Exploration and Mapping with Autonomous Robot Teams
Results from the MAGIC 2010 Competition

Edwin Olson Johannes Strom Rob Goeddel Ryan Morton Pradeep Ranganathan Andrew Richardson
University of Michigan
{ebolson, jhstrom, rgoeddel, rmorton, rpradeep, chardson}@umich.edu

***Abstract*— The potential impact of autonomous robotics is magnified when those robots are deployed in teams: a team of cooperating robots can greatly increase the effectiveness of a human working alone, making short work of search-and-rescue and reconnaissance tasks. To achieve this potential, however, a number of challenging problems ranging from multi-robot planning, state estimation, object detection, and human-robot interfaces must first be solved. The MAGIC 2010 competition, like the DARPA grand challenges that preceded it, presented a formidable robotics problem designed to foster fundamental advances in these difficult areas. MAGIC asked teams of robots to collaboratively explore and map a 500×500m area, detect and track benign and dangerous objects, and collaborate with human commanders while respecting their cognitive limits.**

This paper describes our winning entry in the MAGIC contest, where we fielded a team of 14 autonomous robots supervised by two human operators. While the challenges in MAGIC were diverse, we believe that cooperative multi-robot state estimation is ultimately the critical factor in building a successful system. In this paper, we describe our system and some of the technological advances that we believe were responsible for our success. We also contrast our approach to those of other teams.

a) Keywords: Multi-Agent Systems, Human-Robot Interaction, SLAM

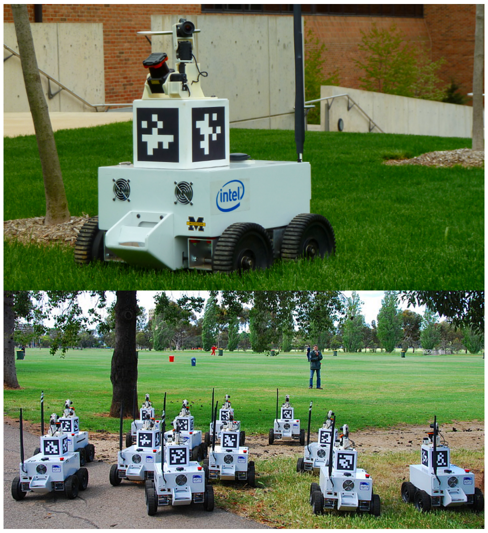

Fig. 1. Team Michigan Robots. We deployed fourteen custom-made robots that cooperatively mapped a 500×500m area. Each robot had a color camera and a laser range finder capable of producing 3D point clouds.

I. Introduction

Urban reconnaissance and search-and-rescue are ideal candidates for autonomous multi-robot teams due to their inherent parallelism and to the danger they present to humans. However, this domain presents many challenging problems which arise from working in complex, stochastic and partially observable environments. In particular, non-uniform and cluttered terrain in unknown environments presents challenges for both state-estimation and control, resulting in complicated planning and perception problems. Limited and unreliable communications further complicates coordination amongst the individual agents and with their human commanders.

To help address these difficult problems, the Multi-Autonomous Ground robot International Challenge (MAGIC) was conducted in November of 2010, where five teams comprising nearly 40 robots competed for over a million dollars in prize money. Teams were instructed to explore and map a large indoor-outdoor area while recognizing and neutralizing threats such as simulated bombs and enemy combatants. Although the contest showcased the abilities of teams to effectively coordinate autonomous agents in a challenging environment it also showed the limitations of the current state-of-the-art in state estimation and perception (e.g. map building and object recognition).

The MAGIC competition was the most recent of the robotics grand challenges, following in the tradition of the well-known competitions sponsored by the Defense Advanced Research Projects Agency (DARPA). These competitions ultimately trace back to a congressional mandate in 2001 requiring one-third of all ground combat vehicles to be unmanned by 2015. Over the course of the three DARPA challenges, teams developed technologies for fully autonomous cars, including the ability to drive in urban settings, navigating moving obstacles and obeying traffic laws [1], [2]. These contests fostered the development of new methods for planning, control, state estimation, and perhaps most importantly, robot perception and sensor fusion.

Unfortunately these advances were not mirrored in smaller robots, such as those used by soldiers searching for and neutralizing improvised explosive devices (IEDs) or for robots intended to help first responders with search and rescue missions. Instead, tele-operation (remote joystick control by a human) remains the dominant mode of interaction. These real-world systems pose challenges that were not present in the DARPA grand challenges which has held them back:

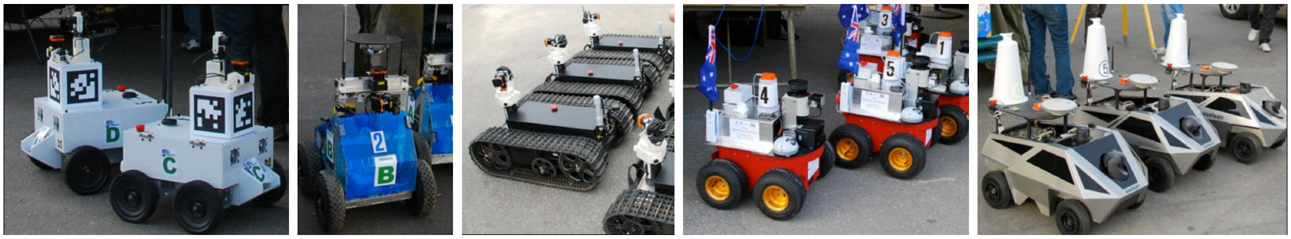

Fig. 2. Finalist robots. Each team used a unique robot platform (left-to-right, in ranked order): Team Michigan used 14 custom-built robots; University of Pennsylvania fielded 7 custom robots; RASR based its 7 robots on the Talon commercial platform; Magician adapted a commercial base for its 5 robots; Cappadocia built 6 tailored vehicles.

1) **Limited/Unreliable GPS.** GPS is often unreliable or inaccurate in dense urban environments or indoors. GPS can also be jammed or spoofed by an adversary. The winning DARPA vehicles relied extensively on GPS.
2) **Multi-robot cooperation.** Individually, robots are generally less capable than humans. Their potential arises from multi-robot deployments that explicitly coordinate.
3) **Humans-in-the-loop.** By allowing a human to interact with a robot team in real-time, the system becomes more effective and can adapt to changes in the mission objectives or priorities. This entails developing visualization methods and user interface abstractions that allow the human to understand and manipulate the state of the team.

The MAGIC contest focused on increasing the effectiveness of multi-robot systems by increasing the number of robots that a single human commander could effectively manage. This is in contrast to current robot systems, which typically have one or more operator per robot. The contest was jointly organized by the United States Army and the Australian Defence Science and Technology Organisation and required participants to deploy a team of cooperating robots to explore and map a hostile area, recognize and catalog the location of interesting objects (people, doorways, IEDs, cars, etc.), and perform simulated neutralization of IEDs using a laser pointer. Two human operators were allowed to interact with the system, but the interaction time was measured and used to calculate a penalty to the team's final score.

The contest attracted 23 teams from around the world, and through a series of competitive down-selects, was reduced to five finalists who were invited to Australia for the final competition. The venue was the Adelaide Showgrounds, a 500x500m area including a variety of indoor and outdoor spaces. Aerial imagery provided by the contest organizers constituted the only prior knowledge. While DARPA challenges provided detailed GPS waypoints describing the location and topology of the safe roads, MAGIC robots would have to figure this out on their own. Whereas other search-and-rescue robotics contests typically focus on smaller environments with significant mobility and manipulation challenges, (e.g. RoboCup Rescue league), MAGIC was conducted at a much larger scale with an increased focus on autonomous multi-robot cooperation [3].

To succeed in MAGIC, a team needed to combine robot perception, mapping, planning, and human interfaces. This paper highlights some of the key decisions and algorithmic choices which led to our team's first place finish [4]. Additionally, we will highlight how our mapping and state estimation system differed from other competitors, one of the key differences which we believe set our team apart from our competitors.

II. System Design

We begin by describing how our system worked at a high-level. Fundamentally, most teams pursued a similar strategy.

Our system was largely centralized: a ground control station collected data from individual robots, fused it to create an estimate of the current state of the system (the position of the robots, the location of important objects, etc.), then used this information to assign new tasks to the robots. Most robots focused on exploring the large competition area, a task well-suited to parallelization. However, other robots could perform additional tasks, such as the neutralization of an improvised explosive device. The discovery of such a device would cause a "neutralize" task to be assigned to a nearby robot.

The human operators were located at the ground control station and were able to view the current task assignments, a map of the operating area, and (perhaps most importantly) guide the system by vetting sensor data or overriding task assignments.

The robots received their task assignments via radio and were responsible for executing that task without additional assistance from the ground control station. For example, robots used their 3D laser range-finder to identify safe terrain and avoided obstacles on their own. They were also responsible for autonomously detecting IEDs and other objects. The information gathered by the robots (including object detection data and a map of the area immediately near the robot) was heavily compressed and transmitted back to the ground control station. (In practice, these messages were often relayed by other robots in order to overcome the limited range of our radios.)

With the newly-collected information, the ground control station updated its map, user interfaces, and computed new (and improved) tasks for each of the robots. This process continued until the mission was completed.

Such a system poses many challenges: How does the ground control station compute efficient tasks for the robots in a way that maximizes the efficiency of the team? How can a human be kept informed about the state of the system? How can the human contribute to the performance of the system? How do the robots reliably recognize safe and unsafe terrain? How do they detect dangerous objects? How can the information collected by the robots be compressed sufficiently to enable it to be transmitted over a limited and unreliable communications network? How does the ground control station combine information from the robots into a single globally-consistent view?

Recognizing that many of these tasks rely on a high-quality map of the world, our team focused on the challenge of fusing robot data into a globally-consistent view. Not only was the accuracy of this map a primary evaluation criterion in the MAGIC competition, but it was also a critical component of effective multi-agent planning and the human-robot interface. For example: it is difficult to know where to send the robots next if one does not know where they are now, or if one does not know where they have already explored.

One of the more obvious differences between our team and other teams was the accuracy of the maps that we produced. Map quality pays repeated dividends throughout our system, with corresponding improvements in human-robot interfaces, planning, etc. The variability in map-quality between different teams is a testament to the difficulty and unsolved nature of multi-robot mapping. Our team began with a state-of-the-art system, but these methods were inadequate both in terms of scaling to large numbers of robots, and in terms of dealing with the errors that inevitably occur. New methods, both automatic and human-in-the-loop, were needed in order to achieve an adequate level of performance. The following section explores a few of these methods.

III. Technical Contributions

While MAGIC posed many technical challenges, mapping and state estimation were arguably the most critical. Using the Global Positioning System (GPS) may seem like an obvious starting point. However, even under best-case conditions, GPS can not provide a navigation solution for the significant fraction of the time that robots spend indoors. Outdoors, GPS data (particularly from consumer grade equipment) is often fairly good—within a few meters, perhaps. But GPS can also be wildly inaccurate due to effects like multi-path. In a combat situation, GPS can be easily jammed or even spoofed. Consequently, despite having GPS receivers on each robot, we ultimately opted not to use GPS data, instead relying on the robots' sensors to recognize landmarks. This strategy was not universally adopted, however; most teams did use GPS to varying degrees.

A. *Overview of Mapping and State Estimation*

Conceptually, map-building can be thought of as an alignment problem: robots periodically generate *maplets* of their immediate surrounds using a laser scanner. The challenge is to determine how to arrange the maplets so that they form a large coherent map, much like the process of assembling a panoramic photo from a number of overlapping photos (see Fig. 3). Not only can we recover a map this way, but the position of each of the robots is also known, since they are at the center of their maplets.

Our team's state-estimation system was based on a standard probabilistic formulation of mapping in which the desired alignment can be computed by performing inference on a factor graph. (See [5], [**?**] for a survey of other approaches.) Our factor graph contains nodes for unknown variables (the location of each maplet) and edges connecting nodes when something is known about the relative geometric position of the two nodes. Loosely speaking, an edge encodes a geometrical relationship between two maplets, i.e., "maplet A is six meters east and rotated thirty degrees from maplet B." Of course, none of these relationships are known with certainty, so edges are annotated with a covariance matrix. It is common for a map to contain many of these edges, and for those edges to subtly disagree with one another.

More formally, let the position of all the maplets be represented by the state vector x. This vector can be quite large: it contains two translation and one rotation component for every maplet, and there can be thousands of maplets. Edges convey a conditional probability distribution $p(z_i|x)$, where z_i is a sensor measurement. This quantity is the *measurement model*: given a particular configuration of the world, it predicts the distribution of the sensor. For example, a range sensor might return the distance between two variable nodes plus some Gaussian noise whose variance can be empirically measured.

Our goal is to compute $p(x|z)$, or the posterior distribution of the maplet positions given all of the sensor observations. Using Bayes' rule, and assuming that we have no a priori knowledge of what the map should look like (i.e., $p(x)$ is uninformative), we obtain:

$$p(x|z) \propto \prod p(z_i|x) \tag{1}$$

Our goal is to find the maplet positions x that has maximum probability $p(x|z)$. Assuming that all of the edges are simple Gaussian distributions of the form $e^{(z_i-\mu)^T\Sigma^{-1}(z_i-\mu)}$, this computation becomes a non-linear least-squares problem. Specifically, we can take the logarithm of both sides, which converts the right hand side into a sum of quadratic losses. We maximize the log probability by differentiating with respect to x, which results in a first-order linear system. The key idea is that maximum likelihood inference on a Gaussian factor graph is equivalent to solving a large linear system; see [6] for a more detailed explanation. The solution to this linear system yields the position of each maplet.

Critically, the resulting linear system is extremely sparse. This is because each edge typically depends on only two maplet positions. In our system, each maplet was, generally, connected to between 2 and 5 other maplets. Sparse linear algebra methods can exploit this sparsity, greatly reducing the time needed to solve the linear system for x. Our method was based on sparse Cholesky factorization [7]: we could compute solutions for a graph with 4200 nodes and 6300

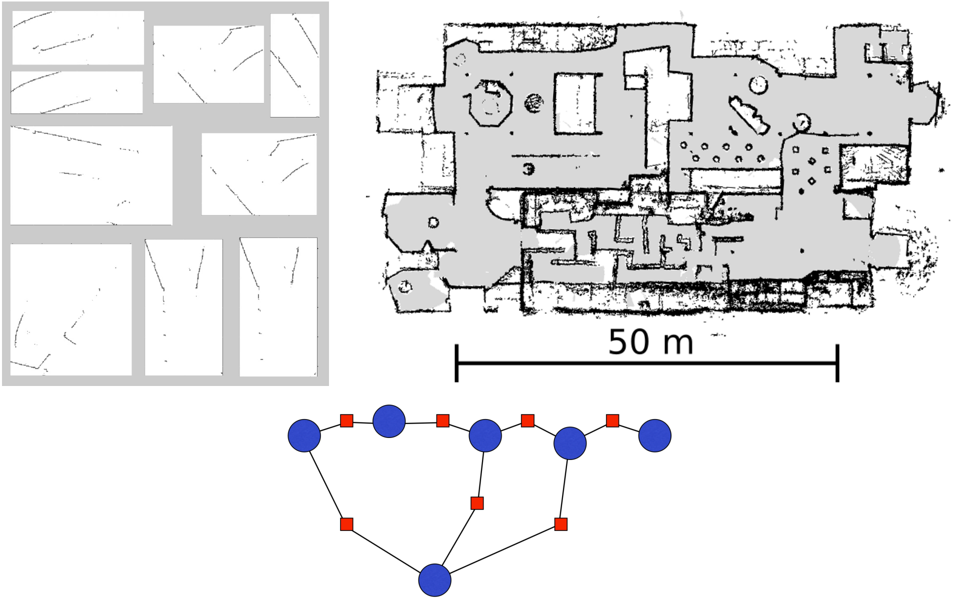

Fig. 3. Mapping Overview. Individual "maplets" (top left) are matched in a pair-wise fashion; the resulting network of constraints can be illustrated in a factor graph similar to the bottom figure, in which circles represent robot positions and squares represent probabilistic constraints. The final map (top right) is computed by reprojecting all of the sensor observations according to the maximum likelihood robot positions.

edges in about 250 ms on a standard laptop CPU. New data is always arriving, and so this level of performance allows the map to be updated several times per second.

An important advantage of using the factor graph formulation is that it is possible to retroactively edit the graph to correct errors. For example, if a sensing sub-system erroneously adds an edge to the graph (incorrectly asserting, perhaps, that two robot poses are a meter apart), we can "undo" the error by deleting the edge and computing a new maximum likelihood estimate. This sort of editing is not possible, for example, with methods based on Kalman filters. In our case, we rely on human operators to correct these relatively rare errors (see Section III-C).

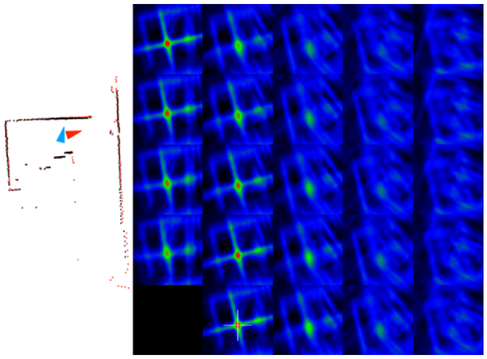

Fig. 4. Brute-force search for best maplet alignments. The search space is 3D (two translation and one rotation component) which is illustrated above as a series of 2D cross-sections. Bright areas indicate good alignments. Finding the best match quickly is critical to a large-scale mapping system. The resulting matches become edges in the factor graph.

B. Scan Matching & Loop Validation

Our mapping approach is dependent on identifying high-quality edges. In general, more edges result in a better map since the linear system becomes over-constrained, reducing the effect of noise from individual edges.

Our system used a number of different methods to generate edges including dead-reckoning (based on wheel encoder odometry and a low-cost IMU) and visual detection of other robots using their 2D "bar codes" (see Fig. 1) [8]. But by far, the most important source of edges in our system was our scan-matching system. This approach directly attempts to align two maplets by correlating them against each other, looking for the translation and rotation that maximize their overlap. One such matching operation is illustrated in Fig. 4: the probability associated with each translation and rotation is computed in a brute-force fashion.

This alignment process is computationally expensive, and in the worst-case, every maplet must be matched with every other maplet. In practice, our dead-reckoning data can help rule out many matches. But with fourteen robots operating simultaneously and each one producing a new maplet every 1.4 seconds, hundreds or thousands of alignment attempts per second are needed.

Our approach to mapping was based on an accelerated version of a brute-force scan matching system [9]. The key idea is to use a multi-resolution matching system: we generate low-resolution versions of the maplets and first attempt to align these. Because they are smaller, the alignment is much faster. Good candidate alignments are then attempted at higher resolution.

While simple in concept, a major challenge is ensuring that the low-resolution alignments do not under-estimate the quality of an alignment that could occur using higher-resolution maplets. Our solution relied on constructing the low-resolution maplets in a special way. Rather than applying a typical low-pass-filter/decimate process (which would tend to obliterate structural details), we used a max-decimate kernel. This ensures that low-resolutions of the maplets are conservative: when aligning low-resolution maplets, we never under-estimate the overlap that could result from aligning the full-resolution maplets.

While our previous two-level scan matcher was fast (it could perform around 50 matches per second), a faster version of the algorithm was needed for MAGIC. We used a full image pyramid of maplet resolutions; when alignments at low-resolution still result in small amounts of overlap, it eliminates large portions of the search space. Our improved multi-resolution method achieved 500 match attempts per second, a rate that was pivotal in being able to keep up with the datarate of our robots.

Other teams used similar maplet matching strategies, though they were not as fast. The Australian team "Magician", for example, reports that their GPU-accelerated system was capable of 7-10 matches per second.

This improvement in our matching speed allowed us to consider a large number of possible matches in real-time to support our global map. However, our state-of-the-art method has a non-zero false positive rate. In short, it will align maplets based on similar-looking structures, even if those maplets are not actually near each other.

There is a fundamental trade-off between the number of true positive and an increased risk of false positives. Increasing the threshold for what constitutes a "good enough" match also increases the likelihood that similar looking, but physically distinct locations will be incorrectly matched. These types of false-positive matches can cause the inference method to distort the map in order to explain the error.

To reduce the false positive rate to a usable level, we performed a loop-validation step on candidate matches before they were added to the factor graph. The basic idea of loop validation is to require that multiple matches "agree" with each other [10], [11], [12]. Specifically, consider a topological "loop" of matches: a match between node A and B, another match between B and C, and a third match between C and A. If the matches are correct, then the composition of their rigid-body transformations should approximately be the identity matrix and can be added to the graph. Of course, it is possible that two matches might have errors that "cancel", but this seldom occurs.

C. Human Robot Interfaces

In simple environments, such as an indoor warehouse, the combination of loop-validation and automatic scan-matching we have presented are sufficient to support completely autonomous operation of our robot team (see Figure 5). However, in less structured environments (like many of the outdoor portions of the MAGIC 2010 competition), mapping errors still occur. For example, the MAGIC venue contained numerous cable conduits which caused robots to unknowingly get stuck, causing severe dead reckoning estimation error. Our system was not able to handle these types of problems autonomously.

However, these types of problems are relatively obvious to a human operator. We developed a user interface that allowed a human operator to look for errors and intervene when necessary. With new (validated) loop closures being added to the graph at a rate of 2-3 per second, it would be easy to overwhelm the human operator by asking for explicit verification of each match.

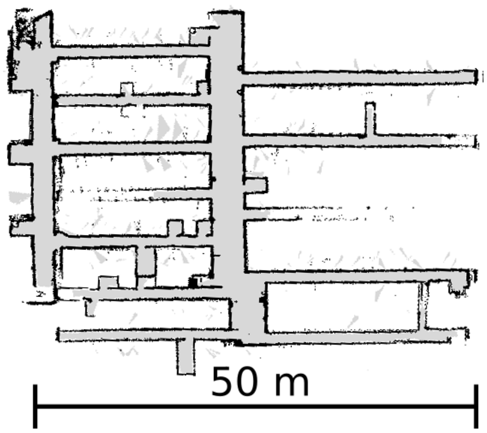

Fig. 5. Indoor storage warehouse map. In uncluttered environments posing few mobility challenges, our system can explore and map with very little human intervention.

Instead, the human operators would monitor the entire map. When an error occurred (typically visible as a distortion in the map), the operator could "roll-back" automatically added matches until the problem was no longer present. The operator could then ask the mapping system to perform an alignment between two selected maplets near where the problem was detected. This human-assisted match served as a prior for future autonomous match operations, and so the autonomous mapping system would be much less likely to make the same mistake again.

We found that this approach, which required only a few limited interactions to remove false positives, was a highly effective use of humans to support the continued autonomy of our planning system. We were the only team to build a user interface that allowed direct supervision of the real-time state estimate; other teams were required to handle failures in automatic state estimation by requiring humans to track the global state manually and then intervening at the task allocation level. Early versions of our system lacked the global mapping system— the human operators were instead provided with separate map displays for each robot. Our experience with this approach indicated that operators could not effectively handle more than 5 or 6 robots in such a fashion. Maintaining a global map is critical to scaling to larger robot teams, and our user interface was a key part of maintaining the consistency of that map.

IV. Evaluation

The main evaluation metrics for an autonomous reconnaissance system are the quality of the final map produced and the amount of human assistance required to produce it. These were also the primary metrics the MAGIC organizers used to determine the winner and subsequent ranking of the finalists (see Figure 2). While the specific performance data used

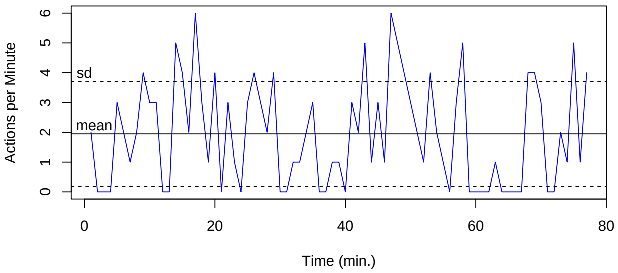

Fig. 7. Map Interaction Experiment. Our mapping operator re-enacted supporting role for phase 2 dataset to measure the frequency of interaction required to maintain a near-perfect state estimate. See Figure 6 for resulting map. Overall, the human work-load was quite modest, averaging two interactions per minute.

during the contest were not made public, we will present selected results we obtained by processing our logs from the contest. Additionally, we will compare with other teams' published results where possible.

Lacking detailed ground truth for the MAGIC venue, the best evaluation of map quality is necessarily subjective. Figure 6 shows post-processed maps for our team in comparison to the mapping software of Magician (4th place) applied to the data collected by UPenn's team (2nd place). Additionally, the actual map produced by our system during the competition is shown inset. These results show that high quality maps can be produced in this domain – our competition-day results show that our state estimation was sufficiently good to be used for support of online planning. This system allowed us to completely explore the first two phases of the magic competition, while simultaneously performing mission objectives relating to dynamic and static dangers such as IEDs and simulated mobile enemy combatants.

Ideally, we would also like to measure the frequency of human interaction required to support our state estimation system during the MAGIC contest. However, the data necessary to evaluate this metric was not collected during our competition run, thus we replicated the run by playing back the raw data from the competition log and having our operator re-enact his performance during the competition. These conditions are obviously less stressful than competition, but are still representative of human performance. The result, shown in Figure 7 was an addition of 175 loop closures, on average two interactions per minute, which generally occurred in bursts. However, at one point, the operator did not interact with the system for 5.17 minutes.

Our evaluation shows that we were able to support cooperative global state estimation for a team of autonomously-navigating robots using a single part-time operator. Yet there remain significant open problems including reducing human assistance to even lower levels by improving the ability of the system to autonomously handle errors. Additional evaluation of our system, and technical descriptions of the other finalists can be found in separate publications [4], [14], [15], [16], [17].

V. Discussion

The MAGIC competition's focus was on increasing the robot-to-human ratio and on efficiently coordinating the actions of multiple robots. Key to reducing the cognitive load on the operators is increasing the autonomy of the robots; for a given amount of cognitive loading, more robots can be handled if they simply require less interaction. We identified global state estimation as a key technology to enable autonomy, and we believe that the mapping system we deployed for MAGIC outperforms the systems of our competitors. While this was one of the key factors differentiating us from other finalists, it was not the only important point of comparison. In fact, many of the other choices we made while developing our system also had an important impact on our performance.

In particular, we made a strategic choice early during our development that our team would emphasize the use of a large team of robots. This is reflected in the fact that we brought twice as many robots to the competition as the next largest team. This strategy ultimately affected the design of all our core systems, including mapping, object identification, and communication. Given that we had a finite budget, it also forced us to deploy economical robot platforms which had only the bare necessities in sensing to complete the challenge. The result was that our robots were also the cheapest of any of the finalists (by a significant margin), costing only $11,500 USD each.

One approach to detecting dangerous objects, for example, is to transmit video feeds back to the human operators and rely on them to recognize the hazard. Given a design goal of maximizing the number of robots, such a strategy is unworkable: there is neither the bandwidth to transmit that many images, nor could the humans be expected to vigilantly monitor 14 video streams. Our system simply had to be able to detect dangerous objects autonomously, whereas other teams with smaller numbers of robots could be successful with less automation. At the same time, however, handling more tasks autonomously meant that our human operators had more time to assist with mapping tasks.

VI. Conclusion

The MAGIC 2010 competition showcases the progress that has been made in autonomous, multi-agent robotics. Our MAGIC experience suggests that competitions like these are won by mastering a set of key technological competencies, in this case collaborative state estimation. Our team's focus on global state estimation allowed us to make several contributions to the state of the art in autonomous map building. We believe the quality of our maps was the most important deciding factor that lead our team to win the contest, both because map quality was explicitly an evaluation criterion and because good state estimation supported high-level autonomy throughout our system, resulting in a net reduction in human interaction.

However, MAGIC also highlights the shortcomings of state-of-the-art methods. It remains difficult to maintain a consistent map for large numbers of robots. While our

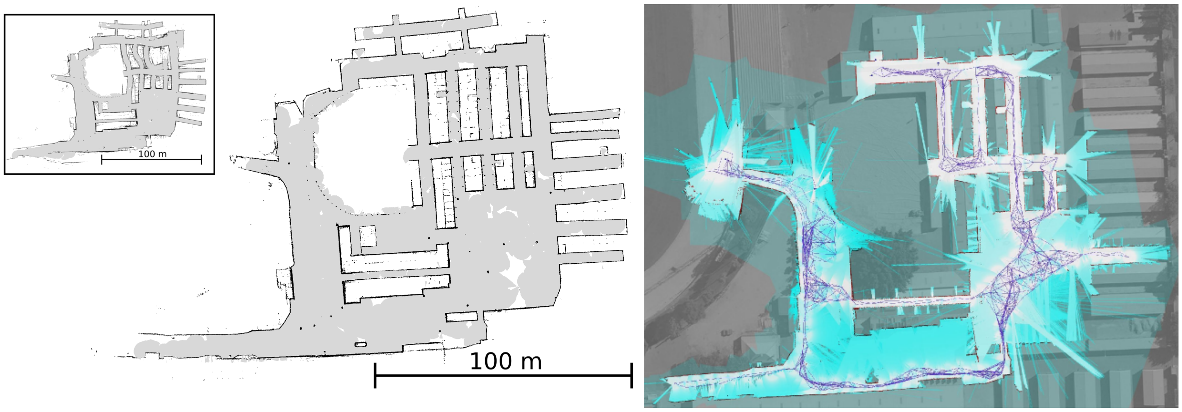

Fig. 6. Comparison of minimally post-processed maps from our team (left) and Magician's mapping algorithm using UPenn's data (right) from [13]. The map we produced online during the challenge is inset top-left.

competition-day maps are fairly good, some distortions are still evident. In particular, the perception systems still add incorrect edges to the factor graph, and current inference methods are highly sensitive to these errors. Our system coped with these errors at the expense of greater operator workload, but further improving these systems remains an important goal for our team.

Ultimately, we feel that competitions like MAGIC 2010, motivated by real-world problems, are invaluable in identifying important open problems and in promoting solutions to them. These competitions serve as a reminder that there are few truly "solved" problems.

Acknowledgments

Team Michigan was a collaboration between the University of Michigan's APRIL Robotics Laboratory and Soar Technology. In addition to the authors of this paper, our core team members included Mihai Bulic, Jacob Crossman, and Bob Mariner. We were also supported by over two dozen undergraduate researchers.

Our thanks also go to the MAGIC contest organizers who mounted a massive effort to organize the competition and an even larger effort to prepare the contest venue. A special thanks goes to our liaison, Captain Chris Latham of the 9th Combat Service Support Battalion in South Australia. Our participation would not have been possible without the help of our sponsors at Intel and Texas Instruments.

References

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the darpa grand challenge," in *The 2005 DARPA Grand Challenge*, ser. Springer Tracts in Advanced Robotics. Springer Berlin / Heidelberg, 2007, vol. 36, pp. 1–43.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. . Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, 2008.

[3] K. Saenbunsiri, P. Chaimuengchuen, N. Changlor, P. Skolapak, N. Danwiang, V. pPoosuwan, R. Tienkum, P. anan Raktrajulthum, T. Nitisuchakul, K. Bumrungjitt, S. Tunsiri, P. Khairid, N. Santi, and S. yan Primee, "RobotCupRescue 2011 - robot league team iRAP JUDY (thailand)," Tech. Rep., 2011.

[4] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Marinier, "Progress towards multi-robot reconaissance and the MAGIC 2010 competition," *Journal of Field Robotics*, To appear.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *Robotics and Autonomous Systems*, June 2006.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[7] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, December 2006.

[8] E. Olson, "AprilTag: A robust and flexible multi-purpose fiducial system," University of Michigan, Tech. Rep., 2010.

[9] ——, "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, June 2009.

[10] M. C. Bosse, "ATLAS: a framework for large scale automated mapping and localization," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2004.

[11] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[12] ——, "Recognizing places using spectrally clustered local matches," *Robotics and Autonomous Systems*, 2009.

[13] R. Reid and T. Braunl, "Large-scale multi-robot mapping in MAGIC 2010," in *Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference on*. IEEE, 2011, pp. 239–244.

[14] J. Butzke, K. Daniilidis, A. Kushleyev, D. D. Lee, M. Likhachev, C. Phillips, and M. Phillips, "The university of pennsylvannia MAGIC 2010 mutli-robot team," *Journal of Field Robotics*, To appear.

[15] A. Lacaze, K. Murphy, M. D. Giorno, and K. Corley, "The reconnaissance and autonomy for small robots (RASR): MAGIC 2010 challenge," *Land Warfare Conference*, 2010.

[16] A. Boeing, M. Boulton, T. Brunl, B. Frisch, S. Lopes, A. Morgan, F. Ophelders, S. Pangeni, R. Reid, and K. Vinsen, "WAMbot: Team MAGICian's entry to the multi autonomous ground-robotic international challenge 2010," *Journal of Field Robotics*, To appear.

[17] A. Erdener, E. O. Ari, Y. Ataseven, B. Deniz, K. G. Ince, U. Kazancioglu, T. A. Kopanoglu, T. Koray, K. M. Kosaner, A. Ozgur, C. C. Ozkok, T. Soncul, H. O. Sirin, I. Yakin, S. Biddlestone, L. Fu, A. Kurt, U. Ozguner, K. Redmill, O. Aytekin, and I. Ulusoy, "Team cappadocia design for MAGIC 2010," *Land Warfare Conference*, 2010.