2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

A Multi-Robot System for Unconfined Video-Conferencing

Nikhil Karnad† Volkan Isler†

Abstract— **Telepresence or tele-immersion technologies allow people to attend a shared meeting without being physically present in the same location. Commercial telepresence solutions available in the market today have significant drawbacks - they are very expensive, and confine people to the area covered by stationary cameras. In this paper, we present a mobile tele-immersion platform that addresses these issues by using robots with embedded cameras. In our system, the users can move around freely because robots autonomously adjust their locations. We provide a geometric definition of what it means to get a good view of the user, and present control algorithms to maintain a good view. The algorithms are validated both in simulation and in real experiments.**

I. INTRODUCTION

Telepresence systems which allow people to save time and money by enabling them to participate in meetings and conferences from remote locations are becoming common. For example, commercial telepresence systems are offered by Cisco, HP, Polycom, Avaya, LifeSize, Tandberg, Teliris and others [1]. These systems, though useful, have the following disadvantages. A typical solution from Cisco costs about USD 300,000 (Figure 1). Telepresence systems in the industry have their origins in academia. Marvin Minsky is credited with the conceptualization of telepresence [2]. Sabri and Prasada [3] discuss early video-conferencing systems from an implementation perspective. Studies on modern multi-party telepresence systems emerged in the mid-1990s from research in computer graphics and virtual reality. Early systems include CAVE and ImmersaDesk [4] from the University of Illinois at Chicago, and Virtualized Reality from Carnegie Mellon University [5]. Other notable telepresence systems include Virtual Auditorium [6], Virtue [7], Digital Amphitheater [8], and Coliseum [9]. The National Tele-immersion Initiative (NTII) emerged as a collaboration between the University of Pennsylvania, University of North Carolina and Brown University [10–12]. The quality of 3D reconstruction of the environment and people has also been studied in this context [13]. More recently, the TEEVE multi-stream 3D tele-immersive system was developed by the University of California at Berkeley together with the University of Illinois at Urbana-Champaign. These researchers studied techniques for image processing and 3D reconstruction, networking (end-to-end adaptations, transmission issues), and streaming data dissemination protocols in a series of papers [14–16]. All of these systems make significant

†The authors are with the Department of Computer Science and Engineering at the University of Minnesota, 200 Union St SE, Minneapolis, Minnesota 55455, USA. {karnad, isler}@cs.umn.edu

This work was supported by the grants NSF CCF-0916209 and NSF IIS-0917676.

Fig. 1. The Cisco TelePresence 3200 System. Courtesy of Cisco Systems, Inc. Unauthorized use not permitted.

(a) The iRobot Create with an Asus EEE PC (b) The WooWee Rovio

Fig. 2. Two off-the-shelf robots that are suitable for indoor use.

contributions to the field of telepresence. However, all of them use cameras that are set-up in predetermined rigs, with little or no discussion on user mobility. Recently, industrial research labs have started working on single robot platforms for teleimmersion. Examples include Nokia's Jeppe [17] and HeadThere's Giraffe [18].

Recent advances in robotics have enabled affordable machines with sensors, actuators and network communication capabilities, such as the iRobot® Create® and the WooWee™ Rovio™ (Figure 2), to permeate the domestic consumer market.

In this paper, we present a mobile video-conferencing platform using commercial off-the-shelf robots. Such a system has the advantage of freeing the end-user from the aforementioned limitations of existing systems. Applications of our system are not limited to home users and large corporations. Educational institutions can use our system to provide opportunities for distance education and international collaboration. When used as a data collection tool, such a system can help the healthcare industry. For instance, health hazards can be prevented by monitoring the deterioration of ambulatory patterns in patients with dementia [19–21].

II. System Overview

Our mobile video-conferencing system consists of multiple iRobot Create robots (differential drive), each carrying an Asus EEE PC connected over a serial interface. These inexpensive netbooks are powerful enough to run a fully-flavored Linux operating system. They also have a built-in embedded 1.3 megapixel camera. The laptops control the robots using the iRobot Open Interface (OI) specification. Communication between the robots and a central workstation uses an ad-hoc wireless network.

A key component of our system is the user's mobility model. In robotics literature, researchers model the human as an adversary and plan worst case strategies that maintain visibility of the user at all times [22, 23]. However, these methods do not directly apply to our case because they are too conservative. In general, under the adversarial model, it is usually not possible to maintain a *good frontal view* of the user. For instance, people can make sudden turns away from cameras. Therefore, we introduce a restricted mobility model, and present algorithms to maximize the time during which the system gets a good view of the user. A formal specification is presented in Section III. Section IV outlines the optimal robot trajectories resulting from our formulation.

Any video-conferencing system has to bring together various components including perception, usability studies, audio and video compression, network protocols and bandwidth. In this paper, we focus on the motion planning aspect of the system. To simplify the sensing component, we use color-based markers (Section VII), and leave vision aspects to future work. Our main contribution is the development of motion planning algorithms that enable video-conferencing systems to be mobile.

III. Problem formulation

We start by defining the coordinate frames and our state representation. Throughout this paper, we refer to the subject of the video-conference as the human, or the user. Define a mobile telepresense system with $n+1$ entities in an environment with no obstacles: the set of n mobile robots $S = \{s_1, s_2, \ldots, s_n\}$ and a human user H.

State space

We eliminate the need for localization with respect to a world frame by defining a relative state space in which the user is always at the origin (0,0). We assume that each robot has an onboard camera, with an optical frame exactly coincident with the body frame. Using on-board cameras, any robot $s_i \in S$ estimates its own distance r_i from the user and its bearing θ_i relative to the user's *direction* of motion (see Figure 3). For a discussion on estimation of these parameters and the assumptions of our vision system, see Section VII.

Motion model

Each robot is a differential drive with two controls [24]: translational speed u_w, and rotational speed u_ψ. Let α be the angle between the optical axis each robot w.r.t. the line joining the robot and the man. When $\alpha = 0$, the robot's optical axis is directed toward the user. The straight-line motion of the user is modeled by a forward velocity $u_1 = 1$. Let u_w^{MAX} be any robot's maximum speed (relative to the user's maximum speed). We require that the robots be at least as fast as the user, i.e. $u_w^{MAX} \geq 1$. The following equations govern the state transition for a single robot with variables r, θ, α, u_w, and u_ψ.

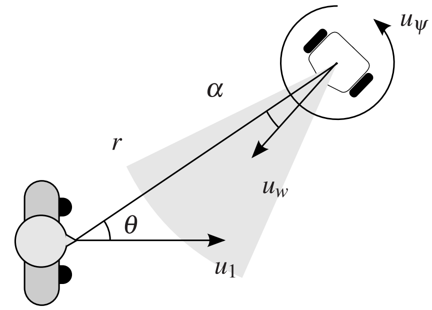

Fig. 3. The state of a robot in polar coordinates, relative to the human. Control inputs are also shown.

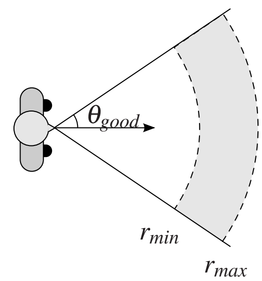

Fig. 4. The desirable set $\mathscr{D}$ is shown (shaded) in the (r, θ) plane.

$$\dot{r}(t) = -u_1 \cos\theta(t) - R u_w(t) \cos\alpha(t) \tag{1}$$

$$\dot{\theta}(t) = \frac{u_1 \sin\theta(t) - R u_w \sin\alpha(t)}{r(t)} \tag{2}$$

$$\dot{\alpha}(t) = \frac{R}{L} u_\psi(t) + \dot{\theta}(t) \tag{3}$$

In (1), (2) and (3), R is the wheel radius of the robot and L is the length of its axle. Since the robots are identical, these constants are the same for all of them.

Objective

The main objective of our robotic video-conferencing system is to maintain a good frontal view of the user for as long a duration as possible. Let $\theta_{good} > 0$ be a constant angle measured with respect to the normal vector to the human torso facing front (Figure 4). Intuitively, if $|\theta_i| > \theta_{good}$, robot s_i is too far out to the side of the user for it to have a good frontal view. We have a similar constraint for the direction the robot faces, i.e. for α_i. We further want the complete height of the human to be within the viewing frustum of the robot's camera and his image in the camera to not be too small. These constraints can be expressed in terms of constants r_{min} and r_{max}, which define an annulus around the user on the ground plane.

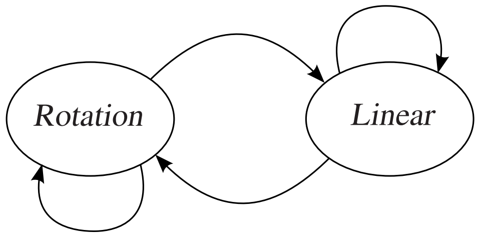

Fig. 5. State machine for the user's motion pattern.

Definition. We define the good view region, henceforth the *desirable set* $\mathscr{D} : [0,\infty[\to \mathbb{R} \times [-\pi,\pi]^2$ at any time $t \geq 0$ as follows.

$$\mathscr{D}(t) = \{(r(t), \theta(t), \alpha(t)) \text{ s.t. } r_{min} \leq r(t) \leq r_{max}; \text{ and } |\theta(t)| \leq \theta_{good}; \text{ and } |\alpha(t)| \leq \theta_{good}\}$$

Property 1. *For any given time t, there exists a robot* $s_i \in S$ *such that its state* $q_i(t)$ *is desirable, i.e.* $q_i(t) = (r_i(t), \theta_i(t), \alpha_i(t)) \in \mathscr{D}(t)$.

The desired region is always centered at the user and aligned with the normal to his torso. Note that it may not be physically possible for the mobile robots to satisfy Property 1 at all times. For instance, when a lecturer turns to face a whiteboard, there might be boundaries covering the desirable region. Therefore, a reasonable objective for our system is to *maximize* the time during which Property 1 is satisfied.

Problem Statement. Given initial configurations for the robots in S relative to the user, find trajectories for each of them such that the time during which Property 1 is maximized.

IV. Human motion models

In our system, the user wants to broadcast his view using the mobile robots as a service. Therefore, we do not model the user as a strategic player who tries to minimize our objective. We observed that when explaining a topic, or giving someone a tour of our research lab, subjects do not rapidly simultaneously change their position and orientation. We model the user's trajectory as a sequence of the following two motion patterns. Modeling other complex motions, such as those in a dance, is part of future work.

1) **Rotation**
This pattern models the case when the user makes minor adjustments to his position, but his orientation can change rapidly.
2) **Linear**
This pattern models the user when he moves from one point to another along a straight line in the direction that his torso faces. For instance, from one part of a large room to another.

The motion model for the user can be thought of as a state machine with one state for each motion pattern (Figure 5). Transitions occur when the user either crosses a threshold distance in the rotation state, or, when he comes to a stop at the end of a linear motion pattern.

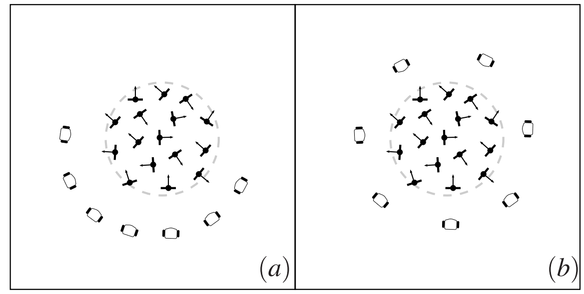

Fig. 6. Angular separation of the robots for the ROTATION motion pattern of a single user: placement (b) covers more instantaneous changes in user orientation than placement (a).

V. Motion Planning Algorithms

In this section, we derive control strategies for our mobile robots using the assumptions made by the human motion model.

A. Motion pattern - Rotation

We say that the user is rotating if (s)he does not deviate from a fixed location by a predefined threshold distance, while his torso can assume an arbitrary orientation. This distance is illustrated in Figure 6 using a dashed circle. When the user makes minor adjustments to his position and orientation, we would like the robots to remain still. If instead, we allow the robots to move around with the user, the view of each robot can change too rapidly for it to be perceived as comfortable.

Since the robots maintain a circular formation for this user motion pattern, we are looking for an optimal placement of robots around the user. Since we ignore the user's small displacements, we now focus on his orientation. We assume that any orientation in $[-\pi, \pi]$ is *equally likely*, which means that the robots need to cover every possible orientation without bias.

The optimal placement for the robots is to spread themselves around the annulus with equal angular spacing. This can be proved by contradiction. Any optimal strategy that does not assign one robot per $2\theta_{good}$ angle will, by pigeonhole principle, assign more than one robot to a sector, thus leaving at least one sector empty. By re-assigning the extra robot to that sector, we end up with a strategy that accounts for more orientations than the optimal strategy - a contradiction.

Remark. Within each sector that subtends an angle of $2\theta_{good}$ at the user, a robot can be placed anywhere such that the user lies within its field-of-view. We chose the bisecting angle for ease of implementation and visualization.

Remark. If we do not have a sufficient number of robots, i.e. when $n < \frac{2\pi}{2\theta_{good}}$, then the strategy that assigns not more than one robot to any $\frac{2\pi}{2\theta_{good}}$ different sectors is optimal, since no other strategy can cover a larger set of user orientations.

Therefore, for the ROTATION motion pattern, the robots maintain a circular formation around the user, with an equal angular separation between them w.r.t. the user.

B. Motion pattern - Linear

The user is in the LINEAR state when he translates beyond the distance threshold of the ROTATION state by moving along a straight line. When this transition occurs, the desirable region $\mathscr{D}$ stops rotating and instead, executes a translational motion parallel to the user's velocity vector (assumed to be the same as the normal to her/his torso).

When the user moves from one point to another in this state, we would like the robots to move with the user to his new location while maximizing the length of time during which Property 1 is satisfied. However, non-holonomic constraints prevent the differential drive robots from executing translational motion in arbitrary directions. Our goal is to find control inputs $u_w(t)$ and $u_\psi(t)$ for each robot $s \in S$, such that the length of time during which Property 1 holds is maximized.

We define two different robot strategies for the linear state, depending on whether the robot initially lies in $\mathscr{D}$ or not. Assume we have a sufficient number of robots in our system. We are guaranteed to have a robot $s_{best} \in \mathscr{D}$ at the time when the user starts his linear motion. In case of ties, e.g. if there are surplus robots, we pick the robot that has the least value of $|\theta|$. We call this the best robot. The goal of the best robot is to obtain the best view possible, i.e. reach the $\theta = 0$ ray as the user moves.

Now that the best robot is assigned the sector bisected by the $\theta = 0$ ray, the role of the remaining $n-1$ robots is to maintain the circular formation around the user by allocating one robot for each of the remaining sectors. This can be implemented by using an ordered sequence of predefined robot IDs, and a fixed direction around the sectors, say counterclockwise.

We have defined the boundary conditions for each of the robots. In the following section, we present optimal motion plans.

C. Breadth-first state-space search

Once the user starts his linear motion pattern, the robots have to start moving as soon as possible to avoid losing view. Therefore, we define a system parameter τ, which is a finite time horizon before which we would like the robots to be at their respective goal regions. Once τ units of time expire, the robots continue with the process of estimating the user's motion pattern and accordingly either maintain a circular formation, or, execute new finite-horizon trajectories.

For a given time interval of length τ, we build a reachability graph [24] $G(V,E)$ for the best robot s_{best}. In this graph, nodes q are discrete robot states and edges correspond to trajectories resulting from discretized controls (move back, move back and turn left, move back and turn right). For a fixed time discretization Δt, we generate $\tau/\Delta t$ levels of G, with a total of $|V| = 3^{\tau/\Delta t}$ nodes. We use heuristics to reduce the number of nodes in this graph, but in general, it is still exponential w.r.t. the time discretization. Non-exponential methods such as those in [25], are part of ongoing work.

An example of a heuristic we use is the relative angle θ. It can be pruned when it exceeds $\pi/2$, because the robot is too far out to the side of the user to return to the best view in time. Note that we also only explore backward robot motion for s_{best} because it faces opposite to the user's velocity vector.

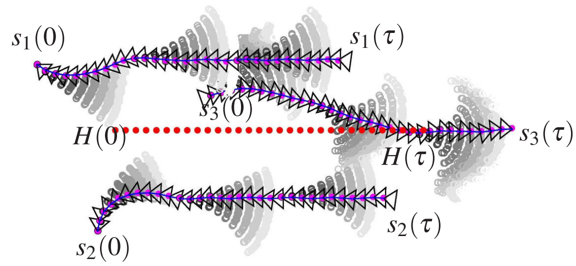

Fig. 7. This figure is best viewed in color. Time increases from left to right. The user moves along the positive X-axis (H, or solid red). The robots (s, or solid magenta) explore the state space (hollow gray circles). The robots have a restricted field-of-view (triangles).

Definition. The value (cost) of a state $q \in V$ of the reachability graph as the length of time up to that state, during which Property 1 was satisfied:

$$\begin{aligned} C(q(t) \equiv \langle r(t), \theta(t), \alpha(t) \rangle) = C(parent(q(t))) \\ + isDesirable(q(t)) * \Delta t \\ - K|\theta(t) - \theta_{sector}| \end{aligned} \tag{4}$$

There are three terms in (4), the first of which counts time from the beginning of the interval. The second term adds on Δt conditionally and the third one drives the value of θ to θ_{sector}. The constant K converts angular difference to time units, for instance by dividing by the maximum angular speed of the robot. Define $isDesirable(q(t)) = 1$ if $q(t) \in \mathscr{D}(t)$ and 0 otherwise. $\theta_{sector} = 0$ for the best robot and we distribute the remaining θ_{sector} values (one for each robot) at equally-spaced angles around the circle.

The best trajectory is the one that terminates at a leaf node with the maximum value. The complete trajectory is found by backtracking.

Remark. The function $isDesirable(q)$ can be thought of as an integral cost that adds up over time. However, the discontinuities at the geometric boundaries of the region $\mathscr{D}$ make it difficult to apply traditional optimal control methods, because the partial derivatives of $isDesirable(q)$ w.r.t. the state q do not exist everywhere.

VI. Simulation Results

Figure 7 shows the solution we obtained on a sample instance with $n = 3$ robots. The optimal robot trajectories are shown as the user moves along a straight line. The figure uses a color gradient for the reachability graph nodes, in which the colors are darker early in the simulation (near $t = 0$) and get progressively lighter as time progresses. To manage the explosion of the number of states, we use depth-limited state exploration. The robots start at relative angles 20°, 120° and $-100°$, and end at relative angles 0.75°, 140.11° and $-120.09°$. The good view angles were defined as $\theta_{good} = 40°$. For the robot that starts at 20° (s_{best}), all locations on its trajectory have a desirable view. The other two robots do

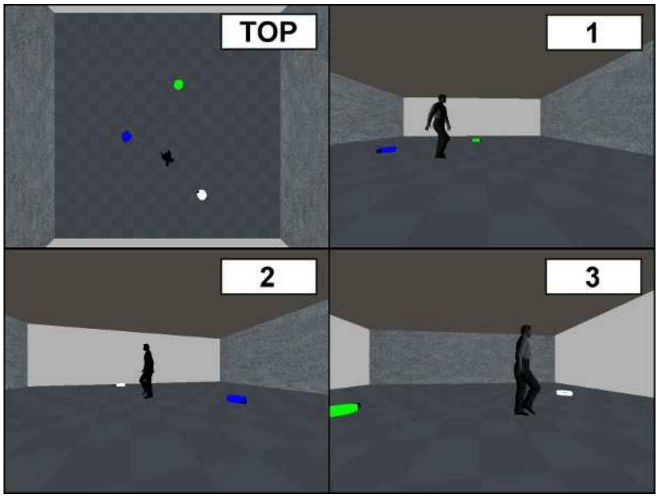

Fig. 8. A simulator (C++/OpenGL/lib3ds) to render camera views as seen by individual robots 1, 2 and 3, and a top view.

not have a good view throughout this scenario. From this instance, we observe that the best robot initially saturates its turning angle and gradually straightens as it reaches the $\theta = 0^\circ$ ray. We are currently trying to use this approach to possibly arrive at a geometric description of the curves the robot should follow, akin to the concept of T-curves by Bhattacharya et al. In their paper [26], motion trajectories were derived for limited field-of-view robots that navigate while maintaining visibility of a stationary landmark.

In order to view the user from each robot's perspective, we developed a new simulator. The application was written using simple data structures in C++ with OpenGL/GLUT for the visualization and user interface. The projections inherent to OpenGL allowed us to render the same scene from different camera viewpoints as the robots navigated in the environment. The human model used in the simulator is a 3D Studio Mesh from Artist-3D [27]. Figure 8 shows four different views, with a top view provided for visualization.

VII. Vision-based State Estimation

Our system uses vision as the primary sensing modality to extract the motion pattern of the human, as well as the state of the robots. Initially, we collected recorded human motion sequences, and extracted human motion parameters using background subtraction. This data was primarily used to validate the motion model.

In the real system, since the robots are moving, background subtraction is difficult. To simplify the complexity of the vision component, we placed a colored marker on the human (see Figure 9). We use the Continuously Adaptive Mean SHIFT (CAMSHIFT) algorithm [28, 29] to compute marker parameters. This algorithm uses the color histogram of a predefined marker as the model density distribution, and a back-projected image as the probability distribution of the model. We chose this algorithm because it works robustly even when the camera is in motion. We used the CAMSHIFT implementation provided in the OpenCV

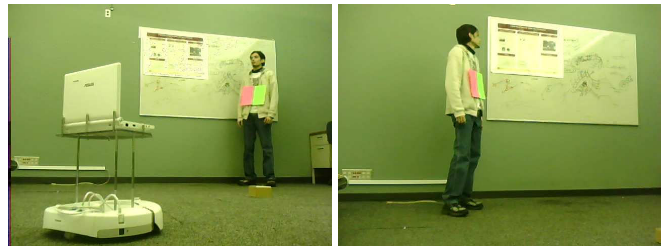

Fig. 9. Proof-of-concept experiment showing the best robot (left) and a view from that robot at a different time (right).

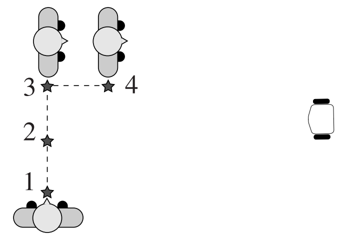

Fig. 10. Example user trajectory for testing the vision-based state estimation from a stationary robot: Linear from 1 to 2, Rotation back and forth at 2, Linear from 2 to 3, Rotation by 90° at 3, Linear from 3 to 4.

library. Once we track the human motion in each frame, we obtain the dimensions of the image of the marker along with the centroid of its bounding box. Using the intrinsic camera parameters and the known height of the marker, we obtain the distance of the person from the camera. The angle of the human motion with respect to the camera's optical axis is also determined from the centroid position of the bounding boxes of consecutive frames. Once these parameters are determined, we estimate the state of the human as follows. The first centroid point of the human is initialized as the reference point. If the centroid in successive frames moves by move than a threshold λ_1 with respect to the reference point, we classify the user to be in the Linear state. If not, the user remains in the Rotation state. If for λ_2 frames, the centroid does not move with respect to the reference, we set the state back to Rotation and the reference point is updated. Figure 11 shows the output of the state estimator.

VIII. Conclusion

Existing tele-conferencing systems are costly, and they confine the user to a small, pre-determined area. In this paper, we introduced a robotic tele-conferencing system which employs inexpensive mobile robots to overcome these limitations. We presented a realistic user mobility model, as well as robot motion strategies to maximize the amount of time the user is covered from a good view. Our preliminary system works for large rooms and open areas, but ongoing work aims to address cluttered environments where robots have to plan for collisions and occlusions.

The primary goal of our future work is to build a complete system which achieves tele-presence anywhere, anytime. This requires addressing a number of challenging issues including markerless state estimation, video transmission and human-factors. An interesting future direction is the explo-

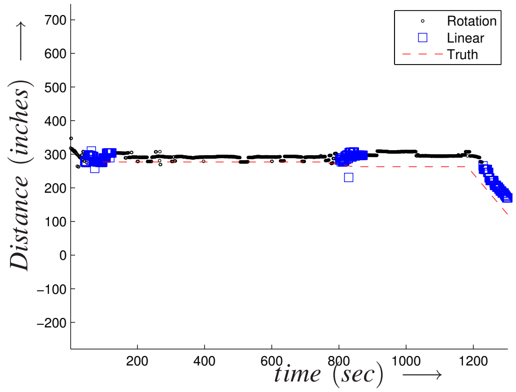

Fig. 11. Real-time vision-based state estimation on the human trajectory from Figure 10.

ration of different objective functions, such as maximizing the accuracy of 3D reconstruction.

Acknowledgments

Thanks are due to Kandan Ramakrishnan for the marker-based state estimation module. The authors would like to thank the members of the Robotic Sensor Networks lab at the University of Minnesota for their help and support.

References

[1] Lucy Hipperson and David Molony. Telepresence: beyond Cisco. Technical Report, Ovum Ltd, February 2008. http://store.ovum.com/Product.asp?pid=38759&etr=infaus.

[2] M. Minsky. Telepresence. *Omni*, 2(9):45–52, 1980.

[3] S. Sabri and B. Prasada. Video conferencing systems. *Proceedings of the IEEE*, 73(4):671–688, 1985.

[4] J. Leigh, T.A. DeFanti, A. Johnson, M.D. Brown, and D.J. Sandin. Global tele-immersion: Better than being there. In *ICAT*, volume 97, pages 10–17, 1997.

[5] T. Kanade, PJ Narayanan, and PW Rander. Virtualized reality: Concepts and early results. In *Proceedings IEEE Workshop on Representation of Visual Scenes, 1995.(In Conjuction with ICCV'95)*, pages 69–76, 1995.

[6] Milton Chen. Design of a virtual auditorium. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 19–28, New York, NY, USA, 2001. ACM.

[7] O. Schreer, N. Brandenburg, S. Askar, and E. Trucco. A virtual 3d video-conferencing system providing semi-immersive telepresence: A real-time solution in hardware and software. In *Proc. Intl. Conf. eWork and eBusiness*, pages 184–190, 2001.

[8] Ladan Gharai, Colin Perkins, Ron Riley, and Allison Mankin. Large scale video conferencing: A digital amphitheater. In *Proc. 8th International Conference on Distributed Multimedia Systems*, 2002.

[9] H.H. Baker, D. Tanguay, I. Sobel, D. Gelb, M.E. Goss, W.B. Culbertson, and T. Malzbender. The coliseum immersive teleconferencing system. In *Proc. International Workshop on Immersive Telepresence*, volume 6, 2002.

[10] J. Lanier. Virtually there. *Scientific American*, pages 66–75, April 2001.

[11] Herman Towles, Wei chao Chen, Ruigang Yang, Sang-Uok Kum, Henry Fuchs, Nikhil Kelshikar, Jane Mulligan, Kostas Daniilidis, Loring Holden, Bob Zeleznik, Amela Sadagic, and Jaron Lanier. 3d tele-collaboration over internet2. In *In: International Workshop on Immersive Telepresence, Juan Les Pins*, 2002.

[12] David E. Ott and Ketan Mayer-Patel. Coordinated multi-streaming for 3d tele-immersion. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 596–603, New York, NY, USA, 2004. ACM.

[13] J. Mulligan, V. Isler, and K. Daniilidis. Trinocular stereo: A new algorithm and its evaluation. *International Journal of Computer Vision*, 47:51–61, 2002.

[14] Sang-Hack Jung and Ruzena Bajcsy. A framework for constructing real-time immersive environments for training physical activities. *Journal of Multimedia*, 1(7):9–17, 2006.

[15] Zhenyu Yang, Bin Yu, Klara Nahrstedt, and Ruzena Bajcsy. A multi-stream adaptation framework for bandwidth management in 3D tele-immersion. In Brian Neil Levine and Mark Claypool, editors, *NOSSDAV*, page 14. ACM, 2006.

[16] Wanmin Wu, Zhenyu Yang, Dongyun Jin, and Klara Nahrstedt. Implementing a Distributed 3D Tele-immersive System. *IEEE International Symposium on Multimedia (ISM)*, 2008.

[17] Nokia Jeppe - Domestic videoconferencing. http://research.nokia.com/research/teams/extended_home/jeppe.html, 2008. Nokia Research Labs, Helsinki, Finland.

[18] Headthere giraffe - a telepresence robot. http://www.headthere.com/technical.html.

[19] Richard A. Hussian and Debbie C. Brown. Use of Two-Dimensional Grid Patterns to Limit Hazardous Ambulation in Demented Patients. *J Gerontol*, 42(5):558–560, 1987.

[20] C.C. Lin, M.J. Chiu, C.C. Hsiao, R.G. Lee, and Y.S. Tsai. Wireless health care service system for elderly with dementia. *IEEE Transactions on Information Technology in Biomedicine*, 10(4):696–704, 2006.

[21] Chieko Greiner, Kiyoko Makimoto, Mizue Suzuki, Miyae Yamakawa, and Nobuyuki Ashida. Feasibility Study of the Integrated Circuit Tag Monitoring System for Dementia Residents in Japan. *American Journal of Alzheimer's Disease and Other Dementias*, 22(2):129–136, 2007.

[22] S.M. LaValle, C. Becker, and J. Latombe. Motion strategies for maintaining visibility of a moving target. In *In Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 1997.

[23] T. Muppirala, S. Hutchinson, and R. Murrieta-Cid. Optimal motion strategies based on critical events to maintain visibility of a moving target. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3826–3831, 2005.

[24] Steven Michael LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[25] S.M. Lavalle and P. Konkimalla. Algorithms for computing numerical optimal feedback motion strategies. *The International Journal of Robotics Research*, 20(9):729, 2001.

[26] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson. Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints. *IEEE Transactions on Robotics*, 23(1):47–59, 2007.

[27] Free 3d models by artist-3d.com. Digital art and 3d media exchange directory on the Internet. http://artist-3d.com/free_3d_models/dnm/model_disp.php?uid=567.

[28] G.R. Bradski et al. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(2):12–21, 1998.

[29] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1197–1203 vol.2, 1999.