。空間形状の解析に基づき、有効な視点の集合を計算するとともに人物の将来の移動を予測し、見失うことなく、しかもできるだけ無駄な移動を抑えるような視点間移動を計画する。

2 関連研究

2.1 美術館問題

美術館問題 (art gallery problem) は閉空間を監視する警備員の数を最小化する配置を求める問題であり [1, 2], 本論文で扱う見守りロボット計画問題と関係が深い。通常美術館問題では複数の静的な警備員を想定するのに対し、本論文では1台の見守りロボットが複数の視点 (静的な警備員のいる場所と考えることもできる) の間を動的に移動する点が異なる。

2.2 追跡逃亡問題

追跡逃亡問題 (pursuit-evasion problem) は、追跡者 (pursuers) が逃亡者 (evaders) を捕えるための行動を計画する問題であり、多くの研究が行われている [3, 4, 5]。通常追跡逃亡問題では、逃亡者は常に追跡者から逃れようと移動し、追跡者の目的は最終的に逃亡者を捕えることであるが、本論文では、対象人物はロボットと独立に移動し、またロボットは対象人物を常に視野に捉えることを目的とする。

2.3 人物追従ロボット

われわれは複雑環境下で人物追従を行うロボットを開発している [6, 7]。人物追従によりある程度の見守りは可能であるが、ただ追従するだけでは角を曲がる場合などに一時的に視野から外れることは避けられない。本論文では空間構造の解析により、そのような状況が生じないようなロボットの位置取りを自動的に計画する手法を提案する。

3 骨格化に基づく視点候補生成

視点計画問題を、オフラインで視点候補を生成するステップとオンラインで視点を選択するステップに分けて考える (Fig. 1 参照)。

見守り問題において、例えば、廊下の曲がり角や交差点はさまざまな方向を観測でき視点として有用だが、部屋のコーナーの有

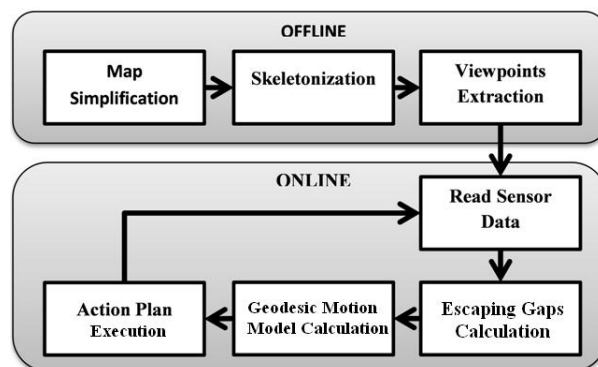


Fig. 1: Overview of viewpoint planning algorithm: action for extracting viewpoint in offline stage (top), viewpoint planner execution in online stage (bottom).

用性は低い。そこで、自由空間の骨格化 (skeletonization) によって得たトポロジ情報を利用し、視点候補を抽出する。

3.1 空間、ロボット、人物の表現

対象環境の地図 \mathbb{C} が与えられているとする。地図中の自由空間を \mathcal{F} 、障害物領域を \mathcal{N} とする。SLAM などで得られた地図 \mathbb{C} を以下の手順で簡単化する。

- 2 値化: $p_{x,y}$ を \mathbb{C} 内の位置とすると、2 値化地図 B は以下のように得られる:

$$B(p_{x,y}) = \begin{cases} 1 & \text{for } p_{x,y} \in \mathcal{F} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

- 境界抽出: Suzuki らの境界抽出アルゴリズム [8] により、自由空間の外側の境界 Γ と自由空間内の障害物の境界 Γ_{H_i} ($i = 1, 2, \dots, n$) (n は障害物の数) を抽出する。
- 境界の多角形近似: Douglas-Peucker アルゴリズム [9] により境界 Γ と Γ_{H_i} を多角形近似したものをそれぞれ Φ , H_i とする。 Φ の内側の点の集合を $int(\Phi)$, H_i の外側の点の集合を $ext(H_i)$ とすると、自由空間 $\mathcal{F} \subset \mathbb{C}$ は以下の式で再定義できる:

$$\mathcal{F} = \{\forall p_{x,y} | p_{x,y} \in \{int(\Phi) \cap ext(H_i)\}\}. \quad (2)$$

ロボットの状態はベクトル $\mathbb{R} = \{x_r, y_r, \theta_r, v_r, w_r\}$ で表す。ここで、 (x_r, y_r, θ_r) はロボットの位置と向き、 v_r, w_r はそれぞれロ

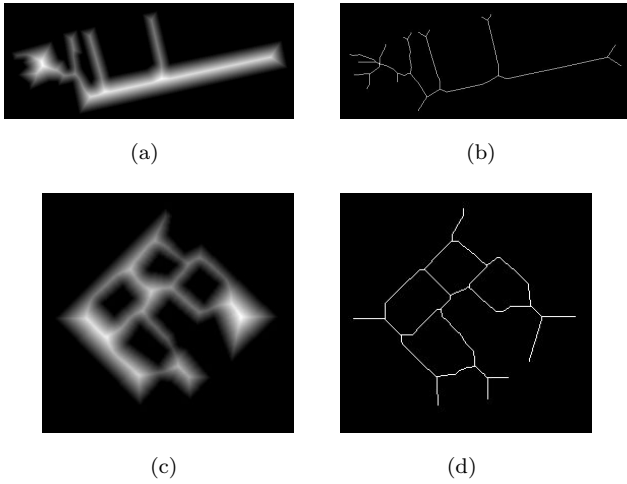


Fig. 2: Map skeletonization: Distance Transform Map (left) and final skeleton map result (right).

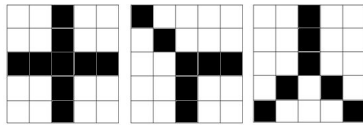


Fig. 3: Examples of templates (5x5 cells).

ロボットの並進速度, 回転角速度を表す. ロボットは自動焦点のパンチルトカメラを備えているとし, $\varphi = \{\varphi_{pan}, \varphi_{tilt}, \varphi_{zoom}\}$ でパン, チルト, ズーム位置を表す. また, 見守り対象人物は, 位置及び速度のベクトル $\mathcal{O} = \{x_o, y_o, v_{x_o}, v_{y_o}\}$ で表す.

3.2 骨格化

自由空間地図に距離変換を適用し, 以下の地図 \mathcal{D} を求める:

$$\mathcal{D}(p_{x,y}) = \begin{cases} \sqrt{(x-x_2)^2 + (y-y_2)^2} & \text{for } (x,y) \in \mathcal{F} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

ここで (x_2, y_2) は (x, y) に最も近い障害物領域である. \mathcal{D} にラプラスアン \mathcal{K}

$$\mathcal{K}(p_{x,y}) = \frac{\partial^2 \mathcal{D}}{\partial x^2} + \frac{\partial^2 \mathcal{D}}{\partial y^2}. \quad (4)$$

を適用し, さらに2値化を行うことにより, 骨格地図を得る (Figs. 2(b), 2(d) 参照).

3.3 視点候補の抽出

骨格線上の交差点や端点は有用な視点の候補となる. そこで, Fig. 3 に示すような骨格パターン (30 種類) を同時に適用し, 以下の差異度が最も小さくかつあるしきい値以下となるものを探し, 条件に当てはまる場所を視点候補とする.

$$d(x, y) = \sqrt{\sum_m \sum_n [\mathcal{W}(m, n) - \mathcal{K}(x+m, y+n)]^2} \quad (5)$$

ここで, \mathcal{W} はある交差点パターン, (m, n) は \mathcal{W} の大きさ, \mathcal{K} は骨格線地図である.

3.4 視野の重なりを考慮した視点候補集合の最適化

前項の処理では, 近い場所に複数の視点候補が生成される可能性がある. そこで, 自由区間全域をカバーし, かつ最小の視点数となるような視点候補集合を求める.

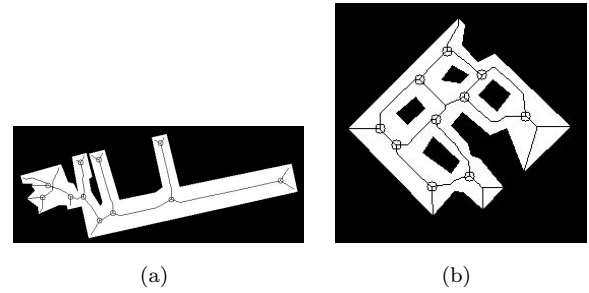


Fig. 4: Maps with skeleton and viewpoints. Small circles denotes viewpoints.

ある視点 $P_i (i = 1, 2, \dots, k)$ から見える範囲 (視野) を $V(P_i)$ とすると, 視点集合最小化問題は以下のように定式化できる:

$$f(k) = \bigcup_{i=0}^k \text{int}(V(P_i)) \quad (6)$$

$$\arg \min_k f(k) \cong \mathcal{F}. \quad (7)$$

式 (7) を以下の手順で解く. まず, 端点の視点候補のうち, より広い視野を持つものから順に調べていき, その点を候補集合から削除しても条件 $f(k) \cong \mathcal{F}$ が満たされる場合に削除する. 次に, 同様のチェックを交差点の視点候補についても行う. この手順で得られた k を最終的な視点候補の数とする. Fig. 4 に得られた視点候補集合の例を示す.

ロボットの計測範囲が限られている場合には, 上記の手順を行った結果, $f(k) < \mathcal{F}$ となる場合がある. この場合には, カバーされていない領域について, 骨格化と視点候補抽出を繰り返すことにより, $f(k) \cong \mathcal{F}$ が満たされるようにする.

4 オンライン視点選択

見守りロボットの視点プランニングの目的は, 対象人物を視野から外すことなくロボットの移動を最小化することである. ロボットはカメラによる対象の自動追尾を行うこととし, 視点プランニングにおいてはロボットを中心とする全周の視野内に対象人物が入るかどうかなどだけを考慮する. まず, 人物が視野から外れる境界 (逃亡境界 (escaping gap) と呼ぶ) を計算し, そこから人物が出ていく時間を予測して, ロボットの視点を決定する.

4.1 逃亡境界の計算

逃亡境界は, 人物が視野から外れる可能性のある点の集合であり, 未知空間探索における“フロンティア”[10] に類似した概念である.

各視点候補についての視野 $V(P_i)$ と同様に, ロボット位置 (x_r, y_r) で定まる視野 $V((x_r, y_r))$ を考える. ここで, 視野 V の境界上の点の集合を $b(V)$ とすると, 逃亡境界 λ は $b(V)$ のうち, 自由空間の内外の境界 Φ と H_i に属さない点の集合として定義できる:

$$\lambda = \{p_{x,y} | p_{x,y} \in b(V((x_r, y_r))) \wedge \neg(p_{x,y} \in b(\Phi) \vee p_{x,y} \in b(H_i))\} \quad (8)$$

Fig. 4(a) の環境で, 逃亡境界を計算した例を Fig. 5 に示す.

4.2 測地線移動モデル

見守り問題においては, 対象人物はロボットと独立に移動し特にロボットから逃れようとする意図はないが, 視点プランニング

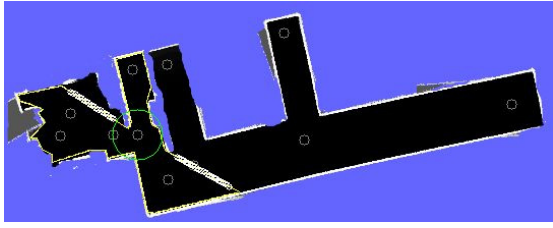


Fig. 5: Escaping Gaps. The small circle inside the green circle is the robot position and other small circles are viewpoints. The bold white line in the middle of free space (black area) is escaping gaps. Yellow lines denotes the visibility polygon.

においては最悪シナリオを想定する．そこで，対象人物は常に最も近い逃亡境界へ向かうと想定し，そこへ到達する時間を予測する．人物やロボットの移動は環境の形状の影響を受けるので，ユークリッド距離に基づく時間予測は適切ではない．そこで，波面伝播アプローチ [11] に基づく測地線移動モデル (geodesic motion model) を利用する．

各地点 $p_{x,y}$ における速度を $J(p_{x,y})$ とし，ある位置 (現在位置) から伝播する波面が到達時間 T を次式で計算する:

$$|\nabla T(p_{x,y})| = \frac{1}{J(p_{x,y})}. \quad (9)$$

この式は Eikonal 方程式と呼ばれ，1 次の差分方程式で近似できる [11]:

$$\max\left(\frac{T(p_{x,y}) - T_1}{\Delta x}, 0\right)^2 + \max\left(\frac{T(p_{x,y}) - T_2}{\Delta y}, 0\right)^2 = \frac{1}{J(p_{x,y})^2} \quad (10)$$

ここで，

$$\begin{aligned} T_1 &= \min(T(p_{x+1,y}), T(p_{x-1,y})) \\ T_2 &= \min(T(p_{x,y+1}), T(p_{x,y-1})) \end{aligned} \quad (11)$$

であり，式 (10) の解は次式で与えられる．

$$T(p_{x,y}) = \begin{cases} T_1 + \frac{1}{J(p_{x,y})} & \text{for } T_2 \geq T \geq T_1 \\ T_2 + \frac{1}{J(p_{x,y})} & \text{for } T_1 \geq T \geq T_2 \\ \text{quadratic solution of (10)} & \text{for } T \geq \max(T_1, T_2) \end{cases} \quad (12)$$

速度関数 J は以下のように与える．

1. ロボットについては，障害物からの距離に依存した速度を，距離変換地図 \mathcal{D} (式 (3) 参照) を用いて以下のように定義する．

$$J_{robot}(p_{x,y}) = \|\mathcal{D}(p_{x,y})\|_{norm(0, v_{rmax})}. \quad (13)$$

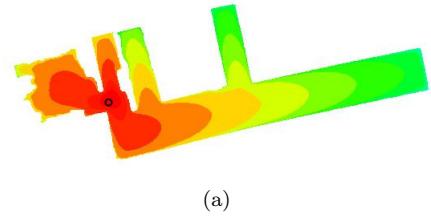
この速度関数を用いて，到達時間は Fig. 6(a) のように計算できる．

2. 人物については，現在の移動速度と障害物からの距離の両方を考慮する．まず，人物の移動領域として以下の円錐を考え

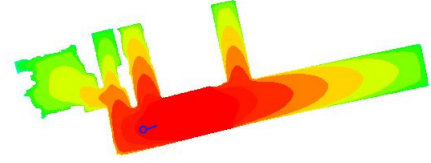
$$\mathcal{C} = \left\{ \forall p_{x,y} | p_{x,y} \in \mathcal{F} \wedge \left(\angle p_{x,y} \leq \arctan\left(\frac{v_{xo}}{v_{yo}}\right) \pm \frac{\pi}{3} \right) \right\}, \quad (14)$$

この円錐内の速度関数 J_c を以下のように定義する．

$$J_c(p_{x,y}) = \begin{cases} v_{target} & \text{for } p_{x,y} \in \mathcal{C} \\ \varepsilon & \text{for } p_{x,y} \notin \mathcal{C} \wedge p_{x,y} \in \mathcal{F} \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$



(a)



(b)

Fig. 6: The travel time map of the robot (top) and the target (bottom). The black circle represents the robot's current position. The blue circle with line denotes the target position and its predicted movement.

ここで， $v_{target} = (v_{xo}^2 + v_{yo}^2)^{\frac{1}{2}}$ であり， ε は定数である．最後に J_c と \mathcal{D} を統合して，以下の速度関数を得る．

$$J_{target}(p_{x,y}) = J_c(p_{x,y}) \|\mathcal{D}(p_{x,y})\|_{norm(0,1)}. \quad (16)$$

この速度関数で表される測地線移動モデルの利用が本論文で提案する視点プランニングの重要な点である．Fig. 6(b) に人物の到達時間分布を計算例を示す．

4.3 コスト最小化に基づく視点プランニング

最悪シナリオを考えるために，人物が最も早く到達可能な逃亡境界 $\lambda_{critical}$ を計算する:

$$\lambda_{critical} = \arg \min_i T_{target}(\lambda_i) \quad (17)$$

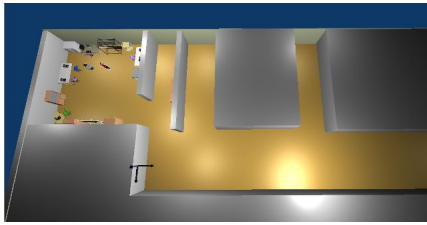
この式は，ロボットが停止していると $T_{target}(\lambda_{critical})$ で対象人物を見失うことを示している．

対象人物を常に視野に入れつつ，ロボットの移動を最小化することを考える．以下の記号を定義する．

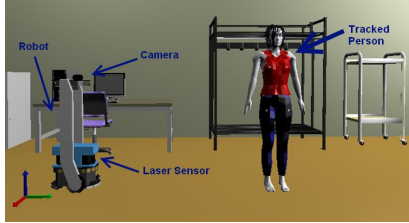
- $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$: 視点集合．
- $\mathcal{S} = \{s_0, s_1\}$: 状態集合． s_1 はロボットが視点上にいる状態， s_0 は視点間を移動している状態とする．
- $\mathcal{A} = \{a_0, a_{P_1}, \dots, a_{P_k}\}$: 動作集合． a_0 は何もしない行動 (止まったまま) を， a_{P_i} は視点 P_i への移動行動を表す．

視点選択にあたり，以下の3つの規則を適用する．

- 規則 1: 動作 a_0 は状態 s_1 にも適用可能．すなわち，ロボットは視点においてのみ停止できる．
- 規則 2: $\lambda_{critical}$ が視野に入らない視点への移動は行わない．
- 規則 3: $\lambda_{critical}$ を見ることができ，かつ最も近い視点を $P_{critical}$ とする．対象人物が $\lambda_{critical}$ に到達する時間を \mathbb{T}_{target} ，ロボットが $P_{critical}$ に到達する時間を \mathbb{T}_{robot} とする．する．ロボットが余裕を持って $P_{critical}$ に到達できる時，すなわち $\mathbb{T}_{robot} \leq 0.8\mathbb{T}_{target}$ のとき，すべての動作が実行可能．時間的余裕がない場合，すなわち $0.8\mathbb{T}_{target} \leq \mathbb{T}_{robot} \leq \mathbb{T}_{target}$ のとき， $a_{P_{critical}}$ のみが実行可能とする．

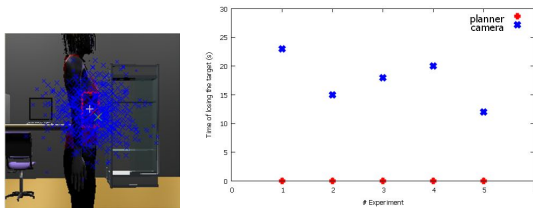


(a)



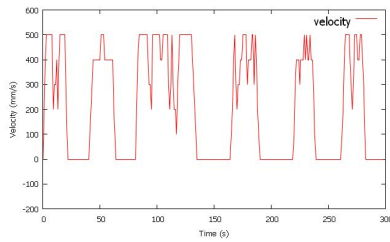
(b)

Fig. 7: 3D simulator appearance: (a) top view of the environment, (b) robot and target appearances.



(a)

(b)



(c)

Fig. 8: Result of guard robot simulations: (a) Tracked person, (b) Time of losing the target, (c) Velocity profile of the robot.

そして、上記規則で実行可能な動作のうち、コスト最小のものを選んで実行する。

このプランニングの結果として、ロボットは以下のような行動を示す。

- ロボットがある視点上にいるとき、規則 3 で時間に余裕があると判断されるときには、その場所に留まる。
- ロボットが視点間を移動しているときには、次の視点に到達するまで動き続ける。

以上の視点プランニングによって視点が選択され、それが現在位置と異なる場合には、局所経路計画 [7] を行って移動動作を生成する。

5 シミュレーション実験

高機能シミュレータ MORSE と RT ミドルウェアを利用したシミュレーション環境 [12] で実験を行う (Fig. 7 参照)。距離センサ

とカメラを備えたロボットを用い、SLAM による地図生成と人物追跡 (Fig. 8(a) 参照) を行う。Fig. 8(b) に、5 回行った実験それぞれについて、カメラの視野から外れた時間の割合と、ロボット位置を中心とする視野領域から外れた時間の割合を示す。シミュレータ上のパンチルトカメラによる追跡の制御の遅れから、カメラ視野からは外れる時間は存在するが、常に対象人物はロボットから見て陰になる位置にはいないことが分かる。

Fig. 8(c) は 1 回の実験中のロボットの速度変化を示す。止まっている時間が長いほどエネルギー消費が少なく効率的であることを示す。以下のエネルギー消費式を用いて、通常的人物追従アルゴリズム [7] と効率を比較した。

$$Energy = \sum_{k=0}^T \frac{1}{2} m v_k^2 + \sum_{k=0}^T \frac{1}{2} I \omega_k^2 \quad (18)$$

ここで、 T は総移動時間、 m 、 I はそれぞれロボットの質量および慣性モーメント、 v 、 ω はそれぞれ並進速度、回転角速度である。その結果、人物追従では 724.50 [J]、視点プランニングでは 561.75 [J] となり、プランニングによる効率化の効果が示された。

6 おわりに

移動型見守りロボットのための視点プランニングアルゴリズムを提案し、シミュレーション実験でその有効性を確認した。環境のトポロジ情報を自由空間の骨格化によって解析して視点候補を抽出し、追跡可能性を考慮した規則とコスト最小化により、対象を見失うことなく効率的に見守る行動が実現できた。3 次元の可視性の考慮や実ロボットによる実験が今後の課題である。

参考文献

- [1] J. O'Rourke. "Art Gallery Theorems and Algorithms". Oxford University Press, 1987.
- [2] S.K. Ghosh. "Approximation algorithms for art gallery problems", *Proc. Canadian Information Processing Society Congress*, pp. 429-434, 1987.
- [3] L. Guibas, D. Lin, J.C. Latombe, S. LaValle, R. Motwani. "Visibility-based pursuit evasion in a polygonal environment". *Int. J. of Computational Geometry Applications*, 2000.
- [4] L. Guilamo, B. Tovar, and S.M. LaValle. "Pursuit-evasion in an unknown environment using gap navigation trees". *Proc. IROS-2004*, 2004.
- [5] B.P. Gerkey, S. Thrun, and G.J. Gordon. "Visibility-based Pursuit-evasion with Limited Field of View". *IJRR*. Vol. 25, No. 4, pp. 299-315. 2006.
- [6] J. Miura et al. "Development of a Person Following Robot and Its Experimental Evaluation". *Proc. IAS-2010*, pp. 89-98, 2010.
- [7] I. Ardiyanto and J. Miura, "Heuristically Arrival Time Field-Biased (HeAT) Random Tree: An Online Path Planning Algorithm for Mobile Robot Considering Kinodynamic Constraints". *Proc. ROBIO-2011*, pp. 360-365, 2011.
- [8] S.Suzuki, K.Abe. "Topological structural analysis of digital binary image by border following". *CVGIP*, Vol. 30, No. 1, pp. 32-46, 1985.
- [9] D.H. Douglas and T.K. Peucker. "Algorithms for the reduction of the number of points required to represent a line or its caricature". *The Canadian Cartographer*, Vol. 10, pp. 112-122, 1973.
- [10] J.W. Durham, A. Franchi, F. Bullo. "Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration". *Proc. ICRA-2010*, pp. 3562-3568, 2010.
- [11] M.S. Hassouna, A.E. Abdel-Hakim, and A.A. Farag. "PDE-based robust robotic navigation". *Image and Vision Computing*, Vol. 27, pp. 10-18, 2009.
- [12] I. Ardiyanto, Y. Okada, and J. Miura. "RT components for using MORSE realistic simulator for robotics". SI-2012, 2012.