2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Trust-Driven Interactive Visual Navigation for Autonomous Robots

Anqi Xu and Gregory Dudek

Abstract— **We describe a model of "trust" in human-robot systems that is inferred from their interactions, and inspired by similar concepts relating to trust among humans. This computable quantity allows a robot to estimate the extent to which its performance is consistent with a human's expectations, with respect to task demands. Our trust model drives an adaptive mechanism that dynamically adjusts the robot's autonomous behaviors, in order to improve the efficiency of the collaborative team. We illustrate this trust-driven methodology through an interactive visual robot navigation system. This system is evaluated through controlled user experiments and a field demonstration using an aerial robot.**

I. INTRODUCTION

We present a generic model of "trust" in human-robot systems that is inspired by analogous properties of trust among humans. We demonstrate the use of this mechanism to improve an autonomous robot's task performance through an interactive visual navigation system. This autonomous controller is capable of steering robots based on visual feedback from an on-board camera. We integrated an interactive interface to this autonomous system, to allow a human operator to assist the robot in correcting its poor behaviors and changing navigation targets during an operational session. This user interface is coupled with an adaptation process that dynamically adjusts the robot's autonomous system based on human feedback, and is driven by the objective of maximizing trust within the human-robot team.

The autonomous system presented in this paper addresses the task of boundary tracking, which entails guiding a robot along the boundary of a region of interest that is visually distinct from its surroundings. This is an essential component within many large-scale applications, including the mapping of endangered ecosystems such as coral reefs and rain forests, and the monitoring of disasters such as oil spills and wildfires. On a smaller scale, this boundary tracker can also be potentially used to automate a wide variety of activities, including the inspection of roads and power lines.

Our robot controller includes a number of parameters that can be tuned to track different types of boundaries. A fundamental challenge shared by almost all systems with a generic design like ours is that a poor system configuration may lead to undesirable behaviors that hinder task performance. In one of our previous field trials [1] for instance, while following a coastline, our fixed-wing robotic aircraft encountered a coral reef and incorrectly labeled it as land rather than as water. This mistake caused the robot to fly towards the sea and forced the operator to terminate the session.

The authors are with the School of Computer Science, McGill University, 3480 University Street, Montréal, QC, Canada H3A 2A7 {anqixu,dudek}@cim.mcgill.ca

Fig. 1. We tested our interactive boundary tracking system by using it to guide a quadrotor to follow footpaths and sidewalks on a grass field.

Our work addresses this critical issue in parameterized autonomous systems using a *trust-driven methodology*, which entails allowing a human operator to take over control of the robot whenever it misbehaves. In addition, an adaptive mechanism dynamically adjusts the robot's autonomous behaviors based on the human's interactions. This approach can improve the robot's performance potentially and also enables the task target to change on-the-fly, all the while without requiring the human to have expert knowledge of the algorithms driving the robot's autonomy. We characterize this type of collaboration as *supervisor-worker* (S-W) relationships, since the robot is performing the bulk of the task on its own, whereas the human operator only needs to intervene when the robot is performing poorly.

Our research is motivated by the need to improve a supervisor-worker team's efficiency. This entails both optimizing the robot's task performance as well as reducing the human's task load. Our trust-driven methodology addresses these objectives by building trust within the S-W team, based on the rationale that a trustworthy team will tend to operate more efficiently than an untrustworthy one.

In this paper we describe a generic model that allows a robot under human supervision to infer the human's level of trust in it, based on their interactions. We then demonstrate its application within an interactive boundary tracking system, which is based on work presented in [1]. This robot navigation system includes an adaptive mechanism that is driven by the need to increase trust within the human-robot team. Finally, we present extensive evaluation of this interactive system, comprising of controlled experiments that examine benefits of our trust-driven methodology, as well as a field demonstration using a quadrotor robot, as seen in Fig. 1.

II. BACKGROUND

A. Characteristics of Trust

Our work is inspired by an extensive literature observing the critical role played by trust in teams composed of human

978-1-4673-1405-3/12/$31.00 ©2012 IEEE

agents. Trust is a very rich concept in the modeling of human behavior that incorporates a multitude of interpretations under different contexts, such as within a society, an organization, or a mutual relationship [2]. In this work, we restrict our focus to the last context since it is the most applicable to human-robot systems. Within a mutual relationship, trust comprises of two major elements:

- *the degree of trust*: a quantifiable subjective assessment towards another individual;
- *the act of trust*: the decision and behavior of relying upon an individual's abilities or services.

The trust-driven methodology presented in this work uses the human's inferred *degree of trust* to influence changes in the robot's behaviors, as a means to encourage the human to adopt *the act of trust*.

The degree of trust measures the amount of the truster's assessment, and possibly prediction, of the trustee's abilities. This measure can be affected in two ways: through *direct experiences* or using *trust delegation*. In the former case, the truster builds confidence in the trustee by interacting with it and evaluating the quality of this direct engagement. In contrast, trust delegation [3] involves delegating the task of trust assessment to a third party witness: a truster can choose to adopt a witness's degree of trust in a new trustee by compounding it with the truster's own assessment of the witness. Since our work studies human-robot trust within a mutual relationship context, we will develop a model that updates the degree of trust based solely on direct experiences.

Another prominent topic studied in social sciences is the characterization for the basis of trust. Lee and See [4] surveyed numerous proposed theories on this subject, including bases such as ability, willingness, predictability, and ethics. When dealing with a robot trustee, the bases of trust can be categorized into two major dimensions: typically, a human's trust in a robot is based on notions of its *competence*, such as its accuracy and consistency in carrying out its task. These differ conceptually from factors related to the trustee's *intention*, which pertain to its willingness to perform the assigned task. Our work will take intention-centric bases of trust for granted, by assuming that the robot is always willing to succeed and will never deceive the human on purpose.

Formulations on the bases of trust have motivated the development of different models for the degree of trust:

- In a human-robot system [5], a binary trust attribute is assigned to each state of the world, isolating *trusted* states from *unknown* ones. When the robot encounters an unknown state, it first requests the user to declare that state as trusted; if declined, it then gracefully terminates into an idle trusted state. This model is compatible with our trust-driven methodology, although its drawbacks include limitations in scalability and the assumption that the per-state trust attribute is static and permanent.
- In a multi-agent marketplace, each seller's reputation is represented as a real, non-negative and unbounded value [6]. This metric is updated based on direct experiences: reputation is awarded for satisfactory transactions and deducted for fake or non-satisfactory ones. Akin to an escrow process, buyers avoid purchasing goods from sellers who do not have sufficient amount of reputation to cover a transaction. Our proposed trust inference approach shares many commonalities with this experience-based reputation metric.
- Yu and Singh [7] applied the Dempster-Shafer theory of evidence to measure each trustee's reputation within an electronic market. Degrees of *support* and *plausibility* are used to represent lower and upper bounds on the subjective assessment of a trustee's reliability. This model is combined with trust delegation to devise optimal transaction strategies.

A common theme among various models of trust is the need to distinguish mistrust from uncertainty: knowing with full certainty that an agent will perform poorly is different from deducing that the agent behaves irrationally. Depending on the application and context, trust can be quantified in contrast to mistrust [6], uncertainty [5], or both [7].

B. Related Work

Our work resolves shortcomings in an autonomous robot's programming by allowing a human to assist the fallible agent. This philosophy lies at the core of many research topics in Human-Robot Interaction (HRI), which includes sliding autonomy and Learning from Demonstration.

Sliding autonomy refers to the ability of an autonomous agent to share control over its low-level behaviors with another agent. Brookshire [8] showed that a human-robot team using sliding autonomy can achieve better task performance compared to either a purely tele-operated system or a fully autonomous robot. Dias *et al.* [9] similarly found that enabling sliding autonomy within a peer-to-peer human-robot team lead to faster task completion times and fewer mistakes. Our work uses the properties of sliding autonomy to derive a model of trust for human-robot systems.

The topic of Learning from Demonstration (LfD) addresses the transfer of task-domain knowledge from (typically human) experts to robot learners. Argall *et al.* [10] conducted an extensive survey of these techniques; many of them are compatible as alternatives to our trust-driven adaptation approach. Nicolescu and Matarić [11] presented an LfD technique where a robot learns to complete a given task by observing changes in the world state caused by a demonstrator. This indirect learning approach has the added benefit of allowing a robot student to learn from either a human or robot teacher. Chernova [12] developed a similar LfD framework in which a robot, initially with no autonomous capabilities, can learn new behaviors by incorporating demonstrated state-action pairs into its policy. A unique aspect of this work is that the robot can request a demonstration when the action recommended by its policy has an insufficient *self-confidence* value. This is of particular interest to our work, since self-confidence and trust are complementary properties in human-robot systems.

Our visual navigation system shares design commonalities with other vision-based controllers for aerial vehicles [13],

autonomous surface crafts [14], and autonomous underwater vehicles [15]. In addition, our boundary tracking system draws inspiration from model-based approaches in the autonomous driving literature for automobiles [16], [17].

III. TRUST IN SUPERVISOR-WORKER SYSTEMS

Our research deals with collaborative human-robot systems where the robot is acting under the supervision of a human operator; we refer to this as a *supervisor-worker* (S-W) relationship. The main focus of our work is to maximize the efficiency of these systems, which involves simultaneously increasing the robot's task performance and decreasing the human's task load. To this end, we present a trust-driven methodology that incorporates an adaptive mechanism, known as the *trust module*, to the robot's task-level autonomy. Whenever this module detects that the human has lost trust in the robot's abilities, it will then adjust the robot's behaviors to regain the human's confidence.

This work can be seen as an integration of key concepts behind collaborative systems using sliding autonomy and Learning from Demonstration techniques. From the former perspective, the trust module is designed to bias the sliding autonomy scale towards full autonomy, by adjusting the robot's behaviors to be in line with the human's intentions. This approach also avoids burdening the human with incessant explicit queries regarding the robot's performance, and the need to manually adjust inscrutable parameters of its autonomous system. From the latter perspective, the trust-driven adaptation process is self-initiated and carried out by the robot on its own, and thus does not divert the human's attention away from the collaborative task for the team.

A. System Constituents

Fig. 2 depicts the main elements of a supervisor-worker (S-W) system. Given an assigned task to the team, the robot's *planner module* determines the appropriate actions that will help advance the task progress. Our trust-driven methodology is applicable to any type of robot planner, as long as it can be re-configured to change its behaviors. The planner issues *internal commands* to the *behavior module*, which realizes these requests by affecting the robot's actuators. Separately, a human supervisor can override the planner's commands at any time by issuing *external commands* to the behavior module. Given the asymmetry in the supervisor-worker relationship, commands from the human will always supersede those generated by the robot's planner.

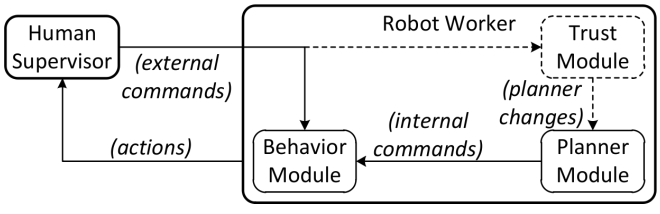

Fig. 2. Block diagram for a supervisor-worker (S-W) human-robot team.

To enforce the command priority and also mitigate time synchronization issues, we assume that the behavior module will process commands only at fixed time instances iQ, for indices $i \in \{1,2,...\}$ separated by the time interval $Q \in \mathbb{R}^+$. Q is a fixed value set to the larger of the two average time delays between consecutive command signals originating from the robot's planner and from the external control interface. Within this discrete system, we define the presence of the supervisor's commands $I_s(i)$ and of the worker's commands $I_w(i)$, for a given time interval, as indicator functions:

$$I_s(i), I_w(i) \triangleq \begin{cases} 1, & \text{if command issued during } (iQ-Q, iQ] \\ 0, & \text{otherwise} \end{cases}$$

Our work introduces a *trust module* into this collaborative human-robot system. The trust module incorporates a quantitative model for inferring the supervisor's degree of trust in the worker, based on the human's intervening commands. This module is also responsible for re-configuring the robot's planner whenever a significant amount of trust is lost.

B. Trust Inference Model

The trust module infers the worker's *reputation* $r(i) \in [0, \rho_{max}]$, as perceived by the supervisor, based on direct experiences in the interaction process. Specifically, reputation is reduced either when the supervisor intervenes or when the robot's planner fails to generate a suitable command. Conversely, reputation is awarded when the human does not override a command issued by the robot's planner. This update mechanism is summarized below:

$$r(i) = \begin{cases} r(i-1) - \Delta\rho, & \text{if } I_s(i) = 1 \text{ or } I_w(i) = 0 \\ r(i-1) + \Delta\rho, & \text{otherwise} \end{cases}$$

Inspired by utility theory [18], we define the supervisor's degree of trust in the worker's competence, $t(i) \in [0,1]$, as the *utility* of the worker's current reputation, i.e. $t(i) \triangleq U(r(i))$. $U(r)$ is a twice-differentiable function defined for $r >= 0$, and has two properties: a positive first derivative $U'(r) > 0$, and a negative second derivative $U''(r) < 0$. In utility theory, an analogous function is used to quantify the preference of a given investment based on its expected monetary return.

The first property of $U(r)$ reflects the rationale that an increase in an agent's reputation should correspond also to an increase in its trustworthiness. As for the second condition, a utility function with a negative second derivative exhibits the trait that the same change in reputation $r_0 \pm \Delta\rho$ will result in a greater loss in utility than the gain in the reciprocal case, i.e. $U'(r_0 - \Delta\rho) > U'(r_0 + \Delta\rho)$. This reflects a popular sentiment of trust among humans, that *it is easier to lose trust than to gain trust*. This imbalanced update property can be found in several other applied models of trust in the autonomous agents literature [19], [6].

Our trust inference model spans between absolute certain trust ($t = 1$) and complete mistrust ($t = 0$). This measure reflects the supervisor's personality through three parameters: the reputation increment $\Delta\rho$, the maximum reputation ρ_{max}, and the choice of the utility function $U(r)$. For a fixed ρ_{max}, a small increment $\Delta\rho$ indicates that the supervisor's degree of trust builds up relatively slowly in light of the worker's good performance. The maximum reputation ρ_{max} determines

a practical cutoff threshold where the incremental gain in utility becomes insignificant for cases when the worker's reputation is sufficiently high. Furthermore, both ρ_{max} and the choice of $U(r)$ affect the gap between the incremental amounts of trust gained and lost, $|U(r_0) - U(r_0 \pm \Delta\rho)|$, given the same change in the reputation $r_0 \pm \Delta\rho$. In general, this disparity is more pronounced for a small prior reputation r_0.

In this work we manually tuned the parameters of our generic trust model to adapt it for a specific interactive robot navigation system. In particular, we employed a logarithmic utility function:

$$t(i) \triangleq U(r(i)) = \frac{log(r(i)+1)}{log(\rho_{max}+1)}$$

Exploring the conformity between this trust model and the true progression of a human's trust is an important research topic, though it is outside the scope of this introductory work.

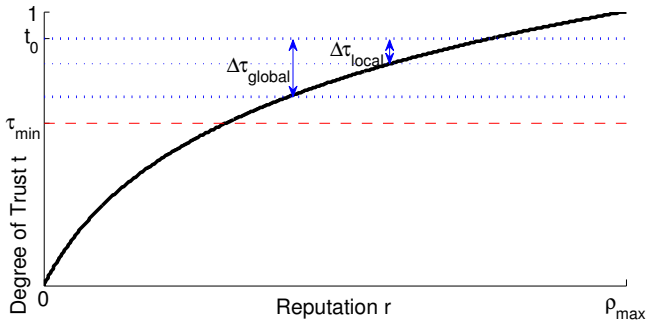

Fig. 3. The supervisor's degree of trust t in the robot worker is modeled as the utility of its accrued reputation r. Given a value of t_0, the trust module will initiate either a *local retraining* process following a sufficient loss of trust $\Delta\tau_{local}$, or a *global retraining* process following a larger loss $\Delta\tau_{global}$.

C. Trust-Driven Adaptation Strategy

The trust module reacts to the human's intervening commands by changing the behaviors of the robot's planner. These interventions can be attributed to one of two causes: either they are meant to make local adjustments to help incrementally improve the robot's current task performance, or they correspond to a complete takeover when the robot is exhibiting poor behaviors persistently or when the objective of the task has changed. Our adaptation strategy addresses these two causes separately, and differentiates between them from the amount of trust lost, $\Delta t(i) \triangleq t(i) - t(i-n)$, within a recently temporal window nQ, $n \in \mathbb{N}^{\star}$. When $\Delta t(i) < -\Delta\tau_{local}$, the trust module will initiate a *local retraining* process that adjusts the robot planner *incrementally* to be more consistent will the human's actions. On the other hand, if a more significant amount of trust is lost, i.e. by more than $\Delta\tau_{global}$, the trust module will then trigger a *global retraining* mode that will completely re-adjust all parameters of the robot's planner based on the human's commands and their effects. Both retraining processes are terminated as soon as the supervisor stops issuing commands, so that the robot can demonstrate its updated behaviors immediately. In the next section we will present concrete implementations of these two retraining processes for a visual robot navigation system.

The trust module will also initiate the global retraining process when the human-robot team first begins to tackle their assigned task, i.e. when $t(i)$ falls below a minimum threshold τ_{min}. Akin to making a good first impression, this bootstrap process configures the robot's planner based on the human's commands, in an attempt to gain the human's confidence immediately. Our complete adaptation strategy is depicted as a finite-state machine in Fig. 4.

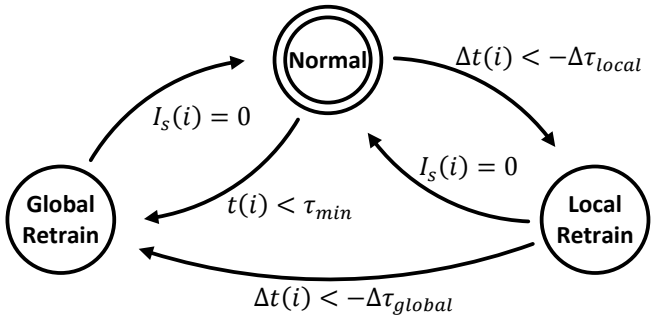

Fig. 4. Finite-state machine representation of our trust-driven adaptation approach, triggered by losses of trust due to the supervisor's interventions.

IV. INTERACTIVE BOUNDARY TRACKER

We embedded our trust-driven methodology in an autonomous robot navigation system for tracking boundaries of various terrains using real-time visual feedback. This robot controller can be deployed on a variety of platforms, including unmanned aerial vehicles [1], autonomous surface crafts [14], and autonomous underwater vehicles [20]. The tracker system extracts boundary information in two phases: first, a classifier segments the scene into disjoint regions, and then a line fit is applied to edge elements belonging to the target boundary. This is subsequently transformed into a navigation directive through the specification of a waypoint or heading in the desired direction. In this paper we present only a brief overview of the visual processing pipeline for our boundary tracker; please refer to [1] for additional details on the system components shown in Fig. 5.

This visual navigation system includes several parameters that can be configured to track different kinds of boundaries. Based on the visual cues that distinguish the target region from its surroundings, the clustering-based classifier uses either hue, brightness, or a combination of both features to achieve the desired segmentation result. In addition, our model-based boundary extraction scheme differentiates between an *edge boundary*, corresponding to two visually distinct regions like the coastline, and a *strip boundary*, where a path is sandwiched between two surrounding regions.

We added a manual control interface and integrated it to our adaptive trust module, which dynamically adapts the boundary tracking process based on user interaction. Specifically, in the local retraining mode the tracker uses the operator's commands to bias its own behaviors incrementally, whereas in the global retraining mode different tracker configurations are evaluated concurrently to determine the boundary model and the classifier mode that are most consistent with the human's behaviors.

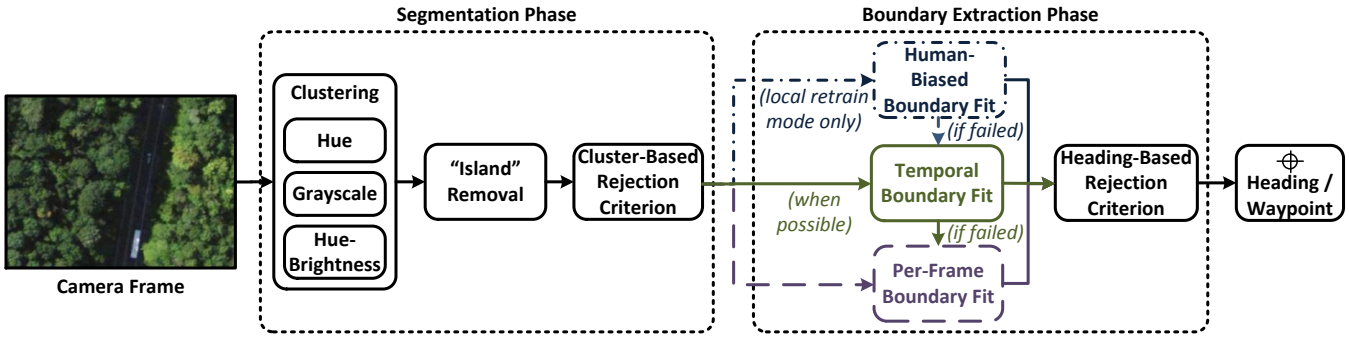

Fig. 5. Block diagram illustrating the visual boundary tracking pipeline of our autonomous robot navigation system.

A. Segmentation Phase

Our visual classifier uses the k-means clustering algorithm to segment images into disjoint regions. The success of this method depends on using an image representation that is capable of discriminating the target region from its surroundings. We use hue clustering to discriminate different-colored terrains, and grayscale clustering to differentiate between neutral-colored regions of varying brightness. For either variant, it is often sufficient to use $K = 2$ clusters. The classifier carries out a small number of k-means iterations per image, to adapt the cluster centers to incremental changes in the scene. In contrast, allowing the k-means algorithm to converge is generally undesirable, since the cluster centers may diverge if the target boundary is not sufficiently in view.

In certain scenes, the target region cannot be discriminated using hue or brightness information alone. For example, submerged patches of rock, sand, or coral near coastlines should be attributed to the blue water region, yet these patches exhibit neutral colors that become ambiguous in hue space. To segment these complex scenes, we carry out cluster analysis using an efficient image representation [21] derived from the hue-saturation-brightness color space. This 2-D feature contains a modified hue-brightness pair that is thresholded by the saturation value, and indicates whether each pixel's color is predominantly hue-based or neutral.

Next, we identify all "*island*" regions in the segmented image, corresponding to connected labeled sets that are not in contact with the image borders. All island regions are merged with their neighboring labels, since these either correspond to extraneous objects in the scene (e.g. boats in the water) or are erroneously-labeled patches produced by the k-means algorithm, as illustrated by Fig. 6(a).

(a) (b) (c)

Fig. 6. (a) For certain scenes such as tropical coastlines, a hue-brightness clustering method is used to isolate neutral-colored regions from hue-dominant ones. (b) The region classifier then removes erroneous "island" regions. (c) Next, a line fit (in yellow) is computed from edge elements (in blue) belonging to the target boundary. *(Note: color images.)*

The quality of each segmented image is assessed by computing the ratio between the two cluster sizes. An image frame is rejected if its cluster ratio indicates that there is insufficient boundary information in the scene. This may occur for instance when the robot momentarily deviates from the boundary due to an external disturbance, or when the classifier mode is poorly configured. We use different minimum and maximum thresholds depending on whether the tracker's boundary model is set to an edge or strip boundary; optimal values for these can be computed experimentally [1].

B. Boundary Extraction Phase

Next, the segmented image is processed through an edge detection algorithm to obtain all boundary locations. The system then computes a line fit for the target boundary, and generates a desired heading command for the robot based on the line's intersection with the image borders.

Given the availability of a recent previous line fit, the boundary extraction process takes advantage of temporal continuity by assuming that the positions of the tracked boundary in consecutive image frames do not change drastically. This temporal tracking strategy entails applying linear regression to the connected group of edge elements closest to the previous boundary line. In contrast, if the temporal tracker fails to produce an adequate heading, or if previous image frames were discarded by the cluster-based rejection criterion, then the line fit is computed statistically from all edge elements in the current frame using RANSAC [22].

Our system evaluates the quality of each generated line fit by comparing the corresponding heading against the last heading command issued by the tracker. In order to prevent large oscillatory movements and also to ensure that the robot does not back-track through its path, a candidate heading is dismissed if its angular distance to the last issued heading exceeds a fixed threshold. This heading-based rejection criterion has been shown in practice to be an effective way of filtering out poor heading commands, even for scenes where humans cannot agree upon the boundary's location [1].

C. Retraining Processes

The trust module adjusts various system parameters as well as the actual boundary tracking pipeline when it detects a significant drop in trust from the human operator. These adjustments are governed by either a local or global retraining process, depending on the amount of trust lost.

When the boundary tracker is operating in local retraining mode, the k-means algorithm is allowed to run till convergence for each incoming image. Although this may cause the cluster centers to diverge from their intended values, in this case we can assume that the human operator is steering the robot attentively to guide it along the target boundary. The boundary extraction process is also altered to compute a line fit from the connected group of edge elements closest to the latest command received from the human. These changes allow the boundary tracker to incrementally adapt its behaviors to the operator's commands based on their effects on the resulting image frames.

If the inferred degree of trust continues to decrease steadily during the local retraining mode, then the trust module will attempt to completely re-configure the tracker's parameters by switching to a global retraining mode. This entails running multiple instances of the boundary tracker simultaneously, for all combinations of classifier modes (i.e. hue, grayscale, and hue-brightness clustering) and boundary models (i.e. edge and strip boundary). After processing each frame, the system computes and stores the error in heading angles between commands generated by each tracker configuration and the latest issued command from the human operator. Finally, when the human releases control over the robot, the trust module terminates the global retraining process, by halting all tracker instances except for the one whose generated headings were most similar to the human's commands.

V. EMPIRICAL VALIDATION

We conducted three sets of human performance studies to determine the efficacy of our trust-driven methodology when applied to the interactive boundary tracking system. The first two sets of experiments are designed to quantitatively measure the increase in the human-robot team's efficiency in terms of task performance and the operator's task load. The third experiment illustrates the practical utility of our approach in an outdoors field trial conducted using a small quadrotor robot. Due to the uncontrolled nature of outdoor flight (e.g. wind gusts, safety, etc.), a quantitative evaluation in the third case is more circumscribed.

We implemented our boundary tracker in C++ in a Linux environment running on a dual-core $1.66\,GHz$ laptop. The visual tracking pipeline operates processes QVGA images (320×240 pixels) in real-time, taking on average $25\,ms$ in normal mode and local retraining mode, and $90\,ms$ in global retraining mode. The generated boundary line fits and heading commands are displayed to the human over a video stream from the robot's camera, and the operator can override the tracker's commands by clicking and holding down the mouse over the display to indicate an alternative heading.

A. Performance Increase for Trust-Driven Boundary Tracker

In the first two sets of experiments, the interactive boundary tracker is used to control a simulated flying robot that returns a stream of aerial images matching its current pose. This allows us to gather real interaction data between the autonomous navigation system and a human operator in a highly controlled and repeatable environment, where external and pragmatic factors such as battery life, lighting conditions, and wind force are either regulated or eliminated. The simulator models a fixed-wing aerial vehicle that operates at speeds of $14m/s$ at an altitude of $250m$. The underlying satellite map used in this experiment was obtained at an effective accuracy of 15 meters per pixel, and upscaled to fit into a simulated QVGA camera frame.

In the first experiment we asked 10 different users, ranging from ages of 20 to 50, to assist the autonomous navigation system in tracking a long and curved street in a partially forested region. None of the subjects had substantive prior experience operating an aerial vehicle under such conditions, and each user was provided with a 5-minute practice interval to familiarize themselves with the control interface.

Each experiment session begins by teleporting the robot to a given location along the long stretch of road. In addition, a random classifier mode and boundary model is assigned to the boundary tracker, to enforce a typical setup where during an ongoing mission the operator decides to switch to tracking the road when it first becomes in view. The user is instructed to activate the boundary tracker but then manually steer the vehicle along the road for roughly 5 seconds, after which control should be relinquished to the autonomous tracker for the remainder of the session. Although in general the road's position is self-evident, the participants are allowed to adopt their own strategy for tracking the road, either by flying down the center or over one of the sides. The experiment session terminates when the autonomous tracker is no longer able to follow the designated boundary.

Each of the 10 participants carried out 10 trials alternating between the use of our trust-driven tracker and a non-adaptive tracker variant, and repeated for 4 different starting locations, culminating in a total of 400 sessions. The resulting trajectories are extracted automatically based on the road's true location.

Because the simulated video stream is generated from actual satellite footage, it exhibits certain visual properties that makes the road recognition task challenging, for example where portions of the road are obstructed by shadows from trees and buildings. In light of these challenges, this experiment examines the extent to which user interaction helped in tuning the tracker to improve its performance.

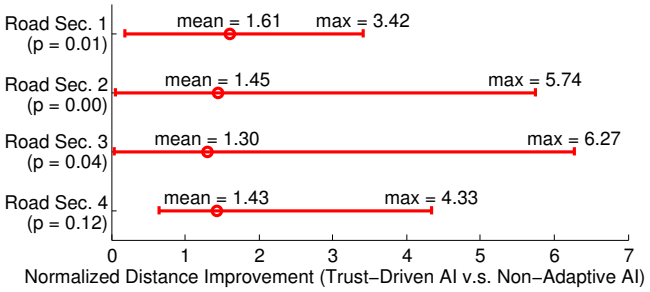

Fig. 7. Performance improvement of our trust-driven adaptive boundary tracker, measured as traveled distances normalized by the average distance of a non-adaptive tracker variant. Error bars represent min. and max. values.

Fig. 7 shows the traveled distance before failure for the trust-driven tracker, normalized over the average traveled distance obtained using the non-adaptive variant of the controller. These results are grouped by the four different starting locations and averaged across all participants. In each set of data, the trust-driven autonomous system resulted in better average performance, and in some sessions it traveled up to 6 times as far as the average tracking distance for the non-adaptive boundary tracker. The differences in the mean traveled distance between the trust-enabled system and the non-adaptive variant are statistically significant, based on 2-tailed t-test ($p_{thresh} = 0.05$, see Fig. 7).

Focusing on the performance of the trust-driven adaptation process, Table I shows that a specific classifier mode was consistently chosen for each road section. In addition, based on empirical observations of these sessions, the split differences between the two boundary models are due to the diversity in each user's steering strategy. The variability in the chosen tracker configuration partially reflects the difficulty of this road following task.

By combining data from all of the 400 experiment sessions carried out, we observe that our trust-driven boundary tracker produced an average increase in performance of 1.45 times over its non-adaptive variant, with a statistical significance factor of $p = 8.80 \times 10^{-5}$. We thus conclude that our trust-driven adaptation process is capable of increasing the task performance of an autonomous robot controller.

TABLE I

PERCENTAGE OF CHOSEN TRACKER CONFIGURATION FOR THE ROAD-FOLLOWING SESSIONS USING THE TRUST-DRIVEN BOUNDARY TRACKER. THE MOST FREQUENTLY SELECTED CLASSIFIER MODE IS HIGHLIGHTED.

Boundary Model	Edge			Strip		
Classifier Mode	Hue	Gray.*	H-B*	Hue	Gray.*	H-B*
Road Section 1	2%	6%	64%	6%	6%	16%
Road Section 2	2%	57%	8%	8%	16%	10%
Road Section 3	16%	50%	22%	2%	10%	0%
Road Section 4	15%	46%	2%	4%	31%	2%

*: Gray. = Grayscale Classifier, H-B = Hue-Brightness Classifier.

B. Reduction of the Human's Task Load

In the second set of user experiments, we examined the ability of our trust-driven strategy to reduce the operator's task load for 2 different navigation courses. The first course involved circumnavigating the coastline of an island, whereas the second course consisted of following the contours of 3 different-colored regions. 10 participants ran each course twice, first using pure tele-operation and then using our trust-enabled boundary tracker. Unlike the previous set of experiments, users were allowed to correct the robot's path as many times as necessary, in order to ensure that it followed the target boundaries accurately.

The resulting 40 experiment sessions lasted between 3 to 5 minutes each. Table II shows that the tele-operated and trust-enabled interactive runs resulted in similar performances. This suggests that our trust-enabled system has an accuracy that is generally comparable to that of a human, though the numerical differences are inconsequential since the setup was not counter-balanced. More importantly however, our results indicate that during the interactive sessions the autonomous tracker was in control for most of the time, meaning that the human users were predominantly monitoring the tracker's performance and did not need to constantly intervene. We also note that the decreased rate of AI control in the "3-terrains" course was due to the user's need to manually switch between different boundary targets in flight. Furthermore, some users were more conservative in their micromanagement attitudes, and continued to override the tracker's commands even after the on-screen overlay indicated that the tracker was following the desired target. We are currently investigating methods to persuade users to adopt trust in the autonomous system in these cases.

TABLE II

TASK PERFORMANCE AND USER INTERACTION RESULTS COMPARING MANUAL TELE-OPERATION AGAINST OUR TRUST-DRIVEN BOUNDARY TRACKER FOR TWO DIFFERENT NAVIGATION COURSES.

	Average Distance from Ground Truth Position		Percentage of Time under AI Control
	(Teleop.)	(Trust-Driven)	(Trust-Driven)
"Coastline"	47.07 m	45.02 m	74.06% (σ=13.63%)
"3-Terrains"	24.59 m	31.79 m	51.12% (σ=11.81%)

The comparable task performance and prolonged automated control results together indicate that our trust-driven methodology, when applied to an interactive robot navigation system, significantly decreased the need to manually steer the vehicle. Consequently, the human's task load is reduced by delegating the navigation task to the autonomous system.

C. Field Trial

We carried out several outdoor field trials where our interactive boundary tracking system was used to steer a quadrotor along sidewalks and footpaths on a grass field. This vehicle has a downward-pointing camera that streams images with a resolution of 176×144 pixels at 5-15Hz, over a wireless network. The interface was modified to allow the operator to override the boundary tracker using a gamepad. These field trials aim to show that the benefits of our trust-driven methodology, as reflected by the previous experiments, can carry over to real-world setups as well. However, our flight results can only reflect these benefits qualitatively, since carrying out controlled quantitative assessments with the same fidelity as in the previous experiments is impractical given the unpredictable nature of outdoor flight due to external disturbances (e.g. wind gusts, lighting variations) and pragmatic concerns (e.g. battery life, safety factors).

Fig. 8 illustrates the performance of our interactive boundary tracker during one of the flight sessions where the human-robot team was tasked to follow three different boundaries sequentially. Because the quadrotor was not equipped with any global localization sensors, we compared the headings generated by the tracker against hand-labeled ground truth headings in each of 1700 resulting frames. Due to wind disturbances, the operator initially struggled to stabilize the quadrotor for the first minute of the flight while following a colored path. For the next 90 seconds, the operator did

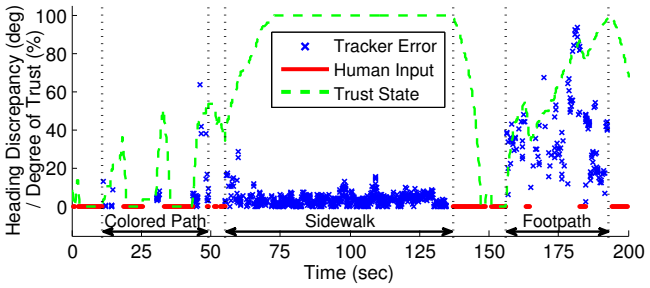

Fig. 8. Errors in heading directives (crosses) generated by our trust-driven adaptive boundary tracker, evaluated against hand-labeled ground truth headings. Periods where the human operator overrode the tracker's commands are shown as solid lines. The dashed line depicts the progression of the human's inferred degree of trust based on our trust model.

not intervene as our automated robot controller correctly detected and followed the sidewalk on the edge of the grass field. After switching to following the footpath, the tracker exhibited moderate performance, due to the visual similarity between the dirt path and its immediate surroundings. This caused the operator to intervene multiple times in the last minute of the session. Nevertheless, during this session our boundary tracker produced an average heading discrepancy of less than 12°, which suggests that in general its steering directives were quite similar to those from a human operator. The success of our boundary tracker in following different boundaries during this field trial, especially while tracking the sidewalk, can be attributed to our trust-driven adaptation strategy.

VI. CONCLUSION

In this paper we presented a general trust-driven methodology that aims to improve the efficiency of collaborative human-robot systems. This approach includes a quantitative measure that infers the human's degree of trust in the robot's autonomy, as well as an adaptation strategy for adjusting the robot's autonomous behaviors to be in line with the human's actions. We applied this trust-driven adaptive strategy to improve the performance of an autonomous robot controller, and we demonstrated through controlled experiments as well as field assessments that the proposed trust-based adaptive strategy both increased the robot's performance and decreased the human's task load.

We are currently investigating several additions to our trust model that will allow it to account for external disturbances and the human supervisor's micromanagement attitude. We are also interested in validating our inferred trust model against the subjective assessment of trust from human operators. Finally, we are motivated to investigate the effectiveness of our general trust-driven methodology when applied to other autonomous robot systems.

VII. ACKNOWLEDGMENTS

We would like to thank Bir Bikram Dey for his invaluable help during the deployment and field evaluation of our interactive boundary tracking system. We would also like to thank all the participants in our controlled experiments.

REFERENCES

[1] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '10)*, Taipei, Taiwan, 2010.

[2] D. H. McKnight and N. L. Chervany, "The meanings of trust," University of Minnesota, Tech. Rep., 1996.

[3] M. Carbone, M. Nielsen, and V. Sassone, "A formal model for trust in dynamic networks," in *Proc. of the 1st Int. Conf. on Software Engineering and Formal Methods*, 2003, pp. 54–61.

[4] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, pp. 50–80, 2004.

[5] R. J. Hall, "Trusting your assistant," in *Proc. of the 11th Knowledge-Based Software Engineering Conference*, 1996, pp. 42–51.

[6] R. Kerr and R. Cohen, "Modeling trust using transactional, numerical units," in *Proc. of the Int. Conf. on Privacy, Security and Trust (PST '06)*, Ontario, Canada, 2006, pp. 21:1–21:11.

[7] B. Yu and M. P. Singh, "Distributed reputation management for electronic commerce," *Computational Intelligence*, vol. 18, pp. 535–549, 2002.

[8] J. D. Brookshire, "Enhancing multi-robot coordinated teams with sliding autonomy," Master's thesis, School of Computer Science, Carnegie Mellon University, 2004.

[9] M. B. Dias, B. Kannan, B. Browning, E. G. Jones, B. Argall, M. M. Veloso, and A. Stentz, "Sliding autonomy for peer-to-peer human-robot teams," in *Proc. of the 10th Int. Conf. on Intelligent Autonomous Systems (IAS '08)*, 2008.

[10] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[11] M. N. Nicolescu and M. J. Matarić, "Experience-based representation construction: Learning from human and robot teachers," in *Proc. of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '01)*, 2001, pp. 740–745.

[12] S. Chernova, "Confidence-based robot policy learning from demonstration," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2009.

[13] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, pp. 217–228, 2009.

[14] Y. Girdhar, A. Xu, B. B. Dey, M. Meghjani, F. Shkurti, I. Rekleitis, and G. Dudek, "MARE: Marine Autonomous Robotic Explorer," in *Proc. of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, San Francisco, USA, 2011, pp. 5048–5053.

[15] J. Sattar and G. Dudek, "Where is your dive buddy: Tracking humans underwater using spatio-temporal features," in *Proc. of the 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '07)*, San Diego, USA, 2007.

[16] Y. Ma, J. Kosecká, and S. S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 521–536, 1999.

[17] R. Aufrère, R. Chapuis, and F. Chausse, "A model-driven approach for real-time road recognition," *Machine Vision and Applications*, vol. 13, pp. 95–107, 2001.

[18] J. Norstad, "An introduction to utility theory," 1999, unpublished manuscript. [Online]. Available: http://homepage.mac.com/j.norstad/finance/util.pdf

[19] T. Tran and R. Cohen, "A learning algorithm for buying and selling agents in electronic marketplaces," in *Proc. of the Conf. of the Canadian Society for Comp. Studies of Intelligence (AI '02)*, 2002, pp. 31–43.

[20] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguère, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis, "A visually guided swimming robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '05)*, Edmonton, Canada, 2005, pp. 3604–3609.

[21] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP '02)*, 2002, pp. 589–592.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.