Mapping and Planning under Uncertainty in Mobile Robots with Long-Range Perception

Pierre Sermanet[1,2] Raia Hadsell[1] Marco Scoffier[1,2] Urs Muller[2] Yann LeCun[1]

(1) Courant Institute of Mathematical Sciences
New York University
New York, NY USA

(2) Net-Scale Technologies
Morganville, NJ USA

***Abstract*— Recent advances in self-supervised learning have enabled very long-range visual detection of obstacles and pathways (to 100 hundred meters or more). Unfortunately, the category and range of regions at such large distances come with a considerable amount of uncertainty. We present a mapping and planning system that accurately represents range and category uncertainties, and accumulates the evidence from multiple frames in a principled way. The system relies on a hyperbolic-polar map centered on the robot with a 200m radius. Map cells are histograms that accumulate evidence obtained from a self-supervised object classifier operating on image windows. The performance of the system is demonstrated on the LAGR off-road robot platform.**

I. Introduction

Building a robust vision-based perception and navigation system for complex outdoor environments is one of the main challenges of field robotics. A key component of such a system is a long-range perception system that can label obstacles and pathways 100 meters away or more. Without long-range vision, a mobile robot acts in a "myopic" fashion, running into cul-de-sacs and systematically exploring long rows of obstacles, not detecting distant pathways that are obvious to human observers. Our group has developed a long-range vision system, described in a companion paper, that accurately labels each pixel in outdoors images into such categories as traversable ground, obstacle foot, and obstacle, as far as 100 meters away. The system uses self-supervised on-line learning to train a classifer operating on feature vectors produced by a convolutional network. One problem, however, is that the range estimates and category labels obtained from a single frame have a considerable amount of uncertainty in the far range.

This paper presents a method for representing spatial and categorical uncertainty in long-range maps. The method allows the accumulation of evidence from multiple frames in a principled manner. The geometry of the map accurately reflects the range uncertainty associated with image-plane obstacle labeling. Furthermore, the map can represent an effectively infinite radius with a finite number of cells. Lastly, the information in the map allows to dynamically adjust the planning policy so as to be more aggressive or more conservative.

Our map representation is based on two key concepts: 1. hyperbolic range mapping; 2. representing categorical evidence by accumulated histograms. In the image plane, a single pixel covers a constant angular extent, but covers a wildly varying range of distances. A single pixel on nearby ground (near the bottom of the image) covers a few cm, while a pixel near the horizon covers an essentially infinite range of distances. For a flat ground plane, the mapping of image-plane pixels to distances is hyperbolic from the bottom of the image to the horizon. For this reason, we propose to represent the environment through a robot-centered *hyperbolic-polar map* (h-polar). This representation allows to map the entire world to a finite number of cells, while being faithful to the type of uncertainty afforded by image-plane labels. To allow the accumulation of evidence for the label of a cell as the robot moves and collects data, each cell in the map contains a *histogram of accumulated probabilities for each category*. At each frame, the classifier produces likelihood values for each category that are accumulated in the histogram in the corresponding map cell. The performance of the system is demonstrated on the LAGR platform (Learning Applied to Ground Robots) which is equiped with two pairs of stereo cameras (Fig 4).

A. Previous Work

The main focus of the LAGR program is the application of machine learning to field robotics [6], particularly for the problem of long-range perception. Mapping with uncertainty has been explored with Bayesian techniques [3] and other methods [12]. We propose a histogram-based approach suitable for multi-class classifiers. Previous work on robot-centered, non-uniform mapping include log-polar representations [10] and multi-resolution grid-based maps [1]. The main advantage of the h-polar approach over these methods is the ability to represent an effectively infinite radius with a finite number of cells. Even more important is a representation of range uncertainty that directly corresponds to that associated with image-plane labeling. Pure image-plane labeling and planning [15], or visual motion planning, presents the advantage of being free of expensive transformations and pose errors, but can only operate with one single frame at the time. The accumulation of evidence over multiple frames can greatly reduce the perception uncertainty and noise that single frame planning is subject to. In addition, visual motion planning relies on the goal being within the current field-of-view, an dangerous assumption in complicated outdoor scenes. A previous version of our system [5] used a robot-centered, tactical Cartesian grid mapping with a radius of 30m. Each time the robot moves and a new video frame is analyzed, the robot-centered map must be translated and rotated before the results from the new frames are incorporated. This process become impracticably expensive for large Cartesian maps. With our new perception system with a range of 100m, a robot-centered Cartesian map with 20cm resolution would have required an impractical 500x500 cells. The h-polar representation offers a considerably more efficient use of memory.

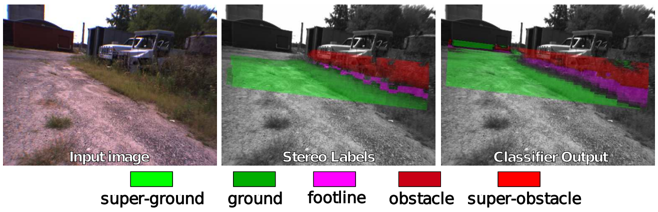

Fig. 1. **Neural network output: 5 classes.**
The input image (left) is fed into the neural network for classification at resolution 512x384, producing traversability labels (overlaid on the right image) up to 100 meters and more. The last layer of the neural network is trained online, using the stereo labels (center) up to 12 meters as training information.

II. Long Range Vision

The existing paradigm for vision-based mobile robots relies on hand-tuned heuristics: a stereo algorithm produces a (x,y,z) point cloud and traversability costs are assigned to points based on their proximity to a ground plane [8], [4]. However, stereo algorithms that run in realtime often produce costmaps that are short-range, sparse, and noisy. Our learning strategy uses these stereo labels for supervision to train a realtime classifier. The classifier then predicts the traversability of all visible areas, from close-range to the horizon. For accurate recognition of ground and obstacle categories, it is best to train on large, discriminative windows from the image, since larger windows give contextual information that is lacking in color and texture features. Other research has explored the use of online learning for mobile robots, but their methods have been largely restricted to simple color/texture correspondences [9], [11], [14]. See [5] for more details about the long-range vision module.

A. Classifier Output

The network produces a floating-point value for each of the 5 following classes (Fig 1) detected by the stereo module: super-ground, ground, footline, obstacle, super-obstacle. Having more visually-consistent classes helps the network to produce better classification in comparison to a binary classifier as used in previous versions of the system. In addition, it conveniently suits the histogram scheme used in our planning cost decision algorithm described in the following section.

III. Label uncertainty: Cost from Histogram

Label uncertainty can be caused by differences of view points, sensor noise or learning phase of the online neural network. Mapping with label uncertainty in mobile robotics has been addressed by several approaches from Bayesian techniques [3] to fuzzy logic [12]. Trivial approaches such as using latest labels only or running averages are simple and fast but lack confidence and accuracy and would cause an early fusion of the multi-class network outputs. We use an histogram approach (Fig 2) suited for multiple classes and able to delay the traversibility decision to planning time: each cell constains K bins, each bin corresponding to a class (or a range of traversibility for single class outputs). Each new label is merged in a cell through a simple addition. Before planning, the histogram is translated into a traversibility cost (Fig 3). At that time, the traversibility decision can be modulated by the current planning policy: conservative vs. aggressive. As label uncertainty increases with distance, a distance decay must be applied to incoming frames.

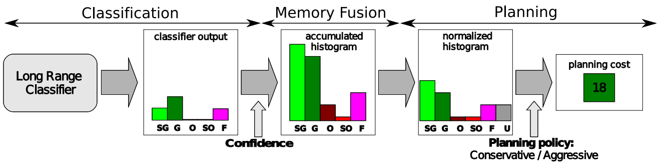

Fig. 2. **Histogram to cost process.**
The output of the classifier is multiplied by the current learning confidence and added to an existing histogram. Before converting the histogram to a planning cost, it is normalized. Finally, the current planning policy modulates the mapping from the normalized histogram to a single planning cost.

A. Histogram to cost transformation

Let $\sum_k bin_k + c$ be the sum of all bins of an histogram with an added constant c which brings confidence to 0 when the sum is small. Let $\sum_k w_k bin_k$ be the weighted sum of all bins. The normalized sum S tuned by gain parameters γ is defined by:

$$S = \gamma \frac{\sum_k w_k bin_k}{\sum_k bin_k + c}$$

S is then used to compute the planning cost $cost$:

If $(S <= 0)$
$\quad cost = cost_{unexplored} + S * (cost_{unexplored} - cost_{min})$
Else if $(S <= 1)$
$\quad cost = cost_{unexplored} + S * (cost_{lethal} - cost_{unexplored})$
Else $cost = cost_{lethal}$
$cost = MAX(cost, cost_{min})$

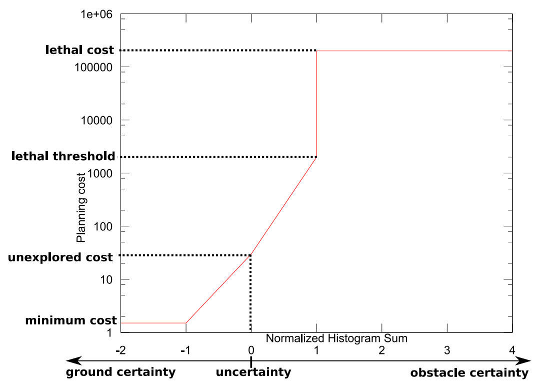

Fig. 3. **Planning cost mapping from normalized histogram sum.**
A sum of 0 represents the full uncertainty, -1 and below are given the minimum ground cost, and 1 or higher the maximum cost.

IV. Range uncertainty: Hyperbolic polar map

Cartesian maps are commonly used in autonomous robots to represent the environment. While this space is convenient for implementation, it is computationaly impractical for long-range vision applications and does not naturaly suit the image resolution distribution. Visual motion planning [15] has its limits. Log-polar mapping [10] and multi-resolution grid-based mapping [1] cannot represent range up to infinity because of their geometric growth. We propose a pseudo-image space approach named hyperbolic-polar (or h-polar) able to handle infinitely long range vision and planning in an efficient and natural manner. Along with the histogram approach, this hyperbolic mapping allows to build confidence over multiple input frames and provide more stable maps than cartesian or linear-polar representations for vision applications. It is important to note that the hyperbolic distribution allows to map infinity in a limited number of cells (cf. Fig. 7).

A. Overall Process

Overall processing of the h-polar map from image labeling to planning:

1) **Image labeling (Fig. 4):** Each pixel of the input images (512x384 in our system) is labeled with an histogram, representing confidences in each class.
2) **Plane extraction (Fig. 4):** A single plane is found using the short-range stereo points (up to 12m). Remaining points with no disparity information are projected onto the single plane and given a disparity. (r,c,d) is then converted to vehicle-centered (x,y,z).
3) **Point cloud transmission:** The point cloud of $(x,y,z,histogram)$ is sent over to the h-polar module.
4) **Recentering memory:** Previously accumulated frames are recentered onto the last available pose.
5) **Adding new point cloud:** Each point of the new point cloud is merged into the h-polar internal point cloud.
6) **Produce planning map (Fig. 5):** A temporary map for planning and display is produced from the h-polar point cloud.
7) **Planning:** Best paths from the goal to all points of the map are computed and sent to the reactive planning layer. Optimal path is shown in white in (Fig. 5).

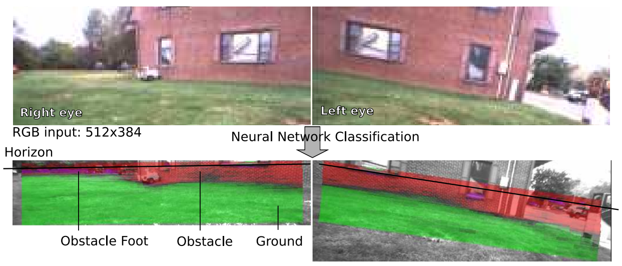

Fig. 4. **Plane extraction and traversibility classification.** To assign disparity information to non-stereo points, a single ground plane is found using the stereo points. This ground plane is also used to normalize 2D bands of images to be fed to the neural network for classification.

B. Geometry Details

The h-polar space is not the exact image space given by the real cameras of the robot, but rather a pseudo-image space defined by the pseudo-camera parameters h_{cam} and hR_{min} (cf. Fig. 6) and able to integrate multiple frames together.

As shown on Fig. 6, the h-polar uses 2 different resolution distributions in the radial dimension:

- From the center of the map to the first hyperbolic cell, the resolution is **constant**, each cell has the same radial size.

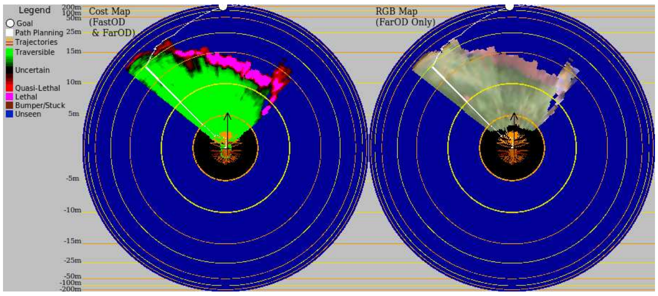

Fig. 5. **H-polar cost (left) and RGB (right) maps.** Groundplane-projected pixels with their traversability labels from left and right eyes are merged together with previous frames into the h-polar map. Once recentered onto the vehicle center, driving costs are computed and planning takes place in the cost map (left). The resulting optimal path in white leads to the goal (white blob). The **RGB h-polar** map (right) is used only for display and is very convenient for humans to figure out what the robot is looking at and how good the classification is. The vehicle is alway facing the top of the map. The orange curves in the 2.5m radius represent the **currently feasible trajectories of the reactive low-level planner** taking in account the vehicle dynamics.

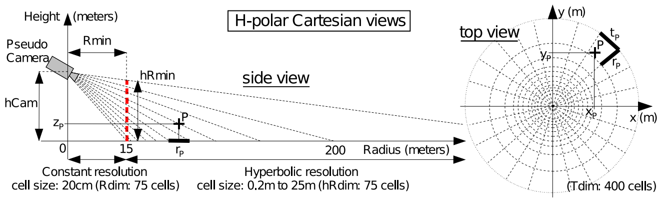

Fig. 6. **H-polar side and top views.** The radial resolution is constant from 0 to 15 meters with 20cm radius. Then from 15m to 200m and more, cells range from .2m to 25m. Integers r_P and t_P are the h-polar coordinates of the cell into which cartesian point $P(x_P, y_P, z_P)$ falls.

- From the first hyperbolic cell to infinity, the radial cell size increases in an **hyperbolic** manner (equation of r_P is of type $\frac{1}{radius}$).

The radius of the first hyperbolic cell is determined based on the desired cell size C_{res} in the constant resolution area. In our system, we estimated a cell size of 20cm would be sufficient for our purpose. Thus to keep continuity in the cell sizes between the constant and the hyperbolic area, the first hyperbolic cell must begin at a radius R_{min} of 15.26m, given the parameters h_{cam} and hR_{min} of our pseudo-camera and hR_{dim}, the hyperbolic radial dimension. R_{min} is determined by the following equation:

$$R_{min} = C_{res} * (\frac{h_{cam} * hR_{dim}}{hR_{min}} - 1)$$

Following are the values used in our system in the real-life tests conducted the LAGR Government Team in January 2008:

C_{res}	=	$0.2m$,	cell radial size in the constant area.
R_{min}	=	$15.26m$,	radius of the first hyperbolic cell.
h_{Cam}	=	$1m$,	height of the peusdo-camera.
hR_{min}	=	$0.97m$,	height of the pseudo-image at radius R_{min}.
R_{dim}	=	75,	cells in the constant area.
hR_{dim}	=	75,	cells in the hyperbolic area.
T_{dim}	=	400,	cells in the angular dimension.

Given those parameters, Fig. 7 plots the cell index versus the cell radius. Note that cells range up to 350m. In practice however, because uncertainty increases with distance and because only 150 cells are used in the radial dimension, the system is considered to be accurate up to 100m. But no there are no theoretical boundaries to this limit, which can be increased as computation budget permits.

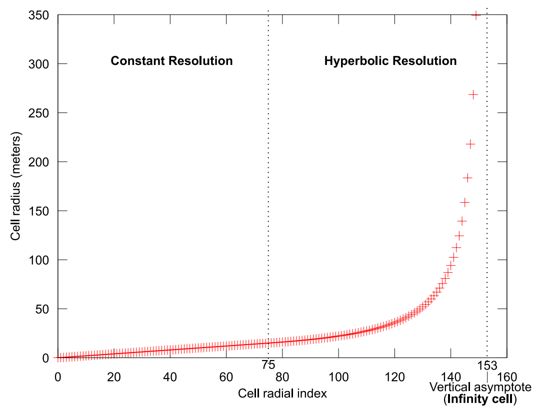

Fig. 7. **H-polar cell radius versus cell radial index** with current configuration. Given our pseudo-camera and h-polar parameters, the radial resolution is constant (20 cm) from cell 0 to 74 with radiuses ranging from 0 to 15.06 meters. From cell 75 to 150, the resolution is hyperbolic and radiuses range from 15.26m to 350m. Cells sizes range from .2m to 80m. If few more radial cells were added to our map, infinity would lay in cell 153.

1) Cartesian to H-polar Transformation (x,y,z) to (r,t,z): Transformation formulas (for the hyperbolic area) from cartesian coordinates (x_P, y_P, z_P) in local pose coordinate system to h-polar coordinates (r_P, t_p, z_P) (z_P remains unchanged) of point P (Fig. 6) :

$$r_P = \left(\frac{R_{min} * (z_P - h_{Cam})}{\sqrt{x_P{}^2 + y_P{}^2}} + h_{Cam}\right) * \frac{hR_{dim}}{hR_{min}} + R_{dim}$$

$$t_P = arctan\left(\frac{y_P}{x_P}\right) * \frac{T_{dim}}{2\pi}$$

Those transformations are called many times. To remain efficient, their computation must be optimized: precomputing a look-up table of all the discrete T_{dim} useful values resulting from the *arctan* function provides a non-negligible speed-up. Note also that only the integer part of (r_P, t_P) is used to determine which cell a point falls into.

2) H-Polar to Cartesian Transformation (r,t,z) to (x,y,z): Transformation formulas (for the hyperbolic area) from h-polar (real) coordinates (r_P, t_p, z_P) of point P (Fig. 6) to cartesian coordinates (x_P, y_P, z_P) in local pose coordinate system (z_P unchanged):

$$radius_P = \frac{R_{min} * (z_P - h_{Cam})}{(r_P - R_{dim}) * \frac{hR_{min}}{hR_{dim}} - h_{Cam}}$$

$$\alpha = t_P * \frac{2\pi}{T_{dim}}$$

$$x_P = \cos\alpha * radius_P \qquad y_P = \sin\alpha * radius_P$$

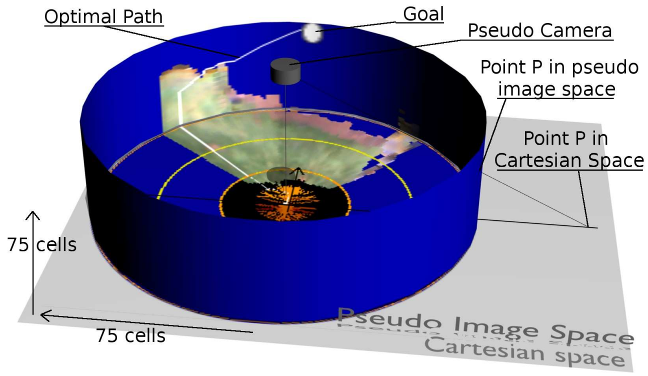

Fig. 8. **RGB h-polar map in pseudo-image and Cartesian spaces.**
The center part of the h-polar map has a constant radial resolution whereas the cylindrical part has a hyperbolic radial resolution or pseudo-image space. When flattened down into cartesian space, the cylindrical part takes hyperbolically more space.
To transform coordinates between the pseudo-image space and the cartesian space, a pixel is projected onto the Cartesian space along the line coming from the omniscient camera at the center of the map at height *hCam*. The reverse transform projects a point from Cartesian space into the pseudo-image space.

C. Implementation Details

1) Hyperbolic-polar Organized Point-Cloud: Internally, h-polar is a point cloud, each point being defined by $(x, y, z, histogram, counter)$. For practical reasons, the number of points must be bound and it is desirable to keep a uniform distribution of points in the map. Therefore, points are kept using the h-polar distribution, i.e. 1 point per cell, in an array of points of dimensions $((R_{dim} + hR_{dim}) * T_{dim} * Point_{dim})$. The dimensions of this map do not have to be the same as the final cost map used for planning. In fact, there are 4 times more cells internally than in this final map. To avoid accumulation of integer imprecisions during recentering and frame additions, the exact (x, y, z) coordinates are kept all times. Only the integer part of (r, t) is used to find the target cell of a point.

2) Translation and Rotation: The map is internally fixed onto the local pose coordinate system. It never needs to be rotated except for display. The rotation is extremely simple, a single (ring) offset t_{offset} is added to the angular index t before access. When recentering, no rotation is necessary, only translation is applied as described in the following section.

3) Recentering: Before adding a new frame to the current memory, the latter needs to be recentered to take in account the movement given by the latest available pose. As pointed out above, recentering of the h-polar map only requires translation of the (x, y) components of each point:

1) Compute translation $(x, y)_{trans}$ between the memory's pose and the last known pose in the local pose coordinate system (z variations are ignored):

$$(x, y)_{trans} = (x, y)_{memPose} - (x, y)_{lastPose}$$

2) For all $((R_{dim} + hR_{dim}) * T_{dim})$ points:
   - Add $(x, y)_{trans}$ to (x, y) components of point.
   - Move point into new h-polar cell: if the new cell exists, merge point into that cell, else delete point.

After recentering, the current angular offset t_{offset} of the heading of the vehicle ($\alpha_{lastPose}$) must be saved in order to render a map fixed to the vehicle coordinate system:
$t_{offset} = \alpha_{lastPose} * \frac{T_{dim}}{2\pi}$
Later, t_{offset} simply needs to be added to each t index before accessing the cells for render.

4) Adding a New Frame: A new frame comes as a cloud of $(x, y, z, histogram)$ points centered on the vehicle's coordinate system at the time of the input image capture. Before adding new points to the memory (already recentered with last known pose), translation and rotation need to be applied to take in account the last known position of the robot:

$$(x, y)_{trans} = (x, y)_{imgPose} - (x, y)_{lastPose}$$

The rotation corresponds to the heading $\alpha_{imgPose}$ of the robot in the local pose coordinate system, given by *imgPose*, the pose of the image. Hence the following transformation matrix (z remains unchanged and can be ignored) produces the centered coordinates $(x, y)'$:

$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} \cos\alpha & -\sin\alpha & x_{trans} \\ \sin\alpha & \cos\alpha & y_{trans} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

To add the recentered new frame to the recentered memory, simply loop through all points of the image point cloud and merge each into their corresponding h-polar cell.

5) Points Merging: When recentering memory or adding a new frame, several points may fall into a same h-polar cell and need to merge together since only one point per cell is kept. There are two different merging operations to consider:

- **Data merging:** Using the histogram scheme, merging is a simple addition. Keeping a counter of the number of points merged can be useful to normalize the RGB display and for coordinates merging.
- **Coordinates merging:** The new coordinates can be an interpolation of each point's coordinates weighted by their counter, thus giving more importance to points seen many times. However in our system, because of a low computation budget, coordinates are simply overwritten by the last added point.

6) Blocking after Obstacles: To avoid projecting down to the ground the part of obstacles for which a z estimate is missing, typically points beyond stereo range, those points are "blocked" after an obstacle. For example, a tree trunc can appear as a giant path or obstacle in the map if not blocked after its foot. Therefore, all points without z information following an obstacle in the radial direction are ignored when adding a new frame.

7) Subsampled and Smoothed Mapping for Planning: The internal high resolution h-polar map is scattered because points can merge during the translation process. This can produce erratic path planning between unknown cells and create holes inside lethal obstacles. To cope with this issue, a lower resolution map is produced for the actual planning: every low-resolution cell is a sum of 4 neighboring cells. In addition to subsampling, as some holes may subsist, the 8 neighbors of each cells are added, thus providing a smooth dense map. Note that no normalization is required since the histogram scheme handles addition of multiple histograms by modulation of confidence.

D. Registration

In comparison to traditional cartesian representations, the pseudo-image space representation could allow to efficient registration of patches to correct for pose imprecisions.

E. Limitations and Further Work

Current limitations prevent the h-polar representation from offering a complete mapping solution. Those shortcomings may be resolved in further research:

- **Lack of global map**: h-polar is used as a vehicle-centered local map and information fades away when the robot moves. Thus planning is limited to the h-polar radius and precise maps of visited areas may be lost. Coupling h-polar with a grid based global map or using a single quad-tree global map as in [7] may address this problem.
- **Points coverage (merging and splitting):** It is assumed that new data appears in front of the vehicle when moving forward, then merging with previous data. Points moving outside of the field-of-view into cells covering smaller areas scatter into tiny dots when they used to represent a bigger area. Merging and splitting heuristics may be improved to cope with those issues but it is not clear if it can solve the issue and at what price. However, this is in practice rarely an issue thanks to subsampling, smoothing and because points appear in front of the vehicle. Thus most points outside of the field-of-view, compress together when moving away from the center.
- **Ground plane assumption:** Although 3D coordinates are kept internally, planning is executed in a flat 2D world. Z costs could be incorporated in the planning to reflect terrain relief.

V. Planning using H-polar

A. Grid-based Planning with Pre-computed Transition Costs

Any grid-based planning algorithm can be used here by using the pre-computed transitions costs between the center of each cell to their 8 neigbors (the transition cost is simply the euclidian distance of the (x, y) real coordinates of each cell's center).

B. Layered Planning

We use a layered architecture [2] to allow decoupled levels of planning in the system [13]: reactive short-range planning which takes in account the vehicle dynamics, and deliberative long-range planning. The reactive planner runs between 5 and 10 Hz independently of the h-polar planner which runs at 1Hz. Thus, all possible paths need to be precomputed in h-polar once, in order to provide the the reactive planner all optimal paths to the goal at every fast iteration. We use the Dijkstra algorithm from the goal to every cell of the h-polar map as our single-source to multi-targets planner. Thus every h-polar cell contains the optimal path to the goal. The reactive planner computes the cost from the vehicle center to a finite number of candidates at a certain radius (5m), then queries and add the remaining cost for each of those candidates, beforehand translated into h-polar coordinates. The candidate with minimum cost contains the optimal path. Finally, the list of wheel commands of the trajectory that initiated the best path is executed until the next iteration.

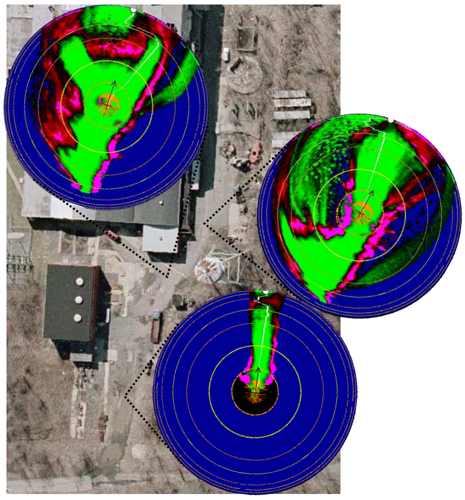

Fig. 9. **H-polar maps along heterogeneous outdoor course.**
Each h-polar map accurately represents the real environment shown in the sattelite image, and plans around obstacle from an approximate distance of 50 meters.

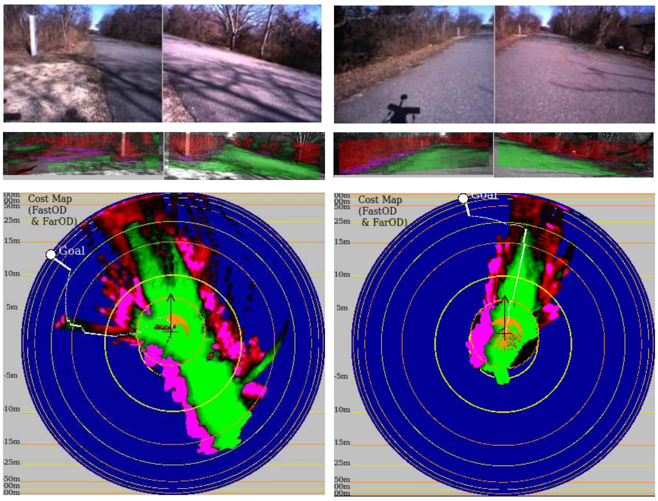

Fig. 10. **Result runs with long-range vision activated in Sandy Hook.**
The top images are left and right input images. The middle images show classification for left and right. At the bottom is the corresponding h-polar map.

VI. Results

The complete system with long-range vision was intensively tested in outdoors environments and consistently showed improvement over the same system with long-range vision disabled. Here are two examples of runs conducted in Sandy Hook, NJ. (Fig 10): each course was run with the long-range module on or off. When the module was off, the robot took 321s and 196.1s to reach the goal with a travel distances of 271.9m and 207.5m and 3 and 1 human interventions to get the robot out of trouble. With the long-range vision enabled, it took the robot 155.5s and 142.2s to reach the goal, travelling 166.8m and 165.1m and it required no human intervention for neither runs.

VII. Conclusion and Further Work

Long-range data and its uncertainties were successfully integrated and used to perform real-time long-range navigation. Using a layered planning architecture with low-level reactive obstacle detection including vehicle dynamics and a high-level deliberative long-range and hyperbolic-polar planner, the system reliably performed during intensive field tests by an independent testing team. For tactical planning architectures, where the local map is tied to the vehicle coordinate system and needs to be recentered at every iteration, h-polar mapping provides a computationaly efficient way to handle always increasing vision range as well as classification and range uncertainties. However the h-polar mapping is limited by the lack of global map. It is questionable whether there is a need of a high-level tactical map when a lower-level perception and planning module already serves this purpose. Further work would explore the global multi-resolution direction given by [7] with a global hyperbolic-polar-like mapping to replace the current h-polar tactical mapping.

Acknowledgements

This work was supported in part by the DARPA LAGR program under contract HR001105C0038.

References

[1] S. Behnke. Local multiresolution path planning. In *Robocup 2003: Robot Soccer World Cup VII*, 2004.
[2] R. Brooks. A robust layered control system for a mobile robot. In *Robotics and Automation, IEEE Journal of [legacy, pre-1988]*, 1986.
[3] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Autonomous Mobile Robots: Perception, Mapping, and Navigation (S. S. Iyengar and A. Elfes, eds.), IEEE Computer Society Press, pp. 6071*, 1991.
[4] S. B. Goldberg, M. Maimone, and L. Matthies. Stereo vision and robot navigation software for planetary exploration. In *IEEE Aerospace Conf. Proc.*, March 2002.
[5] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, J. Han, B. Flepp, U. Muller, and Y. LeCun. Online learning for offroad robots: Using spatial label propagation to learn long-range traversability. In *rss*, 2007.
[6] L. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan. The darpa lagr program: Goals, challenges, methodology, and phase i results. In *Journal of Field Robotics 23(11/12), 945973*, 2006.
[7] S. Kambhampati and L. Davis. Multiresolution path planning for mobile robots. In *Robotics and Automation, IEEE Journal of [legacy, pre - 1988] Volume 2, Issue 3, Page(s): 135 - 145*, 1986.
[8] A. Kelly and A. Stentz. Stereo vision enhancements for low-cost outdoor autonomous vehicles. *ICRA Workshop WS-7*, May 1998.
[9] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and B. A. F. Traversibility classification using unsupervised on-lne visual learning for outdoor robot navigation. In *IEEE Int'l. Conf. on Robotics and Automation (ICRA)*, May 2006.
[10] L. Longega, S. Panzieri, F. Pascucci, and G. Ulivi. Indoor robot navigation using log-polar local maps. In *International IFAC Symposium on Robot Control*, 2003.
[11] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robot*, 18:81–102, 2003.
[12] Oriolo, G. Ulivi, and M. Vendittelli. Fuzzy maps: A new tool for mobile robot perception and planning. In *Journal of Robotic Systems, 14(3), 179-197*, 1997.
[13] P. Sermanet, R. Hadsell, J. Ben, A. N. Erkan, B. Flepp, U. Muller, and Y. LeCun. Speed-range dilemmas for vision-based navigation in unstructured terrain. In *Proc. 6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.
[14] B. Sofman, E. Lin, J. Bagnell, N. Vandapel, and A. Stentz. Improving robot navigation through self-supervised online learning. In *Proc. of Robotics: Science and Systems (RSS)*, June 2006.
[15] H. Zhang and J. Ostrowski. Visual motion planning for mobile robots. In *Robotics and Automation, IEEE Transactions*, 2002.