Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Gain-scheduling Control of a Monocular Vision-based Human-Following Robot ⋆

Michael Burke * **Willie Brink** **

* *Mobile Intelligent Autonomous Systems, Council for Scientific and Industrial Research, Pretoria, South Africa (e-mail: michaelburke@ieee.org)*
** *Applied Mathematics, Department of Mathematical Sciences, University of Stellenbosch, Stellenbosch, South Africa (e-mail: wbrink@sun.ac.za)*

Abstract: This paper presents a gain-scheduling controller for a human-following robot. Typical controllers use either a point-to-point approach where the relative orientation between human and platform is uncontrolled, or a direction-based approach which corrects orientation errors at the expense of additional actuation. We describe the flaws and benefits of each and argue that a gain-scheduling controller combining the two is better equipped to deal with the challenges of human-following. A model of our feature-based, single camera vision system is presented and used to show that the gain-scheduling controller offers better performance than its components, and actual responses to a human following task are used to corroborate this.

Keywords: autonomous mobile robots, robot vision, gain scheduling control, Monte Carlo simulation

1. INTRODUCTION

The ability of a mobile robot to track and follow a target is required in a wide variety of applications, particularly in service robotics. Target-following robots not only need to detect, recognise and track their targets in real time, but also navigate towards them in an intelligent manner. These robots are typically equipped with a diverse set of sensors for locating and recognising targets. Light detection and ranging (LIDAR), for example, provides accurate distance measurements but may lead to ambiguity in target recognition. Electronic tethering is effective but requires that the target followed use a tracking device and often needs a secondary sensor for greater accuracy. As a result many systems employ vision, selected for its ability to provide abundant information about the robot's environment, in a passive manner, at relatively high speeds and low cost.

The control of mobile robots using vision in the feedback loop falls into the well-studied field of visual servo control. Two primary approaches are used: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). IBVS refers to the control of a system from calculations performed only in the image plane with pixel coordinates, while PBVS defines control strategies in terms of the position of the vision system relative to some fixed reference coordinate system. These two approaches are discussed in detail by Hutchinson et al. (1996), and Chaumette and Hutchinson (2008) provide a review. From a PBVS perspective, the control of wheeled robots has typically been divided into two strategies: simple point-to-point positioning without control of orientation and direction-based motion control that takes orientation into account. The latter allows for the tracking of a reference trajectory provided it is feasible. A trajectory is considered feasible if it solves the robot's kinematic model for a set of control inputs (Morin and Samson (2008)). Various control strategies that asymptotically stabilise feasible reference trajectories are discussed in the work of Morin and Samson (2008), Petrov and Parent (2006) and Ma et al. (1999). Point-to-point positioning is the most common approach for moving towards set-points on non-feasible trajectories.

A particularly interesting target following application requiring non-feasible trajectories is that of human-following, as robots equipped with the ability to follow humans could prove extremely useful. Robotic mules could follow humans out to a point and then move back and forth ferrying burdens. Another potential application is in search and rescue, where a robot follows teams of medics and returns stabilised patients to field hospitals.

In this paper we present a hybrid gain-scheduling controller for use in a human-following robot. We argue that there are benefits in each of the typical target following control strategies, and propose a controller combining aspects of each. A feature-based matching scheme is chosen to detect and recognise the human target, in an effort to minimise the likelihood of tracking ambiguity, and an approach to extracting human pose information from a single image is briefly explained. We characterise the limitations in this approach and present a model of a typical feature-based vision system. We use this model to provide simulation results, which show that our controller outperforms two commonly used point-to-point and direction-based controllers. This finding is corroborated through the results of practical experimentation.

⋆ This work was supported by funding from the Council for Scientific and Industrial Research (CSIR), South Africa.

2. METHODOLOGY

2.1 Visual Pose Estimation

Our system operates under the assumption that the pose of a walking person's upper body typically indicates travelling direction. Our approach, which allows a planar fit to the back of a human torso to be obtained from a single image captured by a camera mounted on the robot, is described in Burke and Brink (2010), where we showed that the information thus extracted is suitable for use in a control system. A brief summary follows here. Note that the system requires that relatively salient clothing be worn by the human because the detection is feature-based.

Feature-based target detection relies on the detection of interest points, or features, in a reference and input image. Ideally, these points should be easily recognised from varying scales, angles and under changing lighting. A descriptor of a small region around each feature is extracted and these descriptors matched across images. Numerous matching features indicate a target's presence and can be used with camera geometry and knowledge of a target's structure, to extract relative pose. We use the SURF scheme of Bay et al. (2008) because of its high speed and the good detector repeatability over varying blur, scale and viewpoint. With this method we obtain good matching results for a wide variety of torso motions.

Unfortunately the performance of feature-based detectors typically degrades rather dramatically over changes in scale and viewpoint. Fig. 1a highlights the ability of SURF to detect objects undergoing yaw motions, showing the number of matches detected correctly over angles which can be expected in operation. More and more features are lost as the target rotates, until no detection can be made after about 60°. Fig. 1b shows the robustness of SURF to changes in target scale. As expected, detected features decrease as objects are viewed from further away.

After feature matching a homography is estimated. Projected points $\mathbf{x}_1$ and $\mathbf{x}_2$ in two images (current and desired view) of some feature on the target plane, in homogeneous coordinates, are related by a 3×3 homography matrix $\mathbf{H}$ as $\mathbf{x}_1 = \mathbf{H}\,\mathbf{x}_2$. Note that this relationship assumes a perfect pinhole camera model and therefore requires images to first be dewarped with respect to lens distortion.

The normalised direct linear transform (DLT) (Hartley and Zisserman (2004)) can be used to find the homography from at least four available point correspondences. The problem is likely to be over-specified as typically more than four matches are found. Many correct matches would be useful for a least-squares solution, but incorrect matches can have a drastic negative effect on such a solution. We therefore opt for an iterative RANSAC-based approach (Fischler and Bolles (1981)), in an effort to find a homography that minimises a re-projection error. This robust RANSAC-based homography estimation is effective even in the presence of a large number of outliers, which is especially desirable because of the deformable nature of clothing, the occasional mismatched feature and the slight curvature of a human torso.

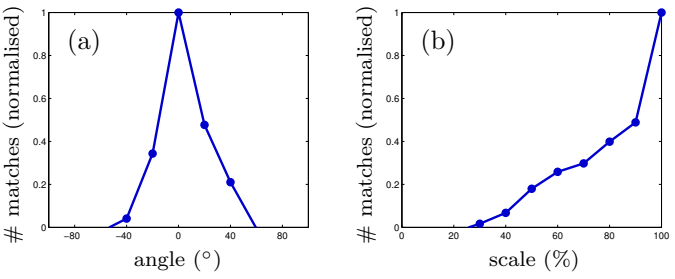


Fig. 1. Robustness of the SURF algorithm to changes in (a) yaw motions and (b) scale.

Once the homography has been determined the various pose parameters, mapping the current camera coordinate system to the desired (template) camera coordinate system, can be retrieved from the decomposition

$$\mathbf{H} = \mathbf{K}\left(\mathbf{R} + \mathbf{t}\,\mathbf{n}^T\right)\mathbf{K}^{-1}, \tag{1}$$

where $\mathbf{K}$ is the intrinsic camera calibration matrix, $\mathbf{R}$ a rotation matrix, $\mathbf{t}$ the translation of the camera and $\mathbf{n}$ a vector normal to the target surface. We use an algorithm of Faugeras and Lustman to calculate the pose parameters in (1) from the estimated homography. The reader is referred to their paper, Faugeras and Lustman (1988), for details.

Note that the translation vector is returned up to scale, because a single camera is used. The unknown scale is time-invariant, however, as each incoming image is compared to a fixed template. Moreover, for the purposes of control, the unknown scale is not a problem as long as the translation components remain monotonic. The controller will minimise error in translation by generating proportional motion commands so, in a sense, the unknown scale is incorporated in the controller gains.

After decomposition a pose vector $\mathbf{p} = [t_x,\ t_z,\ \phi]^T$ is specified. It relates the current view of the human with the desired view, as is illustrated in Fig. 2. (Note the additional angle α. For now we may suppose that $\alpha = 0$.) Here t_x is the cross-track error between the two camera views, t_z the in-track error and ϕ the camera yaw angle. Roll, pitch and out-of-plane translation information is unnecessary and hence discarded. The ability to extract the three parameters of interest independently of the unnecessary degrees of freedom is important though, because it implies some invariance to uneven terrain and various torso motions.

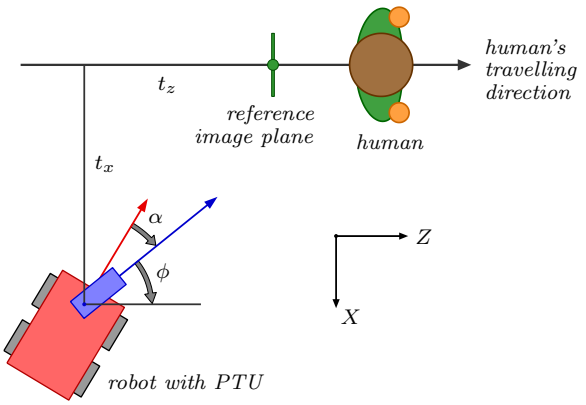


Fig. 2. A graphical interpretation of the parameters t_x, t_z and ϕ that specify the pose vector relating the robot's position and orientation with the human's.

2.2 Tracking

The measured pose estimate can be further refined through the use of a Kalman filter. We select the measurement uncertainty as one third of the typical pose variation in straight line motion, weighted by a factor $w = 1 - n_i/n_t$. Here n_i indicates the number of inliers returned by the RANSAC algorithm and n_t is the total number of available matches. This weighting implies that as the number of inliers increases so does the trust in the estimated homography and its decomposition. It is important to note that the accuracy of the measurement decreases as the target pose moves away from the template or desired view. As a result, there is a greater burden on the controller to ensure that the relative pose stays within acceptable bounds.

We follow an approach similar to one by Yoon et al. (2008) for rigid object tracking to predict pose from a history of estimates. Let $\mathbf{p}_k$ be a pose estimate at time t_k. The predicted pose at time t_{k+1} is then

$$\mathbf{p}_{k+1} = \left(1 + \frac{\delta t_{k+1}}{\delta t_k}\right)\mathbf{p}_k - \left(\frac{\delta t_{k+1}}{\delta t_k}\right)\mathbf{p}_{k-1} + \zeta_{k+1}, \quad (2)$$

where $\delta t_k = t_k - t_{k-1}$ and ζ_{k+1} is the zero-mean prediction noise, with covariance $\mathbf{Q}_{k+1}$, at time t_{k+1}. Assuming that the prediction noise is independent of pose estimates, the predicted covariance of $\mathbf{p}_{k+1}$ is given by

$$\mathbf{P}_{k+1} = \left(1 + \frac{\delta t_{k+1}}{\delta t_k}\right)^2 \mathbf{P}_k + \left(\frac{\delta t_{k+1}}{\delta t_k}\right)^2 \mathbf{P}_{k-1} + \mathbf{Q}_{k+1}. \quad (3)$$

We select the uncertainty in predicted pose, $\mathbf{Q}_{k+1}$, as half the uncertainty in measurement (excluding the homography trust weighting) so as to stabilise the measurements based on their history. This simple model assumes that the time rate of pose change remains constant, incorporating unexpected and unpredictable human motion as noise.

2.3 Control

We use a gain-scheduling controller that phases between the classic strategy for regulating an aircraft onto a reference trajectory and a traditional point-to-point controller. Our system is implemented on a terrestrial skid steering platform with two available control variables: angular and forward velocity. Only forward motion is discussed below, as the control law handling reverse motion is similar. The camera is mounted on a pan-tilt unit (PTU) in order to facilitate the control process and allow certain trajectories to be followed without losing sight of the target. The PTU continually adjusts its azimuth (the angle α in Fig. 2) and elevation so that the centroid of matched features is centred in the camera frame.

The first controller in our scheme is a traditional direction-based controller that regulates both position and orientation. The platform's angular velocity input at time step k is generated by the proportional control law

$$\omega_1(k) = K_1\big[\phi(k) + \alpha(k)\big] + K_2\, t_x(k). \quad (4)$$

Here $\phi(k)$ is the estimated human facing direction extracted from the homography, $\alpha(k)$ the PTU pan angle and $t_x(k)$ the cross track error at time step k. The proportional gains K_1 and K_2 are used to weight the relative errors in the control law. In general these errors cannot be minimised simultaneously as they typically represent conflicting goals, so the weighting is selected such that greater emphasis is placed on minimising cross track error.

The forward velocity control input, $v(k)$, is obtained through the proportional-integral control law

$$v(k) = K_3\left[t_z(k) + \tau_i \sum_{m=0}^{k-1} t_z(m)\right], \quad (5)$$

with $t_z(k)$ the in-track error, K_3 a proportional gain and τ_i an integral term that rejects cumulative errors introduced by a moving target. Note that, because this control system operates in real time, a minimum acceptable sampling rate is required for stability. The integral term is added at the expense of phase lag, which increases the required processing rate and necessitates high-speed image processing.

Unfortunately, while this direction-based controller corrects orientation, it does so by traversing non-ideal trajectories (an example of which is given in the next section). These trajectories make the system vulnerable to losing a fast moving target and are disconcerting as they often differ greatly from the path followed by the target. A more desirable trajectory may be obtained through a traditional point-to-point approach, where the platform's angular velocity input is generated by the proportional control law

$$\omega_2(k) = K_1\,\alpha(k). \quad (6)$$

The forward velocity control law remains unchanged. While this controller follows the target's position closely, it does not take orientation into account. As a result it may be vulnerable to losing the target over sharp turns.

Clearly, a hybrid approach that combines the benefits of each controller is required. If the target is turning sharply, emphasis should be placed on the orientation-regulating controller so as to reduce the risk of losing the target during the turn. When the target orientation does not differ greatly from the platform's, the point-to-point controller, which is less prone to losing a faster moving target, is preferred. The platform's angular velocity input is then generated by the weighted sum

$$\omega(k) = \left(\frac{|\phi(k)|}{\phi_{max}}\right)\omega_1(k) + \left(1 - \frac{|\phi(k)|}{\phi_{max}}\right)\omega_2(k), \quad (7)$$

with ϕ_{max} the maximum detectable orientation angle.

3. RESULTS

3.1 Simulated Controller Responses

The intended behaviour of the three controllers is explained here, with the aid of simulated straight-line and circular responses. These responses are not only useful in describing controller behaviour, but longer paths can be deconstructed into straight-line and circular portions. As mentioned, the point-to-point controller causes the platform to move directly towards a target, with no control of relative orientation. The direction-based controller causes the platform to move in such a way as to approach the target with the same orientation. The hybrid controller attempts to combine benefits of both by phasing between the two depending on the relative orientation angle.

Fig. 3 shows responses to a straight line trajectory. The target trajectory is offset from the platform starting position and orientation. The point-to-point controller does

not converge to the path as quickly as the other two, since it only responds to error in distance. The direction-based controller attempts to cancel out relative positional and orientation errors, but this requires greater control action. The hybrid gain-scheduling controller behaves as expected, producing a trajectory between the other two responses.

Fig. 4 shows the response to an offset circular trajectory. The response of the hybrid controller is not shown, as it is almost identical to the direction-based controller in this case (the relative orientation of a target on a circular trajectory is large enough to enforce a favouring of the direction-based controller). The shorter distance with the point-to-point controller is clearly visible, along with the longer distance required by the direction-based controller. The latter essentially tries to control towards a tangent to the target trajectory, which thus results in a trajectory outside of the target's.

These simulations provide valuable information regarding expected limitations. The point-to-point controller, while traversing a shorter distance, is vulnerable to losing sight of a sharply turning target. This could occur if orientation is not corrected and the relative target orientation moves beyond the limit of the recognition system. While the direction-based controller corrects orientation errors and is not as susceptible to this problem, it does so by traversing a longer trajectory and requires greater actuation. This increases the chances of losing a fast moving target, which may leave the system's recognition range while the platform is attempting to correct orientation and positional errors simultaneously.

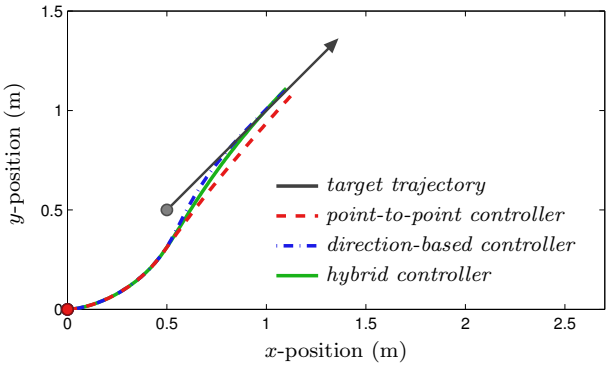


Fig. 3. Simulated responses of the three controllers to an offset straight line trajectory.

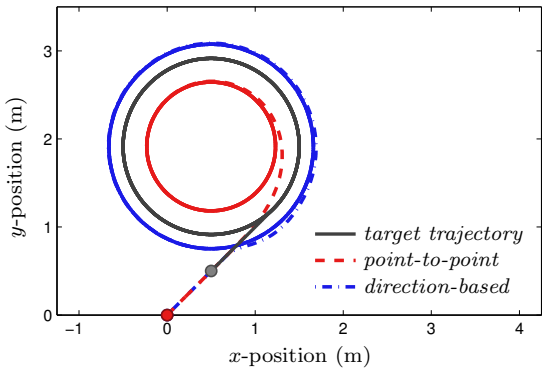


Fig. 4. Simulated responses of the controllers to an offset circular trajectory.

3.2 Monte Carlo Noise Analysis

The operating regions of the control systems described here are difficult to determine, as no closed form model of the vision components is available. We therefore use Monte Carlo analysis in an attempt to show the expected system bounds. We consider only forward and rotational velocities less than or equal to the maximum platform velocities. It is clear that any target motions exceeding these velocities cannot be followed. Moreover, our analysis is concerned only with the platform velocities that can be followed and assumes the platform starts in the desired position relative to the target, with no offset in position or orientation.

Due to the difficulty in accurately controlling the attitude of a target, performance is analysed through a simulation process using a model incorporating limitations and potential problems resulting from the object recognition module. The inclusion of these limitations allows for simulations closely matching reality, because the pose estimation process is not subject to external noise sources.

In our model of the vision system a number of uniformly distributed features are generated on a virtual plane in 3D, and projected onto a fronto-parallel image plane to produce features in the reference image. Given the relative pose between a target and camera, the 3D features are rotated and translated so as to emulate a change in viewpoint. Roll and pitch uniformly distributed between $-20°$ and $20°$ are added to simulate unevenness in terrain or a walking person's torso. The transformed features are projected onto an image plane, and normally distributed noise with variance 10 pixels is added to account for deforming clothing. These features represent matched features in an input image. Given the features in two images the homography-based pose estimate is applied to produce a relative pose measure that incorporates likely noise.

Thus far, no mention on the selection process for the number of 3D points generated, has been made. Fig. 1 showed that the number of matched features decreases with yaw and scale changes, and our model needs to capture this degradation. We model the distribution from which the number of features is drawn naively, under the assumption that the decrease due to scale changes and rotations is independent, and approximate the distributions in Fig. 1 with zero-mean Gaussians with standard deviations $15°$ and 1 m respectively. Fig. 5 shows the resultant distribution used to select the number of features. Note that the

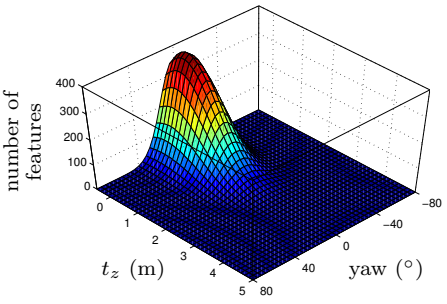
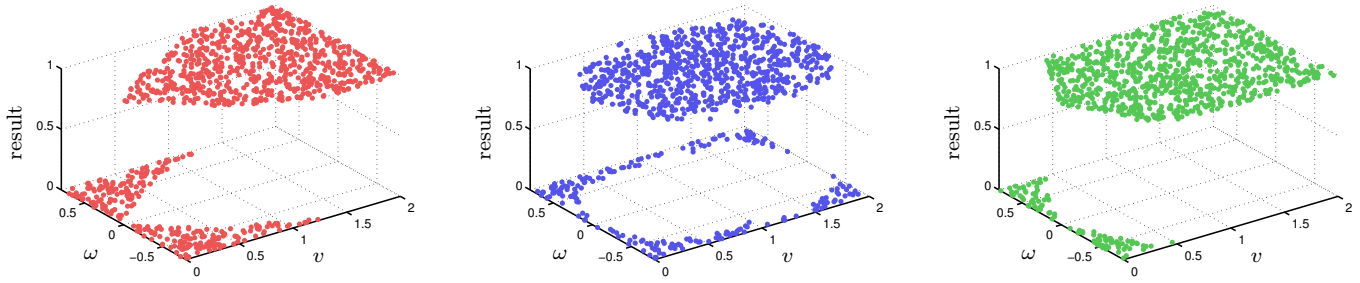


Fig. 5. The distribution from which the number of features in the pose estimation calculations is drawn. Note the dependency on target attitude.



(a) point-to-point controller (b) direction-based controller (c) hybrid controller

Fig. 6. Results of the Monte Carlo analysis. A successful target-following task is denoted by a 1 on the result axis, and a failure by a 0. The angular velocity ω is measured in rad/s, and the forward velocity v in m/s.

chosen standard deviations cause the yaw Gaussian to cut off at about 60°, and the scale Gaussian at about 3 m. This corresponds to the limits of the recognition system as seen in practice, allowing for a realistic simulation.

The simulations consisted of 1000 runs. Initially, uniformly distributed random forward and rotational velocities are generated, used as controls for the target, and remain constant for each iteration. The paths traversed are not important here, but rather whether the platform is able to maintain sight of the target over each simulation. If this occurs, the task is considered successful.

Target position and orientation are adjusted with a standard unicycle motion model. The relative yaw, translations and pan angles used in the controllers are then calculated to form inputs to the vision model. Recall that a certain number of features is required for successful target recognition, so the number of features generated provides a termination criterion to the simulation. If sufficient features are generated, the relative target position and orientation including noise is calculated and used as inputs to the relevant control system which generates platform velocities. They in turn are used to update the platform motion. A delay corresponding to the average image processing rate is also incorporated here. The process continues until a target is lost or a specified distance has been travelled. If the target was not lost the target-following task is assumed to be successful.

Fig. 6 shows simulation results for each of the three controllers. It confirms that while the point-to-point controller is able to follow rapidly moving targets, it is unable to follow sharply turning objects with little forward velocity. The direction-based controller maintains sight of sharply turning targets, but experiences difficulties when following rapid targets. The hybrid controller dramatically improves the performance of both controllers, only experiencing difficulties in following sharply turning targets.

It is important to note that controller stability has not been confirmed here and that the simulations have only covered constant velocity motion within the bounds of the allowable platform velocities. Other motions could still cause the platform to lose sight of a target. In addition, the simulation is model-based and only an approximation of the physical situation. However, the simulations do provide good evidence as to the operation of the controllers, and confirms our proposition that the hybrid controller offers better performance than its components.

3.3 Actual Controller Results

Actual results of the controller responses to target trajectories are now presented. It is difficult to obtain ground truth when following a human target, so the responses here were generated by following a second robotic platform with a salient planar target attached. These pose measurements are not as noisy as those obtained from a human, but still represents a good approximation and allows for the actual behaviour of controllers to be examined. The responses were measured using platform odometry. While this odometry is subject to drift, it is reliable over short distances and hence a sufficiently accurate measure of position for our purposes. We stress that while these platforms are equipped with odometry, our system does not make use of this information and is purely vision-based.

Responses of the controllers to circular trajectories are shown in Fig. 7 and confirm the behaviour exhibited in simulation. Fig. 8 shows step responses to a target offset in both position and orientation. Again, behaviour close to that exhibited in simulation is observed.

In the final experiment a human walked along a predetermined path with the robot following from a preset starting position. Internal robot odometry measurements were logged and are displayed for comparison with the predetermined path in Fig. 9. Although odometry drifts, and the starting point and orientation of the platform could not be accurately controlled, a good idea of the various controller behaviours is obtained. As indicated, both sub-controllers failed during the experiment while the hybrid controller was successful, suitably overcoming the conditions that caused the sub-controllers to fail.

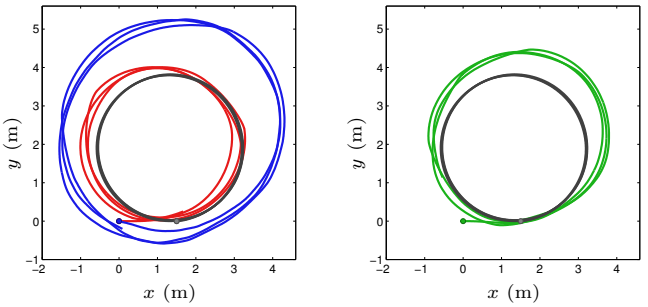


Fig. 7. Response of the point-to-point (red), direction-based (blue) and hybrid controller (green) to a target object following a circular path (grey).

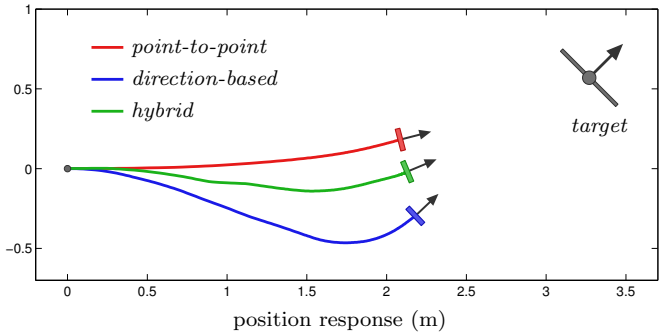


Fig. 8. Responses of the controllers measured as each navigates towards a stationary target.

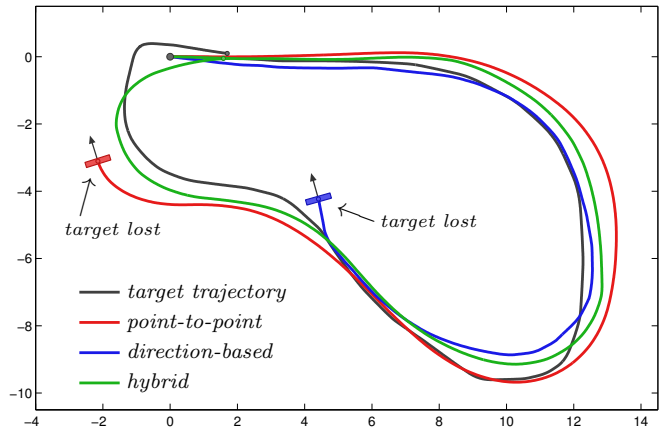


Fig. 9. Position responses (in metres) of the controllers over a predetermined path.

3.4 Limitations due to Motion Blur

The simulation results have indicated that the hybrid gain-scheduling controller is able to follow targets at speeds of up to 2 m/s. Unfortunately, in practice, the allowable target speed is significantly lower due to the effects of motion blur. Practical experimentation, using a laser tracker to determine human walking speeds, shows that the system is capable of following a target at approximately 0.7 m/s. Motion blur introduces significant smoothing and removal of feature information. As a result, insufficient features are extracted and feature-based recognition strategies would fail. While predictive tracking such as the Kalman filter approach used here does assist in countering motion blur, it is only able to do so for a brief period of time.

The primary causes of blur in our human-following system are sharp movements of the PTU and abrupt platform accelerations. These movements form part of the controller response to a rapidly moving target, so the need to minimise them by ramping up speeds results in a decrease in the allowable speeds which can be tracked.

Despite limiting the system's speed of operation, the effects of motion blur do not affect the conclusions made in this work. Analysis of Fig. 6 shows that the hybrid controller still offers better performance than its sub-controllers in the practical system's range of operation. It should also be mentioned that blur can be countered through better control of lighting conditions, by using better cameras, or through software algorithms.

4. CONCLUSIONS

We have presented an approach to robotic human-following using pose information extracted from single images. Traditional point-to-point controllers are prone to losing targets that turn sharply. Direction-based controllers remedy this, but can lose rapidly moving targets due to non-ideal trajectories. A hybrid gain-scheduling controller that combines benefits of the two has been presented and shown to outperform them. While the point-to-point controller would not suffer orientation-induced losses if all sides of the target were recognisable, we believe that the use of the hybrid controller to follow a target intelligently is a much less involved problem and therefore more suitable for real-time following.

A model of our vision system has been presented and used for simulation. Planned future work involves using this model to evaluate potential paths and select those which maximise feature visibility. Collision avoidance is easily incorporated in this approach, making the system practically feasible. In addition, the inclusion of other sensors and target trackers will lift the burden of the vision system and assist in reducing the effects of motion blur.

REFERENCES

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346–359.

Burke, M. and Brink, W. (2010). Estimating target orientation with a single camera for use in a human-following robot. In *Proceedings of the 21st Annual Symposium of the Pattern Recognition Association of South Africa*, 51–56.

Chaumette, F. and Hutchinson, S. (2008). Visual servoing and visual tracking. In *Springer Handbook of Robotics*, 563–583.

Faugeras, O. and Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2, 485–508.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.

Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5), 651–670.

Ma, Y., Košecká, J., and Sastry, S. (1999). Vision guided navigation for a nonholonomic mobile robot. *IEEE Trans. on Robotics and Automation*, 15(3), 521–536.

Morin, P. and Samson, C. (2008). Motion control of wheeled mobile robots. In *Springer Handbook of Robotics*, 799–825.

Petrov, P. and Parent, M. (2006). Adaptive control for reversing a two-vehicle platoon. In *11th IFAC Symposium on Control in Transportation Systems*.

Yoon, Y., Kosaka, A., and Kak, A. (2008). A new Kalman-filter-based framework for fast and accurate visual tracking of rigid objects. *IEEE Trans. on Robotics*, 24(5), 1238–1251.