Continuous Humanoid Locomotion over Uneven Terrain using Stereo Fusion

Maurice F. Fallon[1], Pat Marion[2], Robin Deits[2], Thomas Whelan[3], Matthew Antone[2], John McDonald[3] and Russ Tedrake[2]

***Abstract*— For humanoid robots to fulfill their mobility potential they must demonstrate reliable and efficient locomotion over rugged and irregular terrain. In this paper we present the perception and planning algorithms which have allowed a humanoid robot to use only passive stereo imagery (as opposed to actuating a laser range sensor) to safely plan footsteps to continuously walk over rough and uneven surfaces without stopping. The perception system continuously integrates stereo imagery to build a consistent 3D model of the terrain which is then used by our footstep planner which reasons about obstacle avoidance, kinematic reachability and foot rotation through mixed-integer quadratic optimization to plan the required step positions. We illustrate that our stereo imagery fusion approach can measure the walking terrain with sufficient accuracy that it matches the quality of terrain estimates from LIDAR. To our knowledge this is the first such demonstration of the use of computer vision to carry out *general purpose terrain estimation* on a locomoting robot — and additionally to do so in continuous motion. A particular integration challenge was ensuring that these two computationally intensive systems operate with minimal latency (below 1 second) to allow re-planning while walking. The results of extensive experimentation and quantitative analysis are also presented. Our results indicate that a laser range sensor is not necessary to achieve locomotion in these challenging situations.**

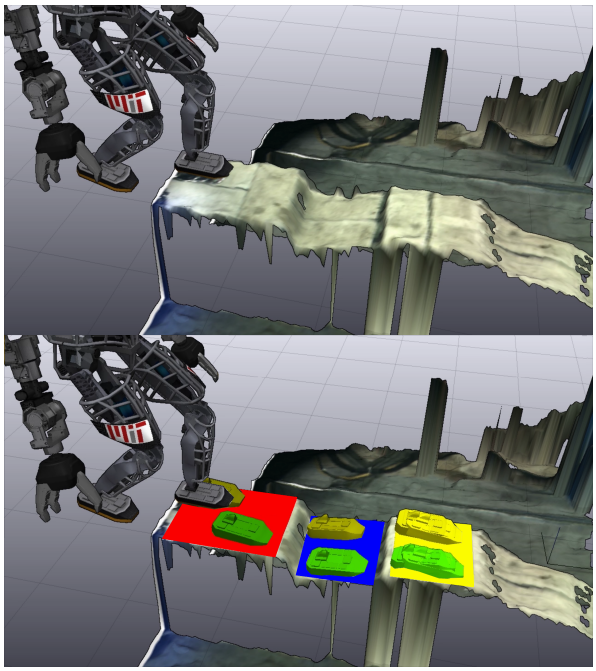

Fig. 1. As the robot walks over the terrain, its visual mapping system builds a dense reconstruction of the environment ahead. Once per step, the planning system captures the reconstruction, detects convex regions which are large enough to contain a foot and determines suitable footstep placements to be executed by the walking controller. Note that self observations of the robot by itself are filtered and that footsteps are rotated to match the local terrain normal.

I. INTRODUCTION

A primary motivation for humanoid robotics research is to develop platforms capable of moving through the same environment as a human being – such as squeezing through confined spaces and under overhanging obstacles as well as crossing challenging terrain. While locomotion research spans actuator development, mechanism development, dynamic planning and control, in this work we focus on terrain estimation and footstep planning.

We take as motivation the recent DARPA Robotics Challenge (DRC) [1], where robots in outdoor conditions were required to walk over a course of uneven and discontinuous terrain. For humanoid robots to be useful, this kind of walking task must be automated such that the robot can locomote around or over any obstacles without stopping.

On-line footstep planning using some form of visual feedback has been demonstrated by a number of research groups. For example Lorch et al. [2] used a single camera to estimate the layout of obstacles in front of the robot and used an on-line footstep planner to search for possible forward strides to clear obstacles. Only forward motion was considered by the footstep planner, and the ground was assumed to be flat. Asatani et al. demonstrated the use of stereo vision to inform footstep planning for a biped robot [3]. The researchers developed a method to detect the edge between two planar regions, and then modified an existing footstep plan to avoid the seam between the two planes. Later work extended the capability to detect the edges of steps so as to execute footsteps precisely placed on a narrow staircase. A single camera identified colored patches on a flat floor which were designated as obstacles, and a footstep planner based on the graph-search approach of [4] then computed safe footstep plans every time the robot took a step.

Gutmann et al [5] more generally explored visual edge detection to carry out plane segmentation using stereo point

*This work was supported by the Defense Advanced Research Projects Agency (via Air Force Research Laboratory award FA8750-12-1-0321).

[1]With the School of Informatics, University of Edinburgh, UK. maurice.fallon@ed.ac.uk

[2]With the Computer Science and Artificial Intelligence Laboratory, MIT, MA 02139, USA. pmarion, rdeits, mantone, russt@mit.edu

[3]With the Department of Computer Science, Maynooth University, Ireland. thomas.j.whelan@mumail.ie, johnmcd@cs.nuim.ie

clouds and later LIDAR. They also used visual SLAM to estimate motion.

A common aspect in these works was simplification of the required visual processing — for example using distinctive color to allow simple segmentation or detecting visible edges using line detectors which typically required regular re-mapping when the robot stopped walking.

Finally [6], used a terrain map generated by a real stereo camera to explore the adaptation of walking trajectories (rather than footstep placement.

Meanwhile there has been significant recent progress of dense mapping methods using active RGB-D cameras [7], [8]. However deficiencies in these sensors (in particular rolling shutters and limited applicability in outdoor environments) and the computational burden of the proposed algorithms has limited their use on humanoid robots.

In this work we build upon these two different research fields by introducing a more thorough terrain estimation system using dense methods and passive stereo vision and then combine it with a new type of footstep planner based on efficient mixed-integer optimization. From the estimated 3D map of the terrain in front of the robot, we extract all planar regions large enough for the robot to place a foot. These planar regions are then used as input to the general-purpose 3D footstep planner presented by Deits and Tedrake [9] which chooses a footstep plan which is globally optimal with respect to a cost function informed by the length of each step, number of steps taken, and distance of the robot's final pose from a desired goal (within our approximations of the terrain geometry and discretized rotations). This entire process is repeated every time the robot lifts a foot, allowing the footstep plans to continually improve as more terrain comes into view.

The research of [10] is the most closely related to ours. The authors demonstrated continuous locomotion similar to our approach but used an actively swinging LIDAR sensor mounted on the robot's torso. Vision sensing (in particular RGB-D sensors) is displacing LIDAR thanks to high frame rates and resolutions which allow robots to be more responsive, to operate in dynamic situations and to avoid complex sensor actuation mechanisms. This motivates our usage of computer vision.

Aside from our demonstration that the LIDAR sensor is not necessary for localization; as mentioned above our footstep planner uses continuous optimization, as opposed to search–based planning in [10], which provides more flexibility when operating in rough terrain with few feasible footstep configurations. Finally, the authors of [10] also give some consideration to longer duration path planning, which we have not as yet explored.

We first discuss our footstep planner assuming sensor-agnostic input before describing the stereo fusion algorithm. (Although the system described works just as well using a laser range-finder to provide the input terrain map.) In Sect. II we outline our approach to footstep planning and discuss how the required inputs to our footstep planner can be extracted from a generic terrain map of the area in front of the robot (Sect. II-A).

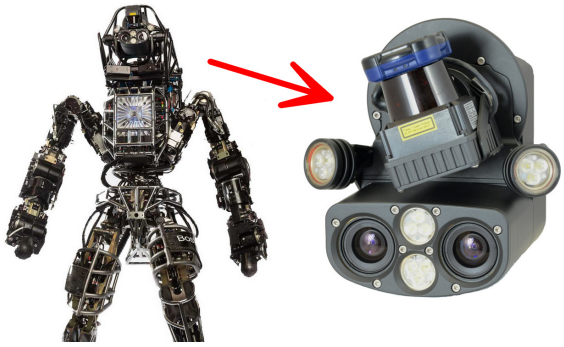

Fig. 2. The Atlas robot's primary sensing is provided by the Carnegie Robotics Multisense SL sensor head which is equipped with a stereo camera and a rotating LIDAR scanner. In this work we use only the stereo camera, except in Sect. IV-A where we compare the terrain estimated by the stereo camera to that estimated using the LIDAR. (photo credits: Boston Dynamics and Carnegie Robotics)

We then provide an overview of recent research in dense mapping (Sect. III) using active RGB-D cameras, before presenting the modifications and filtering necessary to allow dense passive stereo data to produce terrain maps suitable for footstep planning.

Finally in Sect. IV a series of successful experiments compellingly demonstrate the combination of the perception and planning modules. As such this work presents a significant first demonstration: using general purpose footstep planning to locomote over challenging uneven and disconnected terrain using passive stereo fusion to estimate that terrain — all while continuously moving.

We conclude Sect. IV with a quantitative comparison between terrain maps produced by passive stereo and LIDAR.

II. Receding Horizon Footstep Planning

The footstep planning problem typically involves choosing an ordered set of foot positions for the robot, subject to constraints on the relative displacement between those footsteps, in order to bring the robot close to some desired goal pose. Typically, these footstep locations must be chosen in order to avoid some set of obstacles in the environment. Performing a smooth optimization of footstep poses while avoiding obstacles tends to introduce non-convex constraints which can make globally-optimal solutions extremely difficult to find [11].

A common approach among footstep planners is to discretize the set of possible foot transitions, expressed as the relative displacement from one foot pose to the next. From this set of discrete actions, a tree of possible footstep plans can be constructed and searched using existing search algorithms such as A*, D*, and RRT [12], [10], [13], [14], [15]. These techniques, however, are limited by the discretization of the footstep actions, since a small number of actions severely limits the robot's possible footstep plans, while a large number of actions creates an extremely large search space. Choosing an informative and admissible heuristic for footstep planning problems can be very difficult [16].

Recent work by Deits and Tedrake reversed the problem of obstacle avoidance into one of assigning footsteps to some pre-computed safe regions [9]. If the regions of safe terrain are convex, then the requirement that a footstep remain within some safe region is a convex constraint. A problem consisting of convex constraints (and a convex objective function) can typically be solved to its global optimum extremely efficiently [17]. Combining such a problem with a discrete choice of the assignment of footsteps to safe regions results in a mixed-integer convex program with high worst-case complexity but which can often be solved to global optimality efficiently with modern tools and techniques [18], [19].

We choose to use the mixed-integer convex optimization described by Deits and Tedrake in [9] and available from the Drake toolbox [20], to quickly produce optimal footstep plans for an environment which has been decomposed into convex regions of safe terrain. Rather than operating directly on the point cloud data provided by the LIDAR or stereo system, the footstep planner requires only a description of each safe region as a planar area in 3D. Thus, when planning footsteps, the entire perception system can be abstracted away into a tool which produces regions of safe terrain. These regions, along with a desired navigation goal pose, are used as input to the footstep planning optimization, which chooses the number of footsteps to take and the poses of those footsteps. Note that the planner itself decides online how many footsteps to place in each convex region — which varies with the size of the region, the robot configuration and layout of the upcoming terrain.

The procedure by which these planar regions are created is described in full detail in the next section.

A. Continuous Walking Autonomy

The continuous walking routine used by our experiments described in Sect IV-C is detailed with pseudo-code in Algorithm 1. The routine uses a point cloud segmentation algorithm to find footstep regions and a *"carrot on a stick"* approach to generate navigation goals that lead the robot forward over the terrain course. Short-horizon footstep plans within the safe regions are computed and sent to the manufacturer's walking controller for execution.

The routine *OnFootLiftoff* is called once to begin walking, and again each time a foot lifts off from the ground during walking. The first stage computes the robot configuration q that will be used as input to the footstep planner. Because the robot is walking continuously and planning on-line, the configuration q used for planning is chosen to be the next double foot support of the robot after the current swing foot has landed, i.e., the configuration the robot is expected to achieve in the very near future upon completion of the current walking step. The configuration q is a list of the robot's joint positions in generalized coordinates plus the 6 degree of freedom position and orientation of the robot's base link in the world coordinate system. The function *LandingConfiguration* uses inverse kinematics to find the next double support configuration using the current stance foot and the next target footstep f_{step}.

Next, the $TerrainSegmentation$ function described in Algorithm 2 is used to find convex regions for the footstep planner. The point cloud segmentation algorithm is capable of processing sensor terrain data and is fully agnostic as to whether the input data comes from LIDAR or stereo 3D reconstruction. Analysis of the performance differences between the two data sources is found in Sect IV-A. The interested reader is directed to [5] for fuller discussion of walking region segmentation and to [21] for work on more advanced curved region segmention.

Our segmentation algorithm models the terrain course as a series of admissible planar convex regions. The robot was commanded to step only on these regions as it makes progress. Because our focus was not on general navigation but on traversal of the complex cinder block stacks shown in Fig 8, the robot simply halts when it reaches the end of the course.

Next, we apply point cloud surface normal estimation using a local search neighborhood of 5 cm around each point. The points are filtered to keep only those within 30 degrees of horizontal according to surface normals. Steeper regions are deemed infeasible for footsteps.

Algorithm 1 Continuous walking algorithm

procedure ONFOOTLIFTOFF(q, f_{step})
  ▷ Given footstep frame f_{step}, or nil to bootstrap
  if f_{step} **then**
    $q_{next} \leftarrow LandingConfiguration(q, f_{step})$
  else
    $q_{next} \leftarrow q$
  $p \leftarrow PointcloudSnapshot()$
  $r_{safe} \leftarrow TerrainSegmentation(p, q)$
  if $empty(r_{safe})$ **then**
    $ContinueQueuedFootstepPlan()$
  else
    $f_{goal} \leftarrow NavigationGoal(q, r_{safe})$
    $f_{plan} \leftarrow FootstepPlan(q_{next}, r_{safe}, f_{goal})$
    if $IsValidPlan(f_{plan})$ **then**
      $QueueFootstepPlan(f_{plan})$
    else
      $ContinueQueuedFootstepPlan()$

Algorithm 2 Terrain segmentation algorithm

procedure TERRAINSEGMENTATION(p, q)
  ▷ Given point cloud p, Robot configuration q
  $scene \leftarrow RemoveGroundPoints(p)$
  $planarPoints \leftarrow FilterByNormal(scene)$
  $clusters \leftarrow ExtractClusters(planarPoints)$
  $regions \leftarrow []$
  for each c in clusters **do**
    $regions \leftarrow ComputeConvexSafeRegion(c)$
  return $regions$

The filtered points are input to a Euclidean clustering routine that finds individual connected components of uniform planar segments. Each planar cluster is converted to a convex region for input to the footstep planner. These convex regions are shown in Fig 1.

Next, the $NavigationGoal$ returns a navigation goal 1 meter beyond this region. The regions and navigation goal are passed to the footstep planner to compute a footstep plan to navigate towards the goal. The resulting footstep plan is immediately queued for execution. In typical operation, only the first footstep of a plan is executed because the remainder of the plan is overwritten at the next online re-planning stage.

In this work we focused on navigation of the most challenging terrains, we do not present results showing exploration over flat terrain with protruding obstacles or longer distance path planning. In the case of the former the approach would be to populate the flat terrain with a spanning set of walkable regions (as discussed in [11]). For longer distance path planning approaches for traditional wheeled robots seem reasonable.

III. Stereo Fusion

Dense (passive) stereo correspondence, or matching pixels across calibrated camera pairs in order to infer triangulated distance values, is a well-studied problem in computer vision research; the Middlebury Benchmark System [22], for instance, provides a detailed performance comparison of over 150 algorithms. However, the use of dense stereo in real-time robotics applications — particularly at high frame rates — has been limited. This is mainly due to difficult trade-offs between overcoming inherent algorithmic challenges (properly handling object boundaries and occlusions, disambiguating repeated structures that cause visual aliasing, estimating accurate depth values in regions of low visual texture) and realizing computationally efficient, low-latency implementations suitable for robotic platforms.

A. Active RGB-D Mapping

Active sensors such as the Microsoft Kinect have spurred new interest in dense visual mapping because they directly address several of these challenges. These sensors provide color (RGB) images registered with dense, centimeter-accurate depth (D) images at video rates using an infrared pattern projector and camera pair. All computation is performed on-board using specialized hardware, and devices are available at commodity prices. As a result, active sensors have quickly become the de-facto standard and have been adopted for a wide range of indoor robotics applications.

Shortly after the release of Microsoft's sensor, the KinectFusion system [7] demonstrated real-time RGB-D data fusion within a volumetric data structure (the Truncated Signed Distance Function, or TSDF) maintained in GPU memory. A two-step process of (1) camera-to-map tracking followed by (2) update of the TSDF via parallel ray casting produces highly accurate dense reconstructions of small 4–6m volumes at centimeter resolution. The Kintinuous algorithm [8] was subsequently developed to accommodate larger-scale exploration for mobile robots. It builds upon KinectFusion to enable mapping of extended environments in real-time using robust camera tracking and pose graph optimization combined with non-rigid map deformation for loop closure correction.

While active RGB-D sensors address many shortcomings of passive stereo, they still have practical limitations. The Kinect has a narrow field of view and short range of 4m, but in particular the on-board cameras have rolling shutters and utilize active illumination. This produces blurred or distorted images while moving and cannot be used outdoors due to interference by sunlight.

B. Passive Stereo Fusion with Kintinuous

The Atlas robot's sensor head, the Carnegie Robotics Multisense SL (Fig. 2), is equipped with a pair of high quality global-shutter cameras with wide FOV lenses, and also includes an embedded Field-Programmable Gate Array (FPGA) that implements the Semi Global Matching stereo disparity estimation algorithm [23]. This hardware allows the sensor to produce rectified RGB-D images on-board at frame rate (15-30 Hz) with low latency ($\sim$90 msec) while moving through indoor and outdoor environments without impacting the robot's computational load. While the raw stereo depth from the device outperforms other stereo sensors, it produces normals which are unstable from frame to frame — thus requiring fusion to be useful for our purpose.

The Multisense has a FOV of 90°-by-90° (shown in red in Fig. 3 right), which observes a significant ground footprint in front of the robot while also allowing the background scene to be used for visual odometry.

In this work, we adopt the Multisense as a surrogate for active sensors to generate data suitable for RGB-D fusion algorithms – in particular, the Kintinuous algorithm, due to its scalability, high quality fused fully 3D map representation, and real-time operation. We adapt Kintinuous to function with stereo image data and use the fused output to generate terrain maps for footstep planning.

a) Pre-Filtering: As the footstep planner was originally developed using LIDAR data, it is important that the quality of the maps produced by the vision pipeline be comparable. Visual aliasing, caused by poor image texture and repeated non-distinctive patterns, commonly causes spurious outlier regions within the disparity image and consequently incorrect 3D depth values. An example of the raw data is shown in Fig. 3 (left).

As a result it is crucial to correctly filter and fuse the stereo data before attempting path planning. We first apply a de-noising filter on each frame that (1) finds connected components in the disparity image, where connectedness is determined by spatial pixel adjacency and by similarity of depth values; and (2) removes components below a threshold size, in our case 4,000 pixels. This has the effect of suppressing small isolated disparity regions and pixels that disagree with their neighbors.

b) 3D Fusion: The next step is to build a 3D model of the terrain in front of the robot. Data fusion requires precise

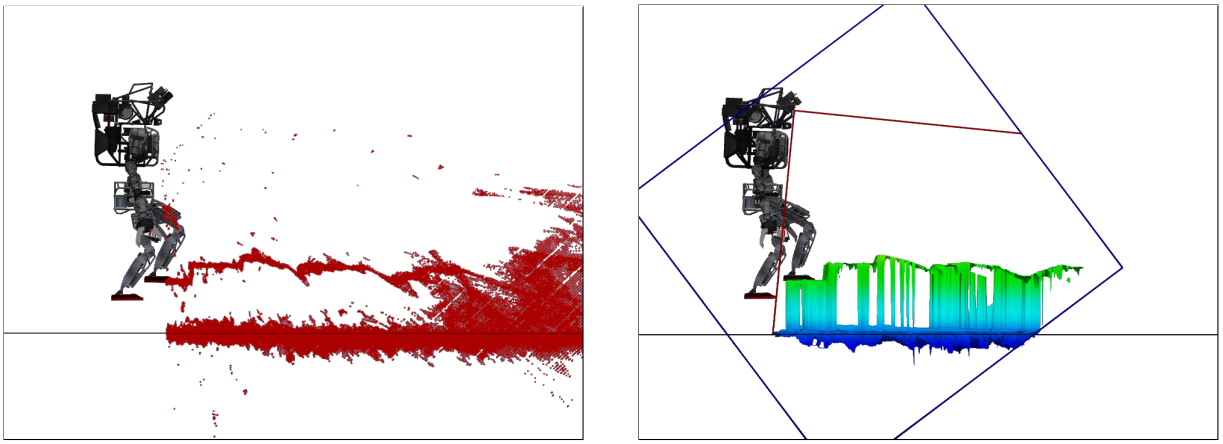

Fig. 3. Left: The raw stereo point cloud input data. The spurious depth pixels in front of the robot can result in the system hallucinating obstructions. Right: A side view of the TSDF volume (blue), the camera FOV (red), and the fused virtual depth image extracted on-line as the robot walks using dense stereo fusion. Note that the unobserved edges of the terrain course cause the untriangulated faces.

knowledge of the sensor's 3D pose over time. To maximize map consistency, we chose to estimate the camera pose and the position of the robot's floating base separately.

The map of the terrain was estimated in the left camera frame C by tracking the motion of the camera frame c relative to its initial position. At time k the pose of camera frame is given by $x_{t,c}^{C:W}$. Several possible tracking methods were discussed in [8], we used the dense photometric warping method described therein which was observed to be more robust to image blur during foot impacts than geometric feature tracking.

Meanwhile our humanoid state estimator, [24], fused inertial and kinematic measurements to estimate the position of the pelvis frame, p, in the humanoid coordinate frame, $x_{t,p}^{H:W}$. Via forward kinematics the camera pose in the humanoid coordinate frame is given by $x_{t,c}^{H:W}$. Thus the map estimated in the camera frame by the Kintinuous system is projected into the humanoid coordinate frame using

$$x_{t,c}^{H:C} = x_{t,c}^{H:W} \oplus (x_{t,c}^{C:W})^{-1} \quad (1)$$

We expected the camera tracking system to regularly fail when foot impacts occur – by causing the camera to shake and motion blur. But this only occurred very occasionally and was aided by the slow, steady speed of locomotion. Nonetheless, in future work we will explore a combined state estimate to avoid this possibility.

Common active dense sensor models have 0.3MP resolution and Kintinuous typically operates at the 30 Hz framerate. The Multisense stereo camera has three times the resolution (1MP) with the processing rate scaling proportionally with the resolution — operating at 9–10Hz.

c) Post-Processing: As described in [8], Kintinuous maintains a 3D TSDF volume of the area in front of the camera on the GPU, which continuously integrates the prefiltered sensor data. The volume is repositioned every few seconds as the robot moves (illustrated in blue in Fig. 3, right). Retrieving the contents of the TSDF volume from the GPU is intensive, so our approach instead is to ray cast a virtual depth image from the camera's viewpoint — an efficient GPU operation that is already part of the algorithm's alignment module. This depth image (also visible in the figure) is then used in subsequent segmentation, applying the algorithms previously described in Sect. II-A.

Each cell within the TSDF has a probability of occupancy which is a measure of confidence. This probability updates as more observations of a surface are made and converges as we become confident of surface's existance. While there is a correlation between the probability and the height estimation error (in Figure 5) typically we found it sufficient to truncate all points from the TSDF which were had only been observed for only a short time.

This approach allows us to implicitly support dynamic scenes. For example quickly moving objects, such as people in front of the robot or the robot's own knees and thighs, are implicitly filtered in this manner, while newly introduced stationary obstacles will be added to the map as observations are accumulated.

IV. Experiments

To demonstrate the described capability we progressively developed the various components of this system with more challenging experiments and more general terrain layouts. Each experiment was carried out in a repeatable manner.

The robot was set up in front of a terrain of uneven cinder blocks and instructed to progress towards a goal, as described in Sect. II-A, and repeatedly did so until it reached the end of the course with no flat surfaces and stopped. We did not implement a general purpose exploration strategy as we focused on the perception and motion planning problems. As mentioned in Sect. II-A, footstep execution was carried out by the manufacturer's stepping controller.

Our computation was amply provided by two identical off-board desktops each with a 3.30GHz Intel i7 CPU and an Nvidia GeForce GTX 680 GPU. The computation time of each step of the vision processing chain was as follows:

- Image acquisition: 105 msec (incl FPGA Matching)

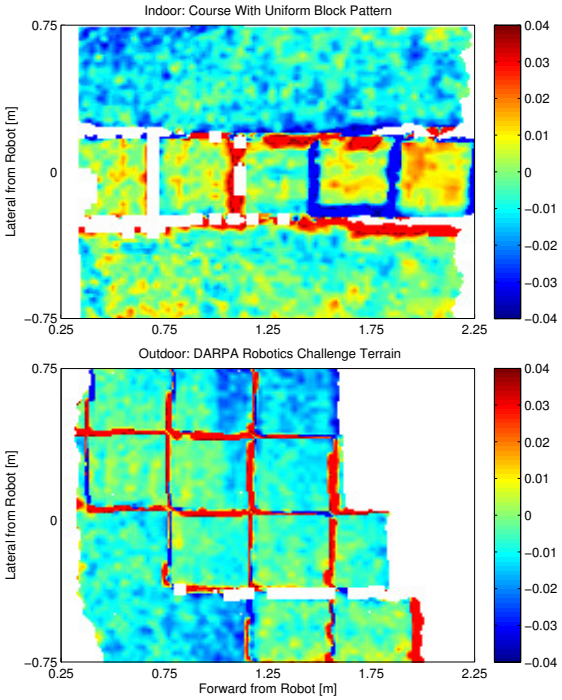

Fig. 4. A visualization of the difference between the heightmap estimated by LIDAR and by stereo fusion (in m) for two experiments — one indoor (upper) and one outdoor (lower). The major deviations around block edges are due to the lower resolution of the LIDAR heightmap.

- Speckle component removal: 40 msec
- Kintinuous stereo fusion: 110 msec (averaging 9.5Hz)

Comparable mean timing for segmentation and planning:

- Planar region segmentation: 615 msec (400-1100 msec)
- Footstep planning: 445 msec (300-600 msec)

Note that each of the modules operated asynchronously on different threads. In particular, the segmentation and planning chain is time critical as there is a 4 sec period (the foot swing cycle) where a new footstep plan can be accepted by the controller.

A. *Comparison between LIDAR and Fused Stereo*

A first experiment used a long row of horizontal cinder blocks at alternating heights and orientations which was approximately 5 m long.

The resolution of our LIDAR heightmaps is a function of the speed at which the actuated sensor was rotated and hence has non-uniform density. So to observe the terrain with enough density to allow normal estimation requires a sweep time of about 6 seconds and a heightmap resolution of 1.5 cm which also requires some unobserved heightmap cells to be filled in by local smoothing. As a result, the scanning speed of this LIDAR sensor, and not the perception, planning or control algorithms, is the unavoidable limiting factor upon the speed of walking of the robot - at roughly half the typical human walking speeding. (Although in these experiments the robot moves slowly enough that this constraint is not as yet a limiting issue).

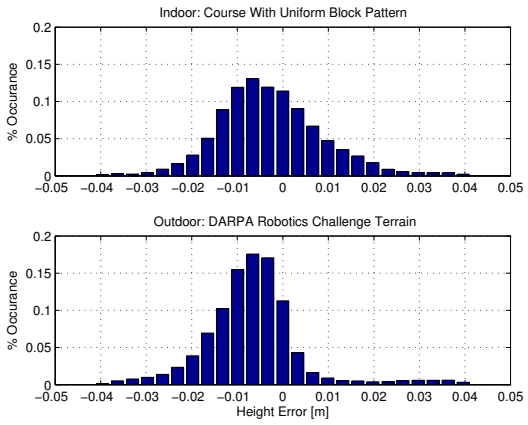

Fig. 5. Distribution of height error for the fused stereo heightmap in Fig 4 (with step edge errors removed). We believe that the slight negative bias is anomalous.

By comparison the stereo camera has an entirely uniform resolution of 1024-by-1024 pixels, which when integrated within the TSDF volume allows us to create a much denser heightmap. We used a resolution of 0.5 cm in this work — 3 times higher than for LIDAR.

In Fig. 4 and Fig. 5 we provide a comparison between LIDAR and fused stereo heightmaps for this indoor experiment as well as a (post-processed) result from the DRC Trials – both collected while in walking motion. We interpret the differences as error in the stereo heightmaps.

The major deviations around block edges are primarily due to quantization of the low resolution LIDAR heightmap. Within footstep regions the mean height error is typically 1 cm — but locally continuous. (The walking controller is robust to errors of 2–3 cm in height and about 20 degrees in the normal of the commanded footstep.)

The marginally degraded performance of the algorithm in the poor lighting conditions of our laboratory is also apparent. The outdoor dataset produced a marginally lower error variance (8 mm *vs.* 11.5 mm). We estimate the variance of a heightmap from raw unused stereo to be approximately 40 mm (and unsuitable for footstep planning). A video showing a like-for-like comparison between raw and fused stereo with LIDAR can be seen at http://youtu.be/0ibv09D3JIw .

A central part of the Kintinuous fusion algorithm is a smoothing and averaging effect. This results in a mild rounding of the edges of the reconstructed cinder blocks which in turn reduces the size of the segmented regions, typically by about a centimeter in the horizontal dimensions. This in turn mildly effects footstep placement.

While this does not cause the robot to step into a dangerous place, it can mean that a slightly longer stride is necessary to clear a hurdle or in rare cases segmentation can fail to find a planar region of sufficient area where the LIDAR would have detected one.

Finally, as equivalent reconstruction performance can be

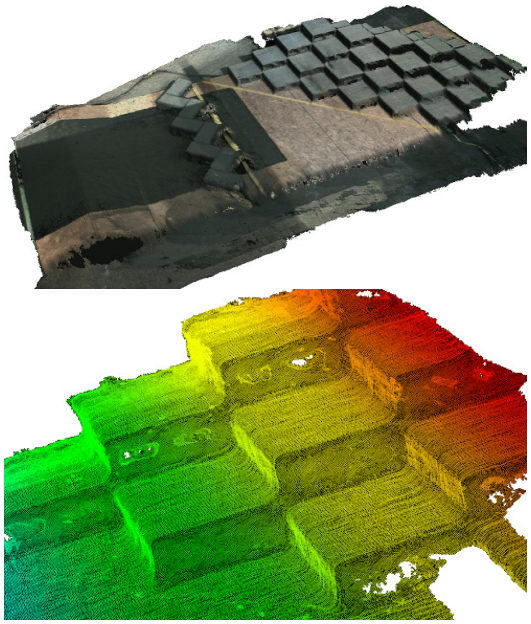

Fig. 6. Reconstruction of the DRC Trials terrain course (approximately 4-by-8m). Note the quality of the reconstruction normals.

achieved with the stereo camera[1], it is also interesting to compare the actuated LIDAR and the stereo sensor on the grounds of added mass and power consumption, which are at a premium on humanoid platforms. The Multisense sensor weighs 2.6 kg and consumes 18-45 W (depending on spin rate) while the stereo component by itself weighs 1.2 kg and uses just 7 W — suggesting that for footstep planning LIDAR sensing is not necessary.

B. Outdoor Mapping Experiments

As referred to in the previous section we also analyzed the data collected during the DRC Trials, which we can use to verify performance of the dense mapping system outside of our own laboratory conditions. Note that the camera used in this experiment had a 40 degree vertical field of view while the collected camera data was at just 5 Hz.

As expected, reconstruction in bright, texture-rich outdoor conditions, results in the precise reconstructions of the terrain course as seen in Fig. 6 — including on the black matte carpet on the left side. We were surprised that the mild texture present in typical industrial and office environments is usually sufficient to produce an accurate fused depth estimate while very poor lighting and high dynamic range are typical limits our approach, as can bee seen in Figure 7.

C. Uneven and Discontinuous Terrain

In our final experiment, we constructed a more general terrain course containing a climb, uneven and tilted steps, cracks, and gaps which is illustrated in Fig. 8 and in the attached video. Note that the spinning LIDAR sensor was specifically covered with a white box to indicate that only stereo vision was used to estimate the terrain.

[1]But of a slightly lower quality than for active RGB-D sensors.

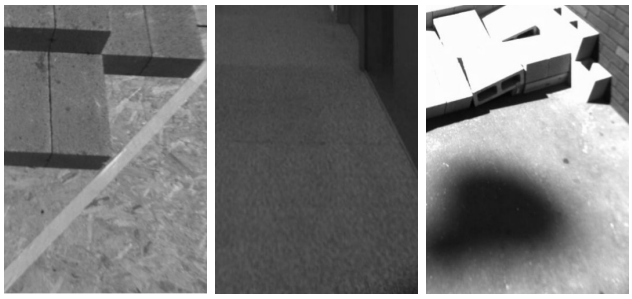

Fig. 7. Texture in industrial and office environments such as the brick, wood (DRC Trials, left) and carpet (center) is sufficient to produce accurate terrain reconstructions, however the poor gain control in an example from the DRC Finals (right) cannot produce a reliable reconstruction.

This experiment demonstrated that the footstep planning could support all these terrain complexities and it also dynamically chose the number of steps to place on each block depending on the configuration. In the latter part of the course the front portion of the steps only became visible just before the robot was expected to step on them which is a particular challenge to the stereo fusion algorithm.

The robot autonomously walked over the entire course in 240 seconds for a total of 25 steps and 14 rows of blocks.

A key part of our success was the low drift rate of our inertial/kinematic state estimator, described in [24]. Over the course of the 25 steps our position drifted in XYZ by $(0.2, 0.0, -0.1)$ meters and 1 degree in yaw — slightly below 1 cm per step. The position drift of Kintinuous' stereo motion estimate was only marginally higher but orientation drift was higher at about 4 degrees.

V. Conclusion

In this work we have outlined terrain estimation and footstep planning algorithms which enable continuous humanoid locomotion across uneven terrain and demonstrated the approach across a 5.5 m, 25–30 step terrain course.

In particular we have also shown how real-time passive stereo fusion can be used as a direct replacement for an actuated LIDAR sensor and that it produces comparable results in challenging conditions. We anticipate the responsiveness and greater resolution of this type visual reconstruction may be required for humanoids to move at human walking speeds in the future. Demonstrations were provided with both indoor and outdoor experiments.

As mentioned in Sect. III, in the future we would like to estimate the robot's motion by fusing visual, inertial and kinematic sensor data so as to robustify the camera tracking system.

It is clear that the optimal footstep placement (i.e. chosen by a human) uses subtle information beyond terrain geometry alone — such as hanging footsteps over step edges and reasoning about occlusions when stepping onto unobserved terrain (e.g. foot-wells). In future work we aim to add such features to our own system as well as to develop more general navigation strategies.

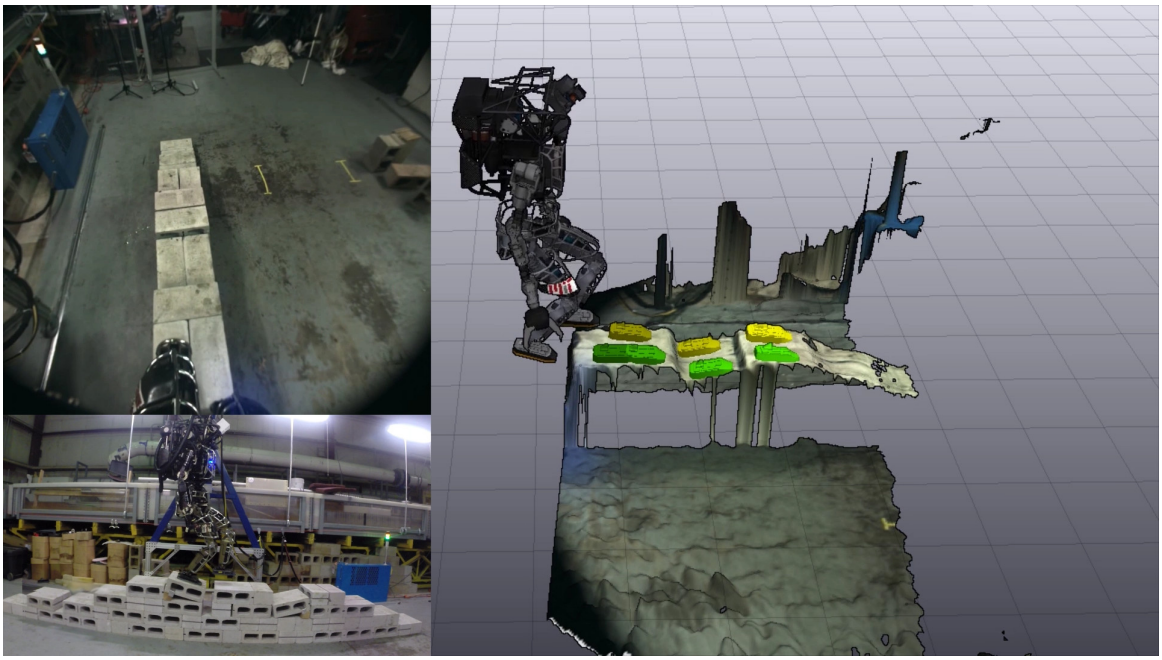

Fig. 8. Visualization of the robot continuously locomoting over a complex terrain course (see Sect. IV-C). The upper left figure shows the camera view from the robot's sensor head. The lower left figure shows a side view of the experiment. The robot's actuated LIDAR sensor is covered up by a white box to demonstrate that only vision is used here. The main figure shows a rendering of the robot's configuration while mid-step and the placements of the next seven steps on the terrain map. Note how the fused terrain map is unaffected by the left leg temporarily appearing in the camera field of view.

REFERENCES

[1] K. Iagnemma and J. Overholt, Eds., *J. of Field Robotics: Special issue on the DARPA Robotics Challenge (DRC)*, vol. 32, no. 2, 2015.

[2] O. Lorch, A. Albert, J. Denk, M. Gerecke, R. Cupec, J. F. Seara, W. Gerth, and G. Schmidt, "Experiments in vision-guided biped walking," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, Oct. 2002.

[3] M. Asatani, S. Sugimoto, and M. Okutomi, "Real-time step edge estimation using stereo images for biped robot," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, USA, 2011.

[4] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, vol. 1, Maui, Hawaii, 2001, pp. 500–505.

[5] J. S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *Intl. J. of Robotics Research*, vol. 27, 2008.

[6] O. Ramos, M. García, N. Mansard, O. Stasse, J.-B. Hayet, and P. Souères, "Towards reactive vision-guided walking on rough terrain: an inverse-dynamics based approach," *Intl. J. of Humanoid Robotics*, vol. 11, no. 2, July 2014.

[7] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE/ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, October 2011.

[8] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald, "Real-time large scale dense RGB-D SLAM with volumetric fusion," *Intl. J. of Robotics Research*, 2015, to appear.

[9] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE/RSJ Int. Conf. on Humanoid Robots*, Madrid, Spain, 2014.

[10] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *Intl. J. of Robotics Research*, vol. 31, pp. 1251–1262, 2012.

[11] R. L. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semi-definite programming," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Istanbul, Turkey, Aug. 2014.

[12] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *IEEE/RSJ Int. Conf. on Humanoid Robots*, Tsukuba, Japan, Dec. 2005.

[13] J. Garimort and A. Hornung, "Humanoid navigation with dynamic footstep plans," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 3982–3987.

[14] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida, "Real-time replanning using 3D environment for humanoid robot," in *IEEE/RAS Int. Conf. on Humanoid Robots*, Bled, Slovenia, 2011, pp. p.584–589.

[15] W. Huang, J. Kim, and C. Atkeson, "Energy-based optimal step planning for humanoids," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 3124–3129.

[16] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE/RSJ Int. Conf. on Humanoid Robots*, Karlsruhe, Germany, 2003.

[17] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK: Cambridge University Press, 2004.

[18] A. Richards and J. How, "Mixed-integer programming for control," in *American Control Conference*, June 2005, pp. 2676–2683 vol. 4.

[19] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2014. [Online]. Available: http://www.gurobi.com/

[20] R. Tedrake, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2014. [Online]. Available: http://drake.mit.edu

[21] D. Kanoulas and M. Vona, "Sparse surface modeling with curved patches," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013.

[22] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Intl. j. of Computer Vision*, pp. 7–42, May 2002.

[23] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[24] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing," in *IEEE/RSJ Int. Conf. on Humanoid Robots*, Madrid, Spain, November 2014.