The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Navigation Framework for Humanoid Robots Integrating Gaze Control and Modified-univector Field Method to Avoid Dynamic Obstacles

Jeong-Ki Yoo and Jong-Hwan Kim

Abstract— **This paper proposes a navigation framework for humanoid robots, which integrates gaze control and modified univector field-based path planning to cope with moving obstacles. To make navigation robust, obstacles are modeled according to their relative velocities and positions. Moreover, partial evaluation values for gaze control architecture are also considered for modifying their virtual size and moving trajectory. In addition, gaze control architecture is proposed, which estimates the size of local map confidence area, self-localization error, surrounding obstacles and obstacle-free distance against those obstacles in the local map. The proposed framework is verified through computer simulations by using a developed simulator for HanSaRam-VIII.**

I. Introduction

Most of humanoid robot researches have focussed on walking issues [1], [2]. From the biologically inspired approach to the dynamic model-based approaches [3], [4], they have focussed on generating the dexterous walking pattern. Due to the remarkable improvements in hardware and walking pattern generation for humanoid robots, researches are now expanding to various other fields of robotics, such as navigation, vision perception, task processing in complex environments, etc [5]–[7].

In the navigation aspect, various navigational concepts for wheeled robots, such as heuristic search algorithm, dynamics-based random state search approach and force-based algorithm, have been applied to humanoid robots. A* algorithm was applied to footstep planning [8], and vision-guided footstep planning in dynamic environment was performed [9]. Among various search algorithms, rapidly-exploring random trees approach was also applied for motion planning of humanoid robots [10]. Since it estimates and explores through sampled states maintaining its dynamic constraints, it can rapidly derive quasi-optimal path. Univector field method using virtual obstacle concept for footstep planning was proposed, which is expanded in this paper [20]. Integration with vision processing algorithms was performed in many researches of humanoids, which focussed on increasing autonomy and obstacle avoidance [11].

Along with these approaches, researches related to gaze control are also important not only for stabilizing vision images, but also for efficient navigation [12]. It plays a major role in gathering surrounding information [13]. Information theory-based approach was proposed in [14]. Even though these are closely dependent on path planning issues, integration aspect has not been considered as a broadly concerned research topic so far.

The authors are with the Department of Electrical Engineering, KAIST, 335 Gwahangno (373-1 Guseong-dong), Yuseong-gu, Daejeon 305-701, Republic of Korea (e-mail: {jkyoo, johkim}@rit.kaist.ac.kr).

This paper proposes a navigation framework, which integrates gaze control and modified univector field method [20]. Since the univector field method is robust for real-time applications, it is modified to deal with moving obstacles efficiently by adopting the proposed dynamic virtual obstacle and velocity modification scheme. Instead of using a simple duplicated virtual obstacle, dynamically resizing and moving obstacle integrated with partial evaluation functions for gaze control is proposed. In addition, its move-to-goal univector field function is modified to pass by via points and arrive at a goal position with arbitrary arriving posture angle. Gaze control system integrates four partial evaluation functions for local map confidence area size, self-localization error from covariance matrix, environmental status of obstacles, and obstacle-free distance of local map. These are also used for deriving the size and length of the path of virtual obstacle. Through this interlinked structure, the performance of the whole framework is expected to have synergy effects for avoiding collision with moving obstacles. Verification of the proposed scheme is performed through computer simulations with a model of a small-sized humanoid robot, HanSaRam-VIII (HSR-VIII), developed at the Robot Intelligence Technology (RIT) Laboratory, KAIST since 2000 [15].

This paper is organized as follows. In Section II, gaze control architecture focusing on partial evaluation functions is described. Section III explains the modification scheme for univector field method using dynamic virtual obstacle and velocity modification approach. Section IV describes an integration of the proposed framework. Section V presents simulation environment and results, and finally conclusions follow in Section VI.

II. Gaze Control Architecture

This section describes the gaze control architecture based on four types of partial evaluations and their integration scheme.

A. Partial Evaluations for Gaze Control

In case of navigation in a complex environment, various types of information, such as relative distances and speed of obstacles relative to the robot, error covariance matrix of all objects in the local map, the state of the robot, etc., have to be combined efficiently. In order to implement efficient gaze control architecture using these informations, four types of partial evaluation functions are defined focusing on self-localization uncertainty factor, relative position and velocity of obstacles, obstacle-free distance of surrounding local map and confident area ratio against the unconfident one in its local map. In addition, these are closely related to the path generation scheme which is described in the next section. The following is the detailed description of the four partial evaluation functions.

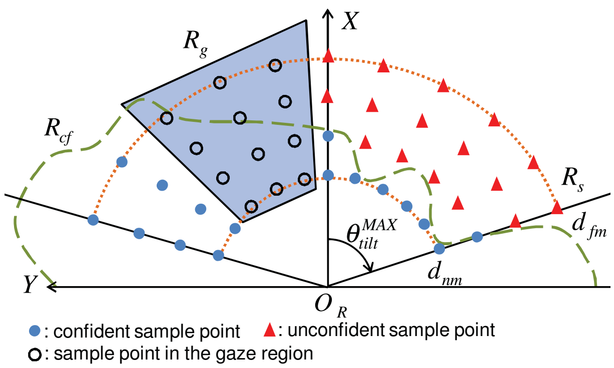


Fig. 1. Local map sample point diagram. The long dashed line depicts the confident area, R_{cf}.

1) Local Map Confidence Area-based Partial Evaluation (ψ_{lm}): Fig. 1 shows a sample point diagram in robot coordinate O_R. Apart from the probability of occupancy grid map in [16], confidence factor $\mathcal{C}(\cdot)$ is additionally defined for each grid. It is set to one when a grid is included in gaze region R_g, and starts to be attenuated by ξ_{lm} after the grid is excluded from the gaze region. In the extended map, local map confidence area-based partial evaluation is defined as

$$\mathcal{M}_{lm}(R_g) = N_{lp}/N_{cp}, \tag{1}$$

with

$$N_{lp} = n(\{\mathbf{p}_i^s | \mathcal{C}(\mathbf{p}_i^s) < \lambda_{cf}, \mathbf{p}_i^s \in R_{s-g}\}),$$
$$N_{cp} = n(\{\mathbf{p}_i^s | \mathbf{p}_i^s \in R_{s-g}\}) - N_{lp}, \quad R_{s-g} = R_s \cap R_g^c,$$

where R_g is the gaze region, R_s is the sampling region defined by $(d_{nm}, d_{fm}, \theta_{tilt}^{MAX})$, and $\mathbf{p}_i^s$ denotes the i_{th} sample point in the grid map. In this paper, (d_{nm}, d_{fm}) are set to (30 cm, 130 cm), respectively. R_g is derived by the maximum pan and tilt angles $(\theta_{pan}^{MAX}, \theta_{tilt}^{MAX})$. For λ_{cf} and ξ_{lm}, 0.85 and -0.001 are respectively used in this paper. This partial evaluation is defined by the ratio between confident and under-confident sample points as in (1). The candidate pan/tilt angles for ψ_{lm} are derived according to arithmetic mean of representative angles for under-confident sample points.

2) Self-localization-based Partial Evaluation (ψ_{sf}):

$$\mathcal{M}_{sf}(\sigma) = \begin{cases} \sigma/\mathcal{T}_{sf} & \text{if } \sigma \leq \mathcal{T}_{sf} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{T}_{sf}$ denotes the user-defined threshold. For self-localization, a model-based sample point approach for extended kalman filter, i.e. distribution approximation filter, is adopted [17]. As a result, error covariance matrix C_{pb} and its magnitude σ of robot posture are obtained. Using σ, partial evaluation for self-localization uncertainty ψ_{sf} is calculated by (2). Apart from the other already explained two evaluation functions, ψ_{lm} and ψ_{sf}, candidate angles for ψ_{ob} and ψ_{sf} are defined for all the detected obstacles in the confident area of the local map, and added up in the process of integration in Section II-B. This is because they vary depending on the number of obstacles located within the gaze area.

3) Obstacle-based Partial Evaluation (ψ_{ob}): Surrounding obstacles also needs to be considered in navigation. In particular, the movement of each detected obstacle in the local map has to be considered [13]. During the navigation process, each detected obstacle is updated in the local map. Since humans mainly pay attention to frontal and approaching obstacles, only frontal and approaching-from-behind obstacles in the robot-centered confidence map, ${}^R\mathcal{O}^{cm}_{dted|bappr}$, are estimated through their relative positions and velocities in O_R as follows:

$$\mathcal{M}_o(\alpha_{o_i}) = \begin{cases} \alpha_{o_i}/\mathcal{T}_o & \text{if } \alpha_{o_i} \leq \mathcal{T}_o \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

with

$$\alpha_{o_i} = \omega_o \sigma_{o_i} \|{}^R\mathbf{v}_{o_i}\|_2 / \|{}^R\mathbf{p}_{o_i}\|_2,$$

where ${}^R\mathbf{p}_{o_i}$ and ${}^R\mathbf{v}_{o_i}$ denote the position and velocity of i_{th} obstacle, respectively, and ω_o is a normalization factor. For this evaluation, localization uncertainty factor of i_{th} obstacle is reflected by σ_{o_i} obtained in the process of localization for ψ_{sf}.

4) Way point-based Partial Evaluation (ψ_{way}): The partial evaluation function for the closest way point is defined using an obstacle-free distance, d_{of}, as follows:

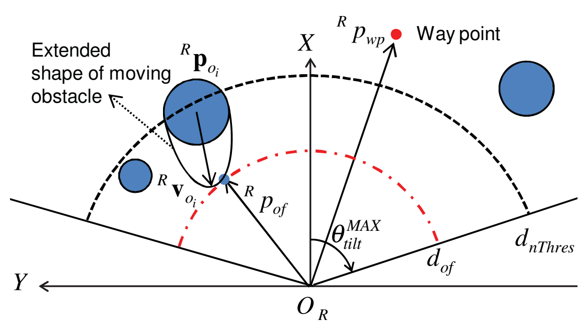


Fig. 2. Obstacle-free distance diagram with a moving obstacle.

$$\mathcal{M}_{way}(d_{of}) = \begin{cases} d_{of}/d_{nThres} & \text{if } d_{of} \leq d_{nThres} \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where d_{of} is the minimum distance without obstacles from robot bounded by a user-defined value d_{nThres}. It is obtained by the closest point ${}^R\mathbf{p}_{of}$ of extended shapes of obstacles in ${}^R\mathcal{O}^{cm}_{dted|bappr}$. The extended shape of moving obstacles is defined and used for the navigation process in Section III according to ${}^{R}\mathbf{p}_{o_i}$, ${}^{R}\mathbf{v}_{o_i}$ and gaze control parameters. The candidate pan/tilt angle for ψ_{way} is the direction of way point in O_R.

B. Integration of Partial Evaluations (ψ_{int})

One candidate gaze direction can be converted to two dimensional pan/tilt gaze angle region. If the pan/tilt angles are set to any of the gaze angles in the derived region, the original candidate gaze direction is shown on the captured image. The conversion for partial evaluations, $\alpha \in \{lm, way, sf, ob\}$, are defined as

$$R_\alpha(\boldsymbol{\theta}_i) = \begin{cases} \psi_\alpha & \text{if } \theta_i \in R_g^{\psi_\alpha}(\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\hat{\psi}_\alpha$ and $\hat{\boldsymbol{\theta}}_\alpha = [\hat{\theta}_\alpha^p, \hat{\theta}_\alpha^t]^T$ mean the derived partial evaluation value and its candidate pan/tilt angles. For target gaze angle $\boldsymbol{\theta}_i$, $R_g^{\psi_\alpha}$ represents the gaze region according to $\hat{\boldsymbol{\theta}}_\alpha$ as depicted in Fig. 1. The obtained gaze angle regions are combined by weighted arithmetic mean over a gaze angle integration map $\mathcal{GM}$, defined as

$$\begin{aligned} \mathcal{GM}(\boldsymbol{\theta}_i) = \omega_{lm} R_{lm}(\boldsymbol{\theta}_i) + \omega_{way} R_{way}(\boldsymbol{\theta}_i) + \\ \frac{1}{N_o}\sum_{i=1}^{N_o}\{\omega_{sf} R_{sf}(\boldsymbol{\theta}_i) + \omega_{ob} R_{ob}(\boldsymbol{\theta}_i)\}, \end{aligned} \tag{6}$$

where N_o is the number of ${}^{R}\mathcal{O}_{dted|bappr}^{cm}$, and ω_{lm}, ω_{way}, ω_{sf}, and ω_{ob} are normalization weights for partial evaluation values. In case of ψ_{sf} and ψ_{ob}, they are added up for ${}^{R}\mathcal{O}_{dted|bappr}^{cm}$ and θ_i. Due to the discrete trait of R_α, $\mathcal{GM}$ is formed in step shape. The highest center point of obtained $\mathcal{GM}$ is selected as a final representative gaze angle, $\boldsymbol{\theta}^*$, (see Fig. 7(b)).

III. Modified Univector Field Method for Dynamic Obstacle Avoidance

This section describes a modification scheme of univector field method considering moving obstacles by integrating partial evaluations, virtual obstacle concept and velocity modification scheme.

1) Modification of Univector Field Method: Due to the simplicity of univector field navigation method, it can be used for real-time control of mobile robots [18]. Compared to conventional potential field methods, it is robust to oscilations. It employs move-to-goal univector field (MUF) and avoid-obstacle univector field (AUF), respectively, for leading robot to the goal and for avoiding collision against obstacles. A robot can be modeled as a point if its maximum radius is added to those of obstacles [19]. Then, path planning can be simplified as a line searching problem. Moreover, it can reflect the result of partial evaluations, which are closely related to path planning and obstacle avoidance. The extended radius of obstacle S_{oe} is defined as

$$S_{oe} = S_r + S_o + \omega_{soe}(\eta_{vo}\|{}^{R}\mathbf{v}_{o_i}\|_2 + \eta_{imp}(\psi_{sf} + \psi_{obs})), \tag{7}$$

where S_r and S_o are the outer shapes of robot and obstacle simplified to cylinders. η_{vo} and η_{imp} are the normalization factors, respectively, and ω_{soe} is the predefined control coefficient.

MUF is modified to control its approaching direction to the goal by adopting a controller concept within a user-defined distance d_{con} as follows:

$$\mathbf{u}_{muf}(\mathbf{p_r}) = \begin{cases} [\;-c(\theta_{\mathbf{rd}}^m)\;\; -s(\theta_{\mathbf{rd}}^m)\;\; \theta_{\mathbf{rd}}^m]^T & \text{if } d_{con} < \rho_{rd} \\ [-k_\rho c(\theta_c^m) - k_\rho s(\theta_c^m)\;\; \theta_c^m]^T & \text{if } \rho_{rd} \leq d_{con} \end{cases} \tag{8}$$

with

$$\begin{aligned} &\theta_{\mathbf{rd}}^m = \angle(\mathbf{p_d} - \mathbf{p_r}), \\ &\alpha = \theta_{\mathbf{rd}}^m - \theta_{\mathbf{r}}, \beta = \theta_{tg} - \theta_{\mathbf{rd}}^m, \\ &\theta_c^m = k_\alpha \alpha + k_\beta \beta, \end{aligned}$$

where $\cos(\cdot)$ and $\sin(\cdot)$ are abbreviated to $c(\cdot)$ and $s(\cdot)$, $\mathbf{p_r}$ and $\mathbf{p_d}$ are the positions of robot and destination, respectively, and ρ_{rd} is their Euclidian distance. $\theta_{\mathbf{r}}$ is the current posture angle of robot and θ_{tg} is the target posture angle of robot at $\mathbf{p_d}$. k_ρ is the user-defined parameter to control the magnitude of MUF. If robot just passes through a via point, k_ρ is set to one. By adjusting arriving control gains, $k_\rho, k_\alpha, k_\beta$, the curvature of arriving to the goal is determined.

In case of AUF, the hyperbolic spiral univector field $\varphi_h(\mathbf{p_r})$ is defined as

$$\mathbf{u}_{auf}(\mathbf{p_r}) = \begin{cases} [c(\varphi_h(\mathbf{p_r}))\;\; s(\varphi_h(\mathbf{p_r}))] & \text{if } S_{oe} < \rho_{ro} \leq S_{oe} + d_e + d_b, \\ & \rho_{rg} + S_{oe} \geq \rho_{go}, \\ & \text{and } \mathbf{p_{ro}} \cdot \mathbf{p_{rd}} > 0 \\ [0\;\; 0]^T & \text{otherwise} \end{cases} \tag{9}$$

with

$$\varphi_h(\mathbf{p_r}) = \begin{cases} \theta_{\mathbf{or}}^i \pm \frac{\pi}{2}\sqrt{\frac{\rho_{or} - S_{oe}}{d_e}} & \text{if } S_{oe} \leq \rho_{or} < S_{oe} + d_e \\ \theta_{\mathbf{or}}^i \pm \frac{\pi}{2}(2 - \frac{S_{oe} + d_e + K_r}{\rho_{or} + K_r}) & \text{if } S_{oe} + d_e \leq \rho_{or} \end{cases}$$

where $\mathbf{p_o}$ and $\theta_{\mathbf{or}}$ denotes the position of obstacle and angle between the robot and the i $_{th}$ obstacle, respectively. d_e is the predefined radius offset used for defining spiral, d_b is the size of the boundary offset of AUF, and K_r adjusts the contour curvature of spiral as depicted in Fig. 3. $\pm$ sign denotes the direction of avoiding obstacle, where $+$ means counter clockwise and vice versa. When $(\hat{\mathbf{p}}_{\mathbf{od}} \times \hat{\mathbf{p}}_{\mathbf{or}}) \cdot [0\ 0\ 1]^T < 0$, $+$ sign is selected in $\varphi_h(\cdot)$, where $\hat{\mathbf{p}}_{\mathbf{od}} = \hat{\mathbf{p}}_{\mathbf{d}} - \hat{\mathbf{p}}_{\mathbf{o}}$ and $\hat{\mathbf{p}} = [\mathbf{p}\ 0]$.

The total univector field at $\mathbf{p_r}$ is calculated by accumulating $\mathbf{u}_{auf}$ and $\mathbf{u}_{muf}$ as follows:

$$\mathbf{u}_T(\mathbf{p_r}) = \omega_{muf}\mathbf{u}_{muf}(\mathbf{p_r}) + \omega_{auf}\bar{N}\Big(\sum_{i=1}^{n}\mathbf{u}_{auf_i}(\mathbf{p_r})\Big), \tag{10}$$

where n is the total number of obstacles, ω_{muf} and ω_{auf} are the normalizing weighting factors, and $\bar{N}(\cdot)$ is the normalization function.

2) Dynamic Virtual Obstacle and Velocity Modification Scheme: Virtual obstacle concept is already used for various navigation methods, such as a potential field method, limit-cycle navigation, univector field method, etc [20]–[22]. However, these are not appropriate to deal with various traits of dynamic obstacles because they used simple duplicated

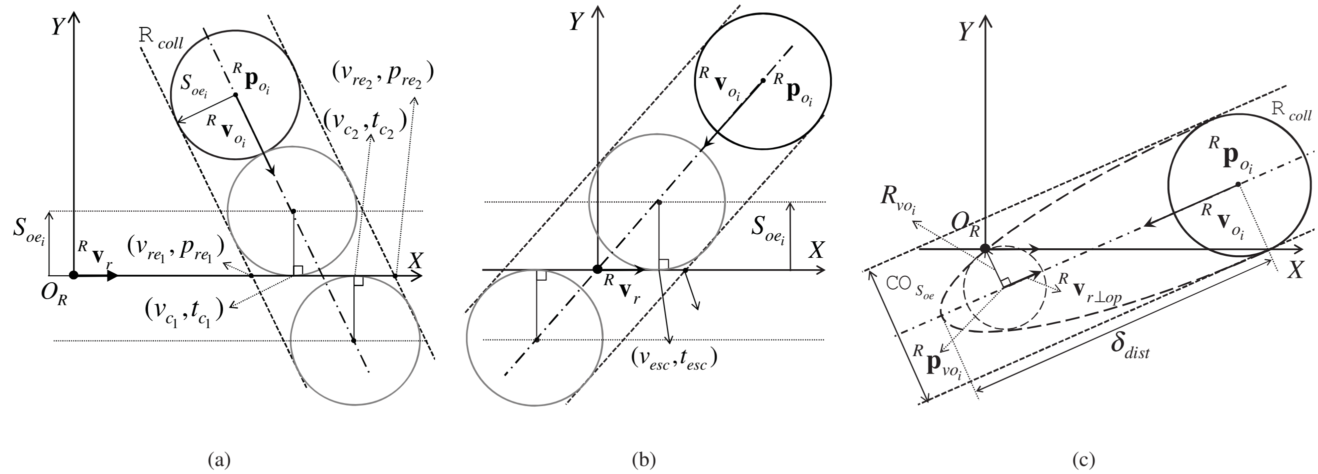


Fig. 4. Situational diagram for collision avoidance of moving obstacles. (a) Velocity modification capable case. (b) Escaping R_{coll} case. (c) Definite collision case (dynamic virtual obstacle). Thick black-lined circle depicts a moving obstacle whose radius is extended to S_{oe}, thin one depicts their collision cases, and long-dashed circle in (c) shows a dynamic virtual obstacle. Robot is simplified to the origin of O_r by adding its radius to obstacle's.

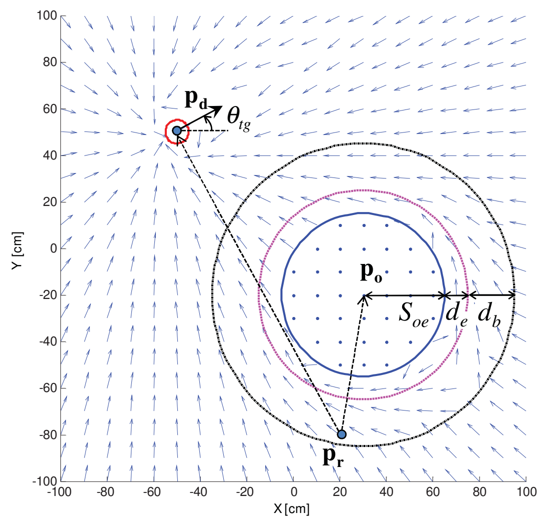


Fig. 3. Integrated univector field. The MUF and AUF values for equally spaced starting points are summarized along with the target angle θ_{tg}, $\pi/6$. In this paper, ω_{muf} and ω_{auf} are set to 0.75 and 0.25, respectively.

virtual obstacle model placed on the path of moving obstacles. Therefore, a dynamic virtual obstacle concept having extended shape and path according to relative velocity and position to robot is proposed. In addition, for efficient collision avoidance, a velocity modification scheme according to collision checking result is proposed.

Dynamic virtual obstacle: By modeling a robot as a point, the collision between a robot and surrounding obstacles can be checked by solving a second order polynomial as suggested in [19] under the assumption of constant velocities of robot and obstacles. Fig. 4 shows their relative situations and a collision band, a zone swept by the obstacle moving along its direction of velocity. As shown in Fig.4(c), if a robot is located in the collision band R_{coll} of moving obstacles, and the extended shape of approaching obstacle overlaps x-axis of O_R, the robot cannot avoid the collision simply by adjusting its velocity maintaining the direction of movement. Therefore, it has to modify its path according to the velocity and direction of approaching obstacles. If a simply duplicated virtual obstacle scheme is used for this case, robot has to steer sharply when it encounters the virtual obstacles instantly generated by moving obstacles. In order to cope with this case, the virtual obstacle concept needs to be modified to adjust its size and position dynamically according to $({}^R\mathbf{p}_{o_i}, {}^R\mathbf{v}_{o_i})$ as in Fig.4(c). The radius of dynamic virtual obstacle, R_{vo_i} and its velocity, ${}^R\mathbf{v}_{r\perp op}$ are defined as

$$\begin{aligned} R_{vo} &= \min\left(\frac{\rho^{(1-h_{vo})}-1}{\rho-1} S_{oe}, \|{}^R\mathbf{p}_{vo_i}\|_2\right), \\ {}^R\mathbf{v}_{r\perp op} &= ({}^R\mathbf{v}_r \cdot {}^R\hat{\mathbf{p}}_{\mathbf{vo}})\frac{{}^R\hat{\mathbf{p}}_{\mathbf{vo}}}{\|{}^R\hat{\mathbf{p}}_{\mathbf{vo}}\|_2}, \end{aligned} \tag{11}$$

with

$${}^R\hat{\mathbf{p}}_{\mathbf{vo}} = ({}^R\mathbf{p}_{vo_i} - {}^R\mathbf{p}_{o_i}), \quad h_{vo} = \frac{\|{}^R\hat{\mathbf{p}}_{\mathbf{vo}}\|_2}{\delta_{dist}}, \quad \rho = \frac{(1-\xi)^2}{\xi^2},$$

$$\delta_{dist} = \omega_{dist}(\eta_{vo}\|{}^R\mathbf{v}_{ro_i}\|_2 + \eta_{imp}(\psi_{sf} + \psi_{obs_i})).$$

The position of dynamic virtual obstacle, ${}^R\mathbf{p}_{vo_i}$, is the projected point of origin on the path of obstacle in O_R. R_{vo_i} is decided according to the control variable of oval curvature ξ and the relative position and velocity of an obstacle. It takes a smaller value between the calculated radius and the distance between the robot and its corresponding center of a virtual obstacle to maintain the effect of a virtual obstacle. ξ decides the convexity of virtual obstacle outer shape. The bigger ξ generates the more convex one. If it is set to 0.5, the oval shape changes to a triangular one. As it is limited to $(0, 1)$, ω_{dist} has to be assigned to maintain this constraint. In deciding δ_{dist} and R_{vo}, extended size of obstacle S_{oe} and partial evaluations, ψ_{sf} and ψ_{obs}, are also considered. By virtue of this, the robot can avoid collisions against moving obstacles.

Velocity modification: When R_{coll} sweeps the path of a robot, it may collides with the obstacle according to its velocity. Though it may avoid collision by virtue of AUF around the obstacle, moving obstacles have to be considered in a different way from static obstacles. Except for the definite collision case, waiting until the obstacle passes by would be one good alternative to avoid collisions with surrounding moving obstacles. When a robot is out of R_{coll} as shown in Fig. 4(a), it collides with the obstacle if $\|{}^R\mathbf{v}_r\|_2 \in (v_{c_2}, v_{c_1})$, where v_{c_i}s are speed values of the robot reaching to the corresponding collision points in Fig. 4(a) on t_{c_i}s, respectively. In this case, two methods are considerable, accelerating or decelerating by $a_i = 2\left(\|{}^R\mathbf{p}_{oc_i}\|_2 - \|{}^R\mathbf{v}_r\|_2 t_{c_i}\right)/t_{c_i}^2$. If it is not possible for the robot to accelerate its speed to v_{c_2}, deceleration is automatically selected.

When it has already entered into R_{coll} as shown in Fig. 4(b), escaping speed v_{esc} along with its reaching time t_{esc} is the only option. If the speed of robot is smaller than v_{esc}, acceleration is performed as described above. If a target acceleration is beyond robot's capability, path modification using dynamic virtual obstacle concept is automatically triggered.

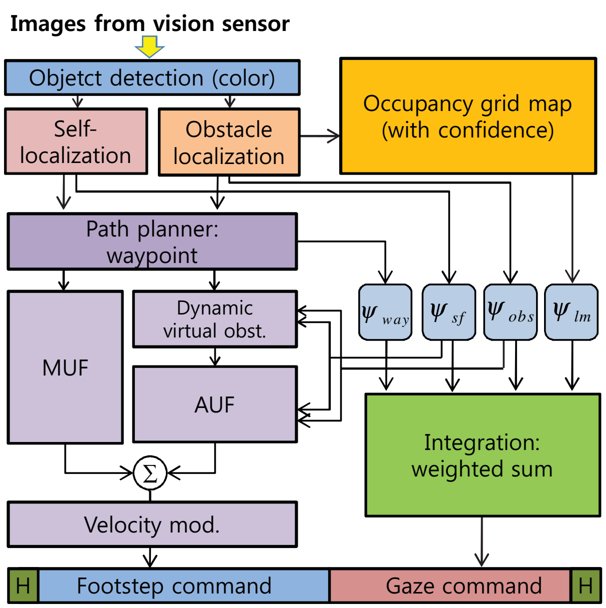


Fig. 5. Integrated architecture of navigation for HSR-VIII.

IV. Integration of The Proposed Framework

Fig. 5 shows the proposed framework. The locations and velocities of detected obstacles are obtained from color-based object detection process. Self and obstacle localizations are performed using sampling-based kalman-filter approach [17]. Using the results, four types of partial evaluations for gaze control are calculated along with the update process of occupancy grid map and path planner. In this paper, path planner simply updates sequential waypoints for the next destination. For the efficient moving obstacle avoidance process, univector field method is extended to use dynamic virtual obstacle along with velocity modification scheme, which makes robot to overtake a moving obstacle or wait until a moving obstacle passes by (see Section III). In addition, the radius of obstacles is extended to reflect those partial evaluations and their velocities. At last, this navigational information is converted to a control command for HSR-VIII. On every 5ms control period, the walking pattern generation is performed according to transferred footstep and gaze commands by Modifiable Walking Pattern Generator (MWPG) algorithm [15], [23].

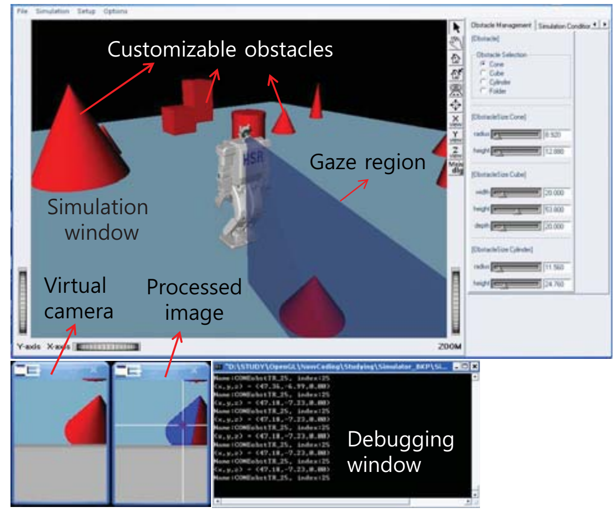


Fig. 6. HSR-VIII Simulator.

V. Simulation Environment and Results

This section explains simulation environments and results using a developed simulator, which simulates the whole navigation framework of small-sized humanoid robot, HSR-VIII [15].

A. Simulation Environments

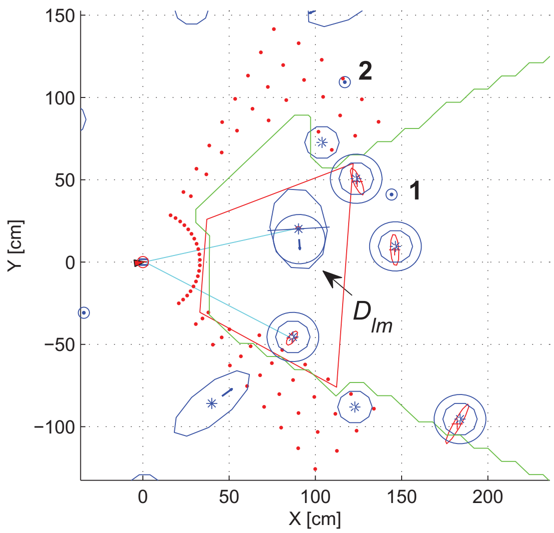
As in Fig. 6, OpenGL-based simulator was implemented using mechanical frame design files of HSR-VIII by Open-Inventor library. As described in [15], color-based vision processing was implemented using the captured images of virtual vision sensor. Moreover, the same walking pattern generation program with HSR-VIII was implemented for the simulation.

B. Partial evaluation integration

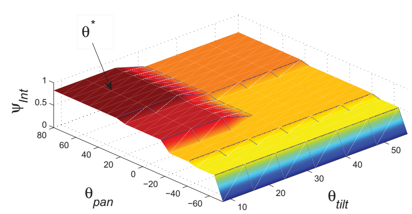
In Fig. 7(a), obstacles depicted with their extended shapes and covariance matrixes of localization error are marked as red ovals. Double circle means that the obstacle is shown on the captured screen at least once. Red dot and green line means the sample point and the confident area in the local map. Fig. 7(b) shows the result of integration ψ_{int} described in Section II-B.

C. Navigation using the proposed framework

For the verification of the proposed approach, a simulation of navigation was performed with 16 randomly generated obstacles, including four moving obstacles. Fig. 9 shows gaze fixation at nearby obstacle with maintaining the confidence map. Fig. 8 shows the snapshot of actual simulation for the case of Fig. 9(d). Fig. 9 shows the sequential results of simulations for 560.2 sec (see the accompanying movie). In Fig. 9(b), the dynamic virtual obstacle is generated along with the path of moving obstacle. Due to its dynamic trait of shape, obstacle avoidance is performed following smooth footstep trajectories as shown in Fig. 9(c). The robot successfully evaded the moving obstacle, and moved towards next via point. Using the proposed u_{muf} vector field, it approached a via point from the opposite direction of next one in order to make smooth trajectory. In moving between via points, efficient distribution of gaze was also performed.



(a)



(b)

Fig. 7. Gaze angle selection. (a) simulation graph. (b) $\mathcal{GM}$ graph and selected final gaze direction $\boldsymbol{\theta}^*$.

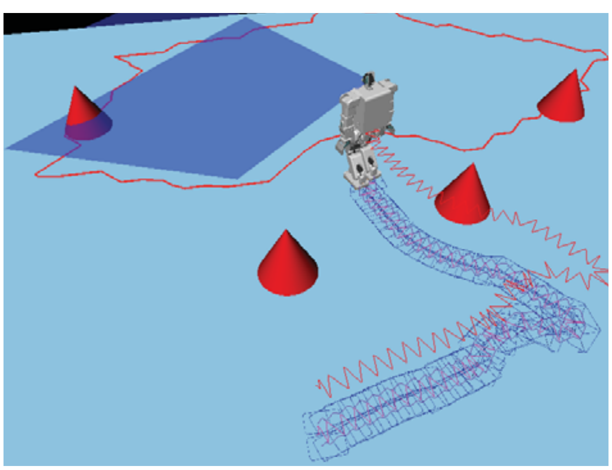


Fig. 8. Actual simulation of Fig. 9(d).

VI. Conclusions

This paper proposed the navigation framework composed of gaze control and modified univector field-based path planning scheme. In addition to path modification by using the dynamic virtual obstacle, velocity modification corresponding to the partial evaluations of gaze control was proposed. Using the proposed dynamic virtual obstacle and velocity modification scheme, humanoid robot could change its walking direction to follow a smooth and effective trajectory in simulations. Moreover, the partial evaluations of gaze control architecture was not only used on its own, but was also closely connected to the path planning scheme through the whole framework. By these, humanoid robot could cope with moving obstacles efficiently for collision avoidance in the performed simulations. However, the proposed modification of univector field scheme assumes the exact detection of obstacle velocity and distance. For this issue, uncertainty terms according to relative distance to obstacles are need to be considered.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0000831).

References

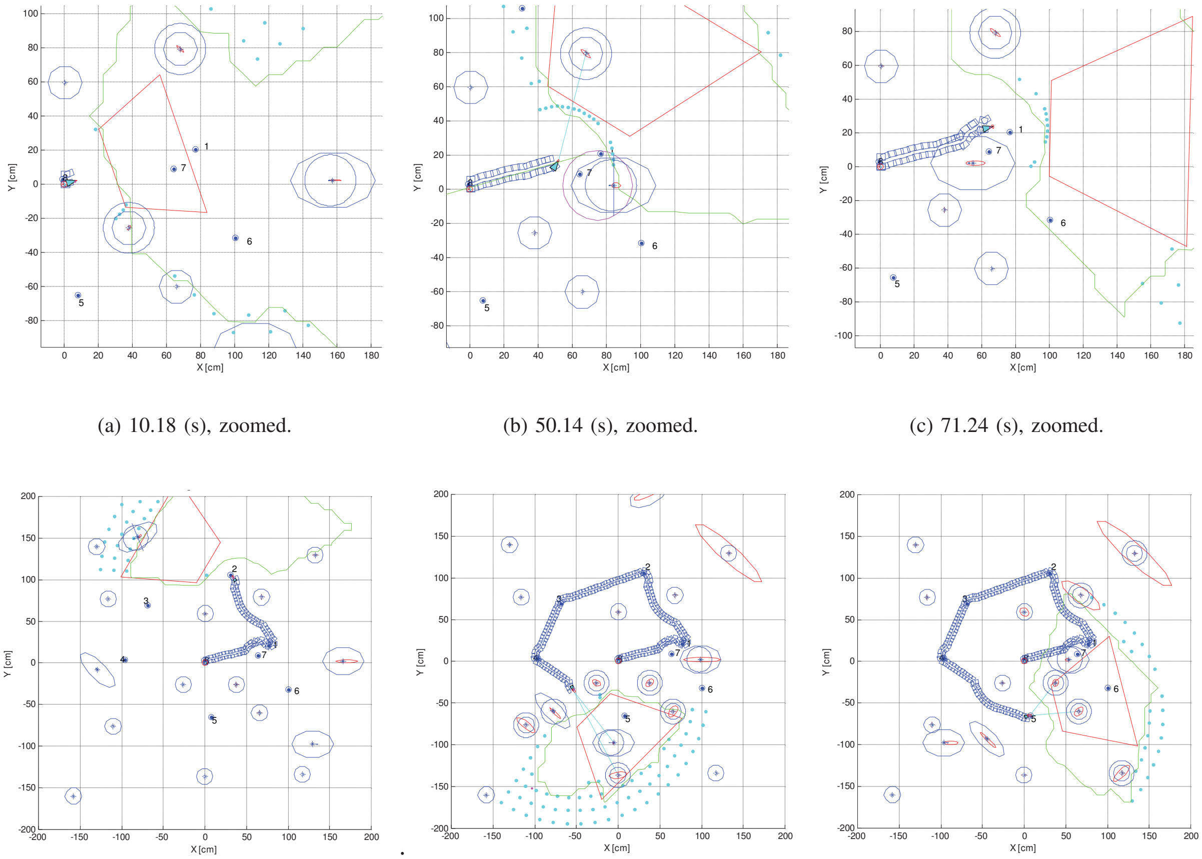
[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, “The intelligent ASIMO: system overview and integration,” in *Proc. IEEE/RSJ Int‘l Conf. Intelligent Robots and Systems*, pp. 2478–2483, 2002.

[2] K. Akachi, K. Kaneko, N. Kanehira, S. Ota, G. Miyamori, M. Hirata, S. Kajita, and F. Kanehiro, “Development of Humanoid Robot HRP-3P,” in *Proc. of IEEE-RAS Int‘l Conf. On Humanoid Robots*, Tsukuba, Japan, pp. 50–55, Dec. 2005.

[3] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, G. Cheng, “Learning CPG-based Biped Locmotion with a Policy Gradient Method:Application to a Humanoid Robot,” *Int‘l Journal of Robotics Research*, vol.27, no. 2, pp.213–228, Feb. 2008.

[4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, “A Realtime Pattern Generator for Biped Walking,” in *Proc. of IEEE Int‘l Conf. on Robotics and Automation*, Washington, DC, pp. 31–37, May 2002.

[5] J.-S. Gutmann, M. Fukuchi, M. Fujita, “3D perception and environment map generation for humanoid robot navigation,” *Int‘l Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, Oct. 2008.



(a) 10.18 (s), zoomed. (b) 50.14 (s), zoomed. (c) 71.24 (s), zoomed.

(d) 203.39 (s) (e) 356.13 (s) (f) 560.2 (s)

Fig. 9. Sequential simulation results of integrated framework.

[6] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato and M. Inaba, “Vision based behavior verification system of humanoid robot for daily environment tasks,” in *IEEE/RAS Int‘l Conf. on Humanoid Robots*, pp. 7–12, 2006.

[7] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schöder and R. Dillmann, “Toward humanoid manipulation in human-centerd environments,” *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 54–65, Jan. 2008.

[8] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, “Motion planning for humanoid robots,” in *Proc. of the Int‘l Sym. of Robotics Research*, Siena, Italy, Oct. 2003.

[9] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, “Vision-guided humanoid footstep planning for dynamic environments,” in *Proc. of the IEEE-RAS/RSJ Int‘l Conf. on Humanoid Robots (Humanoids‘05)*, pp. 13–18, Dec. 2005.

[10] E. Yoshida, I. Belousov, C. Esteves, and J. P. Laumond, “Humanoid motion planning for dynamic tasks,” in *Proc. IEEE-RAS Int‘l Conf. on Humanoid Robots*, 2005.

[11] O. Stasse, T. Foissotte and D. Larlus and A. Kheddar and K. Yokoi, “Treasure hunting for humanoid robots,” in *Workshop on Cognitive Humanoids Vision, 8th IEEE-RAS Int‘l Conf. on Humanoid Robots*, 2008.

[12] S. Ushida, K. Yoshimi, T. Okatani and K. Deguchi, “The importance of gaze control mechanism on vision-based motion control of a biped robot,” in *Proc. of the 2006 IEEE-RSJ Int. Conf. on Intelligent Robotics and Systems*, Beijing, China, Oct. 2006.

[13] J. M. Henderson, “Human gaze control during real-world scene perception,” in *TRENDS in Cognitive Science*, vol. 7, no. 11, pp. 498–504, Nov. 2003.

[14] J. F. Seara,and G. Schmidt, “Intelligent gaze control for vision-guided humanoid walking: methodological aspects,” *Robotics and Autonomous Systems*, vol. 45, pp. 231–248, 2004.

[15] J.-K. Yoo, B.-J. Lee, and J.-H. Kim, “Recent progress and development of the humanoid robot HanSaRam,” *Robotics and Autonomous Systems*, vol. 57, no. 10, pp.973-981, Oct. 2009.

[16] S. Thrun, “Learning occupany grid maps with forward sensor models,” *Journal of Autonomous Robots*, vol. 15, no. 2, pp. 111–127, Sep. 2003.

[17] T. Bailey, “Mobile robot localization and mapping in extensive outdoor environments.” Ph.D. dissertation, Univ. Sydney, Australia, 2003.

[18] Y.-J. Kim, J.-H. Kim, and D.-S. Kim, “Evolutionary programming-based univector field navigation method for fast mobile robots,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, no. 3, pp. 450–458, 2001.

[19] E. Owen, L. Montano, “Motion planning in dynamic environments using the velocity space,” in *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB (CA), pp. 997–1002, Aug. 2005.

[20] Y.-D. Hong, Y.-H. Kim, and J.-H. Kim, “Evolutionary Optimized Footstep Planning for Humanoid Robot,” in *Proc. 2009 IEEE Int. Sym. on Computational Intelligence in Robotics and Automation*, Daejeon, Korea, pp. 261–271, Dec. 2009.

[21] L. Chengqing, M. H. Ang Jr, H. Krishman, and L.-S. Yong, “Virtual obstacle concept for local-minimum-recovery in potential-field based navigation,” in *Conf. on Robotics and Automation*, San Francisco, CA, pp. 983–988, April 2000.

[22] D.-H. Kim and J.-H. Kim, “A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer,” *Robotics and Autonomous Systems*, vol. 42, no. 1, pp. 17–30, Jan. 2003.

[23] B.-J. Lee, D. Stonier, Y.-D. Kim, J.-K. Yoo, and J.-H. Kim, “Modifiable walking pattern of a humanoid robot by using allowable ZMP variation,” *IEEE Trans. Robotics*, vol. 24, no. 4, pp. 917–925, 2008.